EAST WEST UNIVERSITY

Analysis of Real Time Face Mask Detection Using Transfer Learning Method

This Thesis Paper Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Information and Communications Engineering

Prepared For

Dr. Anup Kumar Paul

Associate Professor

Department of

Electronics & Communications Engineering,

East West University

Prepared By

Syeda Fariha Tabassum

ID 2018-1-50-027

Nilufa Yasmin

ID 2017-3-50-001

Department of Electronics & Communications Engineering

East West University

Date of Submission June 2022

Approval

The Electronics and Communications Engineering (ECE) program requires this thesis as a requirement. This is to certify that Syeda Fariha Tabassum (ID: 2018-1-50-027) and Nilufa Yasmin (ID: 2017-3-50-001) have submitted a thesis titled **"Analysis of Real Time Face mask detection using Transfer Learning Method"** to the Department of Electronics and Communication Engineering, East West University, Dhaka, Bangladesh, to fulfill the partial requirements for the degree of Bachelor.

Approved By-

____

Supervisor

Dr. Anup Kumar Paul

Associate Professor

ECE Department,

East West University

Dhaka.

Declaration

We hereby declare that, the contents of this research project are original and have not been submitted in whole or in part for consideration for any other degree or qualification at this or any other university, except where specific reference to the work of others is made where the amount of plagiarism is within acceptable range.

Syeda Fariha Tabassum

2018-1-50-027

Nilufa Yasmin

2017-3-50-001

This paper is dedicated
To
Our beloved Parents and honorable teachers

Acknowledgment

At first, we want to express my heartfelt gratitude to almighty Allah for providing me with the power and capability to complete this assignment on time. We want to convey my deepest thanks and heartfelt gratitude to everyone who assisted me and gave me the opportunity to finish and deliver my report. Moreover, we want to express our heartfelt thanks to our supervisor, Dr. Anup Kumar Paul. We would not have been able to finish our thesis without his insightful mentoring. His suggestions forced us to think in new ways, his observations improved our problem-solving abilities, and his enthusiasm gave us courage in the face of setbacks. Working with him has been an honor and a privilege for us.

Table of Contents

List Of Figure

List Of Table

Abstract

Wearing a mask is one of the non-pharmaceutical strategies that can be utilized to reduce the principal source of SARS-CoV2 droplets ejected by an infected person. Regardless of debates over medical resources and mask varieties, all countries require public use of masks that cover the nose and mouth. A pre-trained Convolutional Neural Network (CNN) is used as a feature extractor for the images. CNNs are a kind of Deep Neural Network that can detect and categorize certain characteristics in images, and they're commonly employed for image analysis. To make the method as accurate as possible, the fundamental Convolutional Neural Network (CNN) model is developed using TensorFlow, Keras, Scikit-learn, OpenCV etc. Pre-processing, training a CNN, and real-time classification are the three parts of the proposed study. Several pre-trained deep Convolutional Neural Networks were developed and validated using the transfer learning approach and image augmentation (CNNs).

In this paper, we are providing model proposal which will detect mask faces of individuals in real-time. Facemask detection algorithms are a subset of object detection algorithms, which are used to detect things in images. Among the numerous object detection algorithms, deep learning outperformed traditional machine learning algorithms in facemask detection because of its superior feature of extraction capacity. This paper presents a transfer learning approach for mask detection. For face mask detection three models (AlexNet, MobileNet V2, VGG-16) are being used here. Each of them has different accuracy. Moreover, applying the same model in a different way also provides different accuracy. Validation accuracy of VGG-16 is 96.67%, where it is 98.50% for MobileNet and for the proposed model it is 98%. Though AlexNet has the validation accuracy 94.26% but when the model is fitted using augmentation the accuracy rose to 97%. Not only that, along with the changing of model fitting pattern if the learning rate, batch size is also changed with accuracy seems to have 98%. And this is the highest of AlexNet. F1 score, recall and precision of MobileNet is (0.99-for with mask and 98 for without mask, 1.0-with mask; 0.99-without mask, 0.98-for with mask and 99 for without mask. For VGG 16, f1 score = 0.44-with mask/0.45-without mask, precision = 0.45, recall =0.44-with mask/0.46-without mask. For AlexNet, f1-score is 0.94- for both with and without mask, recall = 0.92 without mask, 0.96 = with mask, precision = 0.96 for without mask and 0.92 with mask. As the accuracy changes for AlexNet for using different ways to apply the algorithm, precision, recall and f1 score is also different for each state. The Javascript API facilitates camera access for real-time face mask detection.Because Google Colab operates on a web browser, it is unable to access local hardware such as a camera without the use of APIs. But through a code it is done here.

CHAPTER ONE

1 Introduction

1.1 Covid-19

The novel COVID-19 first recorded in the Chinese province of Wuhan in December 2019 exploded rapidly all over the world and soon became a worldwide problem. It seems to have a profound effect on everyday life, public health and the global economy. COVID-19 is the most recent outbreak of Coronavirus disease. COVID-19 is a respiratory infection that affects the body's immune system. The SARS-CoV-2 virus is responsible for the disease. It is primarily disseminated from aerial communication of person to person, particularly by close proximity During the pandemic, a wide range of a number of artificial intelligence-related research.[1] The number of cases in most courtiers has recently increased due to community transmission. Around 49 million confirmed cases have been reported globally, according to the World Health Organization (WHO). Due to the outbreak of COVID-19, the WHO has issued several precautionary guidelines to fight against the spread of coronavirus. Social distancing, sanitization, and wearing masks are the most noticeable guidelines. Wearing a face mask slows the community transmission of the corona. Thus, the majority of the countries have enforced compulsory face mask policies in public areas. Manual observation of the face mask is a tedious task especially in crowded places such as hospitals, airports, railway stations, and shopping malls.[2] since the New Normal has been implemented, the people are forced by law to wear a face mask in the public place and wherever they interact with other people. the government punishes all the people who do not wear a face mask in a public place to do some physical punishment, such as doing a push-up.

1.2 Other diseases and situations for wearing mask

Except the Covid-19 disease, there are move several situations and diseases for which one needs to wear a mask. Flu is one of them. Flu spreads easily through sneezing and coughing. Covid-19 is basically a kind of flu but more dangerous than it. ‘Influenza’, ‘Respiratory infection’, ‘Personal protective equipment’, ‘Disease prevention’, ‘Compliance’ and ‘Adherence’ are the terms which makes wearing a mask as an essential step. Also, doctors need to wear a mask while treating patients. People who are older, those who have specific medical issues, and those who are pregnant or have just been pregnant are at a higher risk of any kind of disease. So, while visiting them one needs to wear a mask as precaution. While traveling, wearing a well-fitting mask can help protect oneself and others.

1.3 Benefits of wearing a mask

Acute respiratory infections are common and continue to be a concern to society. Facemask have been shown to be an efficient barrier in reducing the aerosol transmission of infectious illnesses, but their use in the local population is infrequent, leading to reservations about their efficiency in avoiding airborne infections during epidemics. As a result, we set out to undertake a literature study to see what factors impact the community's usage of facemasks as a main preventive health strategy. By acting as a barrier to virus droplets, wearing a mask dramatically reduces the virus's transmission. They prevent virus-infected droplets from spreading from one sick individual to another. In researches it is shown that, face masks to help limit the transmission of the virus by 17 percent. You will be less exposed to the virus if you use face masks. It will help you avoid becoming infected with the virus and reduce your chances of experiencing severe symptoms if you do become sick. When you wear a mask over your face, you release fewer particles into the air because some are contained inside the mask. It will also protect you from breathing other people's germs.

1.4 Technologies to detect face-mask

Rapid technological advancements have brought us to a place where we can now do feats that were once thought inconceivable. Machine Learning and AI have simplified life and provided solutions to a wide range of challenging challenges in a variety of fields. Machine Learning and Deep Learning algorithms are reaching human-level performance in visual perception challenges. In the fight against the Coronavirus Disease (COVID-19) epidemic, technology is saving lives. Thanks to technology improvements, work from home has supplanted our traditional job routines and has become a part of our daily life. Some industries, however, are unable to adjust to this new normal.

Individuals are still apprehensive to return to work as the pandemic subsides and such sectors return to in-person work. For 65 percent of employees, returning to work has become a cause of anxiety. Multiple studies and have demonstrated that wearing a face mask reduces the risk of viral transmission while also offering a sense of security **[3].** However, executing such a policy and tracking any violations on a broad scale is impossible. Computer Vision is a better alternative. Using a mix of image classification, object identification, object tracking, and video analysis, we built a system that can detect persons wearing face masks in photos and videos. Stage 1 detects human faces, whereas Stage 2 employs a lightweight image classifier to identify and localize the faces recognized by Stage 1 as 'Mask' or 'No Mask.' This method is superior to other object detectors because it employs transfer learning and pre-trained models. As a result, the suggested system achieves excellent accuracy with minimal training data.**[4]** Most current object detectors require a big human-annotated dataset and computational resources. The inference time is substantially faster than that of other object detectors. This system can be integrated with an image or video capturing device, such as a CCTV camera, to track safety violations, promote the use of face masks, and maintain a safe working environment. An object tracker was integrated with our face mask identification algorithm to extend our application to movies.**[5]**

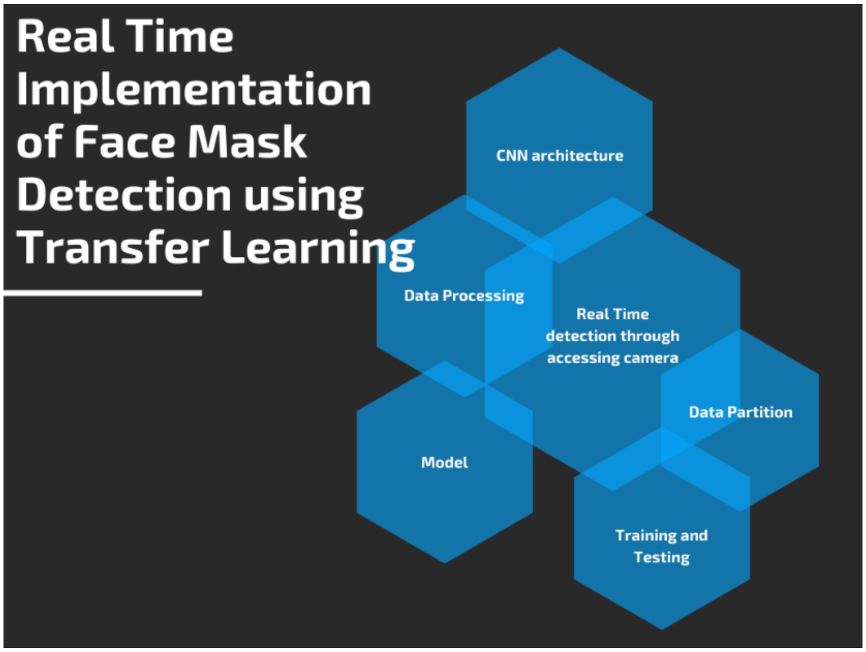

Real Time Implementation of Face Mask Detection using Transfer Learning
CNN architecture
Data Processing
Real Time detection through accessing camera
Data Partition
Model
Training and Testing

Fig 1.1 Face Mask Detection Steps

Artificial intelligence approaches and methods should be used to support decisions made by the healthcare system and the social system in their efforts to sustain every step of the decline and its consequences: identification, security, reaction, recovery, and accelerated study, as disease is becoming a massive disaster.[6] Modern AI algorithms paired with facial imaging may be beneficial for accurate disease detection as well as addressing the problem of medical practitioners being few in distant places. In order to avoid a global pandemic epidemic, the research presented in this paper incorporates relevant developments in the domains of public safety and biomedical research.

Image and video-based detection algorithms powered by artificial intelligence can accurately detect an object and decide whether or not a human is wearing a mask. Face mask recognition can be done using deep learning and machine learning techniques like support vector machines and decision trees on a variety of datasets. This papers' major goal is to Detect face masks with transfer learning method. Moreover, in this paper a proposed model is created using the CNN architecture. A number of machine learning packages, deep learning methodologies, and image processing techniques are used to the OpenCV framework in a face mask detection model.

A real-time application (RTA) is one that operates in a time period that the user perceives to be immediate or current. The delay must be within a certain threshold, commonly measured in seconds. Real-time computing includes the usage of real-time applications. In this paper, real-time has been implemented to detect the face-mask.

CHAPTER TWO

2 Literature Survey

The inherent tasks that a computer vision (CV) method must deal with is object recognition. Image classification and object detection are both included in object recognition **[7].** Face detection models were previously constructed utilizing edge, line, and center near features, with patterns detected from those features. These methods are used to locate binary patterns on a local level. These methods are particularly successful for dealing with gray-scale data, and they involve relatively little computational work **[8].**

A CNN model for fast face detection analyses a low-quality input picture, excluding non-facial areas and properly processing the regions with higher resolution for exact detection is introduced by Li et al**. [9].** In their paper, Khandelwal et al. proposed a deep learning model that binarizes an image as mask or no mask. A mask was applied to 380 photographs, whereas no mask was used to 460 images, and these images were utilized in the training of the MobileNetV2 model. The model has a few flaws that were discovered. The model is unable to recognize faces if the camera height is more than 10 feet, and it could not accurately categorize partially hidden faces **[10].**

Jiang et al. introduced a Retina face mask detector, which is a high-accuracy and efficient face mask detector. ResNet and MobileNet are the models employed. Robust properties were extracted using transfer learning, which was trained on a huge dataset of 7959 photos **[11].** Viola Jones Detector presented a real-time object model for detecting various object categories. It evaluates any picture with edge, line, and four rectangle characteristics using a 24x24 base window size. Harr-like features are similar to convolutions as they examine whether a certain feature is present in the image. When picture brightness fluctuates, this model fails, and it also performs poorly when images are in various orientations **[12].** Satapathy, Sandeep Kumar created a model to detect number plates, which is a critical issue that aids police in pursuing numerous criminal cases. The authors employed an OCR-based technique to recognize characters in the number plate, which were then saved and processed in a client-server paradigm to obtain the owner's information **[13]**

Pathaket developed a multi-dimensional biometric identification system that works well in low-light environments. The accuracy of the system was increased by employing an entropy-based CNN **[14].** Sign language identification was achieved by training a CNN model that can recognize signals in video, according to Ravi, Sunitha. it's even beneficial in machine translation for sign language. CNN employed a joint angular displacement technique to improve its ability to record 3D motion sign language in real time, which may be applied to a variety of real-time applications these days **[15].** Patel, Ashok Kumar, and colleagues devised a methodology for determining the grade of iron ore by extracting characteristics from mining sample material **[16].** SVR is a support vector regressor that is used to measure ore quality in real time. During this procedure, 280 features were extracted in order to identify. Matthias et al. have worked on a face mask detection project that focuses on capturing real-time photos that indicate whether or not a person is wearing a mask. The dataset was utilized for both training and implementing the decision-making algorithm to recognize the primary face characteristics (eyes, mouth, and nose). Putting on spectacles has no negative consequences. Rigid masks performed better, however inaccurate detections might occur owing to lighting and things seen outside of the face [17].

Ristea et al. proposed a data augmentation model for face mask detection from speech. That might be utilized for surgeon communication, forensic fields, or infectious disorders such as coronavirus [18]. They built a project that could conduct binary classification using several ResNet models and trained Generative Adversarial Networks (GANs) with cycle consistency. Masked Face Detection Dataset (MFDD), Real-world Masked Face Recognition Dataset (RMFRD), and Simulated Masked Face Recognition Dataset are the three samples of mask faced datasets which are provided by Wang et al. [19]. Madhura et al. presented Facemasknet as a model for determining if a person is wearing a facemask properly or not, with three classifications: no mask, improperly worn mask, and with a mask. They claimed to have two detectors in their efforts. The face is detected first. The RoI is calculated, and the Facemasknet model is used to classify the cropped photos or live feeds. The Facemasknet model had a 98.6% accuracy rate [20]. Vinitha and Venlantina suggested a computer vision and deep learning model. This model could recognize facemasks in real time from security cameras and photos. In their model, which was trained on a big dataset, they employed TensorFlow, OpenCV, Keras, and MobileNetV2 architectures [21].

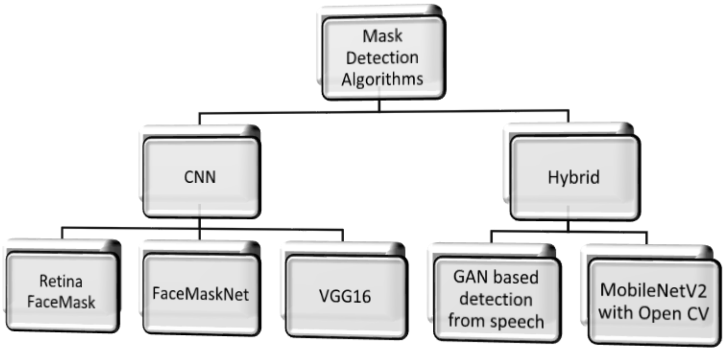


Fig 2.2 A hierarchical representation of facemask detection algorithms

In this paper, we are using three different algorithm and those will detect a person with/without mask. These models are trained in such a way that they will detect eyes, nose and mouth to know if a person is wearing a mask or not. The limitation of these models are they will detect a mask even if it is wearing in an improper way. Well not only that, through one of the applied models, we have implemented a code that will take real time video to detect a person with/without mask. A green rectangular box will appear for with mask and a red one will appear for alternate one.

CHAPTER THREE

3 Dataset, Methodology and Materials

Face mask identification is achieved success by using a machine learning algorithm. The trials in this study used two different datasets. The first dataset was derived using the Kaggle dataset and the Real-World Masked Face dataset (RMFD), and was used for training, validation, and testing the model on the dataset. The model can be created by doing the following steps: (1) data collection, (2) pre-processing, (3) data split, (4) model development, (5) model testing, and finally (5) model implementation. Diagram in Fig 3.14

The model is applied to the dataset using the second dataset. Data availability dictated the selection of some cities. The data comes from a variety of places, including public surveillance cameras, shops, and traffic light cameras. The photos were picked for quota sampling based on the population proportional size of the cities, while the duration of capturing the image was equal for each city.**[29]**

3.1 Dataset

Masks are one of the few COVID-19 precautions accessible in the absence of vaccination, and they serve a critical role in safeguarding people's health from respiratory infections. It is feasible to develop a model to recognize persons wearing masks or not wearing them using this dataset. The datasets are same for every model. Same amounts of datasets but for VGG16 the datasets are already in a split form of train and test. The CNN architecture is quite same for each of the model. The dataset is divided into two sections.

- With Mask
- Without Mask

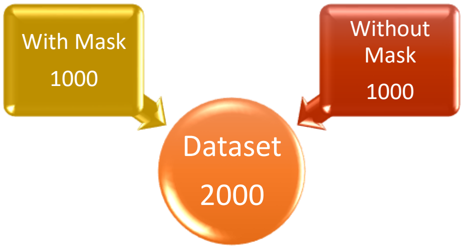


Fig 3.1 Dataset Quantity

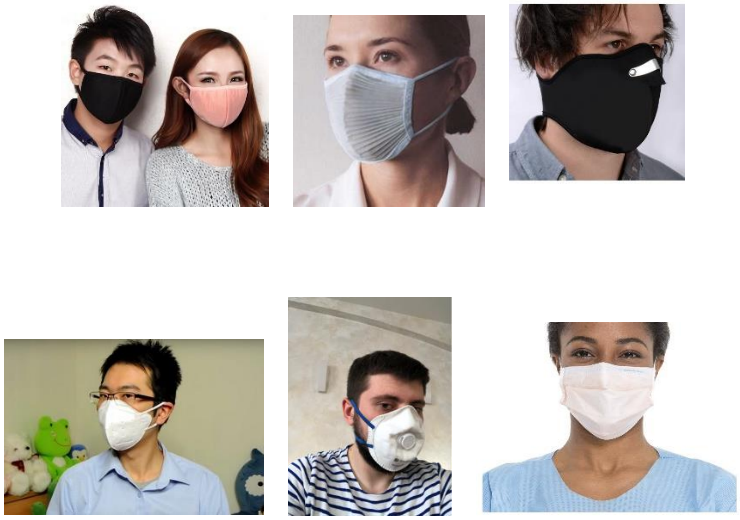


Fig 3.2 Sample of Masked Faces

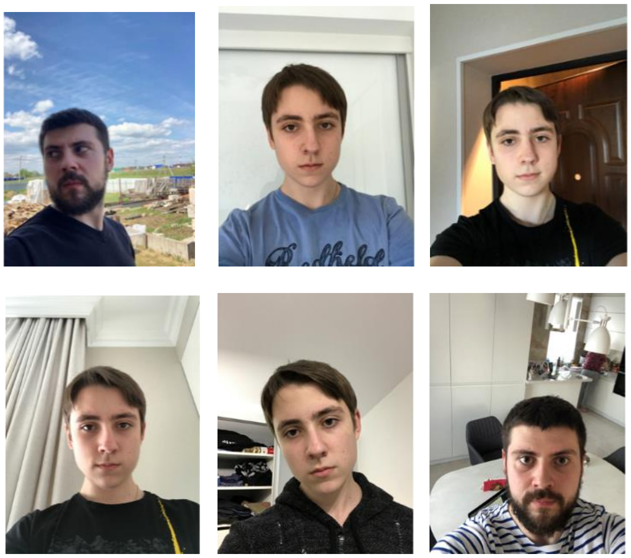


Fig 3.3 Sample of faces without mask

The dataset folder has two sub folder- with mask and without mask. we have split the dataset into such a manner that 80% of the with mask and without mask dataset is used for training and 20% went for testing. The class 0 represents without mask whereas 1 stands for with mask.

Test
20%
Train
80%
With mask/Without Mask dataset

Fig 3.4 Train and Test percentage

For VGG16 the dataset is already divided into 80-20 train-test while in the other models there is a code to categorize the data and have used train-test split for partitioning.

3.2 Methodology

In this paper, we are using CNN architecture to predict a person with or without mask. Three model have been used here. The technique of deep learning or neural network is the core reason of achievement of face mask detection. Our face mask detector model includes three phases- training, testing and detection. The dataset is fed into the training phase of the model, and the model is executed. The trained model is loaded, faces in photos and video streams are identified, and the area of interest is identified. Finally, the face mask detector is used to classify the pictures or faces in the video streams as wearing a mask or not wearing a mask. We have executed the model through python programming language. We have used “Google Colab” over the “Jupyter Notebook” because the library functions like keras, open CV, tensorflow, classification report, PyTorch are pre-installed here as well as dedicated GPUs and TPUs are present. However, we need to access the camera of computer/laptop to detect face mask in real time so we have applied a code in Google Colab for the access. The camera access code is different for Goggle Colab and PyCharm as we have implemented MobileNet in both.

Type	With mask	Without mask
Total Train Image	800	800
Total Test Image	200	200
Total Image	1000	1000

Table 3.1 Test Train split

As images are 2D vector, we have imported Conv2D, MaxPool2D, AveragePooling2D. In order to prevent overfitting, we have to dropout some data. Finally, after we have an ideal metrics, we flatten them, and this flattening layer must be transmitted through the dense layer, which is a Multi-layer Perceptron or Dense Neural Network. Aside from that, we used the BatchNormalization normalizing approach.

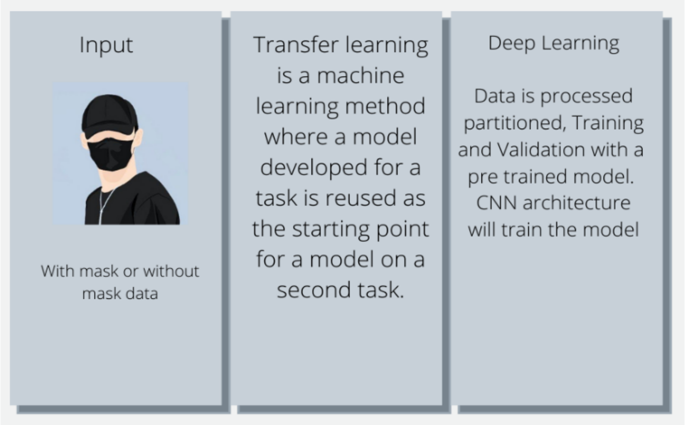

Input
With mask or without mask data
Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.
Deep Learning
Data is processed partitioned, Training and Validation with a pre trained model. CNN architecture will train the model

Fig 3.5 Model Architecture

Input size for the 3 models is different because of their architectural pattern. Also, the CNN based layers that are used in the architecture is also different among three of them. The flowchart (Fig 3.6) represents the basic method of how the model works.

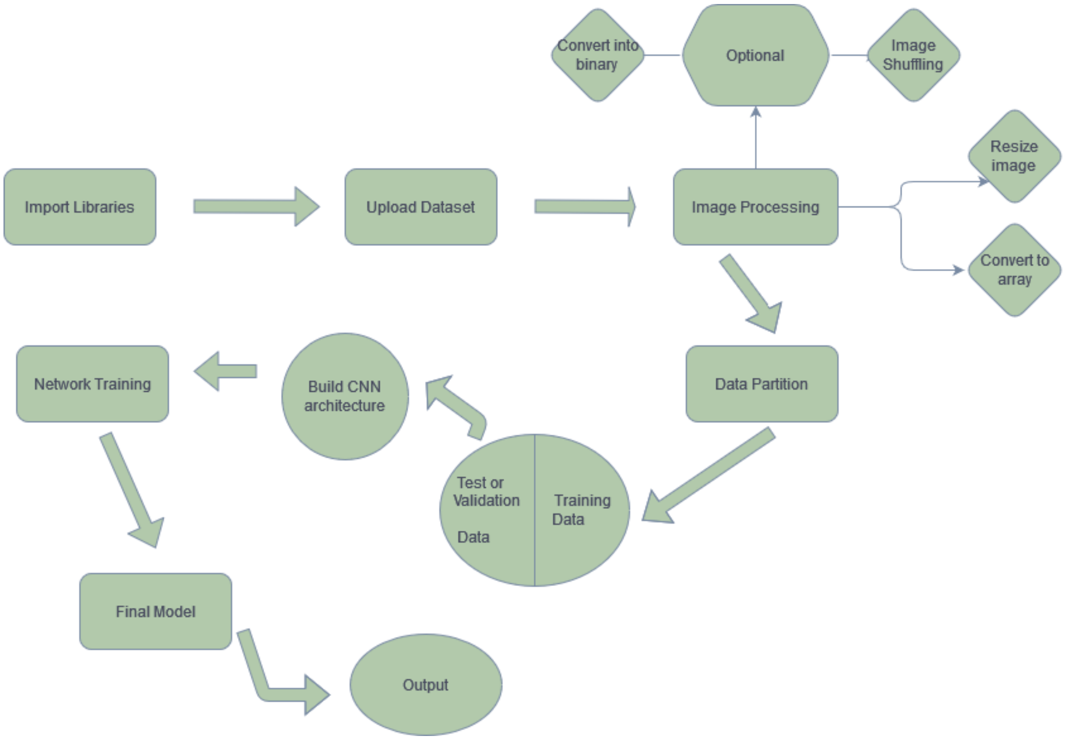


Fig 3.6 Methodology flowchart

For the image processing part, we have shuffled the datasets among them just to check if with mask always predicts 1 or not in AlexNet. On the other hand, in MobileNet, we have converted the dataset into binary after labeling them. So, these two things are not mandatory. But may leave a small change on result.

3.3 Materials

3.3.1 TensorFlow

TensorFlow is an open-source machine learning framework that focuses entirely on deep neural networks. TensorFlow is a diverse and extensive set of libraries, tools, and community resources. It allows developers to create and deploy cutting-edge machine learning applications. The TensorFlow python deep-learning library was created by the Google Brain team for internal use first. Since then, the open-source platform's use in R&D and manufacturing systems has grown. TensorFlow is based on a few key principles. Tensors are the core building components of TensorFlow. A tensor is an array that represents numerous types of data in the TensorFlow python deep-learning framework. A tensor can have n dimensions, unlike a one-dimensional vector or array or a two-dimensional matrix. The shape represents dimensionality. A one-dimensional tensor is a vector; a two-dimensional tensor is matrix; and a zero-dimensional tensor is a scalar. Figure 3.7 shows various tensor dimensions. **[30]**

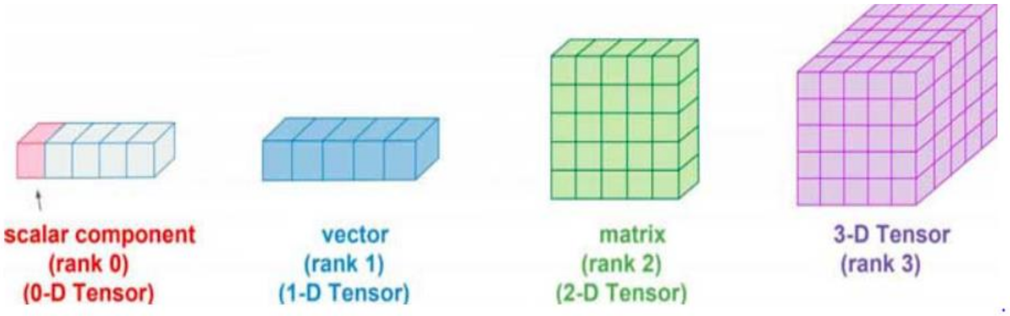


Fig 3.7 Tensor rank

3.3.2 Keras

Keras is a high-level deep learning API for creating neural networks developed by Google. It is developed in Python and aids in the creation of neural networks. It's modular, fast, and easy to use. Francois Chollet, a Google developer, created it. Keras does not handle low-level computation. It instead makes use of a library called "Backend." Keras allows you to switch between different back ends. Keras supports a number of frameworks, including TensorFlow, Theano, PlaidML, MXNet, and CNTK (Microsoft Cognitive Toolkit). Only TensorFlow has recognized Keras as its official high-level API among these five frameworks. Keras is a deep learning framework based on TensorFlow that includes built-in modules for all neural network computations. Simultaneously, the TensorFlow core API may be used to build custom computations with tensors, computation graphs, sessions, and so on. It gives users complete control and flexibility over their applications, as well as the ability to quickly implement ideas. **[31]**

3.3.3 OpenCV-python

OpenCV is a large open-source library for image processing, machine learning, and computer vision. python, c++, java, and other programming languages are among the languages supported by OpenCV. It can recognize objects, faces, and even human writing by analyzing efficient library for numerical operations. One of OpenCV’s goals is to provide a simple-to-use computer vision infrastructure that allows individuals to quickly create rather complex vision applications. Over 500 functions in the OpenCV library cover a wide range of vision topics, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, and robotics. Because computer vision and machine learning are frequently used together, OpenCV also includes a comprehensive machine learning library. **(GeeksforGeeks 2021.)**

3.3.4 Imutils and Matplotlib

Imutils is a package of convenience functions for OpenCV and Python 2.7 and 3 that make fundamental image processing tasks like translation, rotation, scaling, skeletonization, and presenting matplotlib images easier. Matplotlib is a popular 2D array plotting package written in Python. Matplotlib is a multi-platform data visualization software that uses numpy arrays and works with the scipy stack. John Hunter was the first to introduce it in 2022. One of the most significant benefits of visualization is that it allows users to see vast amounts of data in simple images. Line, bar, scatter, histogram, and more graphs are available in Matplotlib.

3.3.5 NumPy and SciPy

NumPy is the most important python package for scientific computing. It is a library that includes a multidimensional array object, derived objects (such as masked arrays and matrices), and a variety of routines for performing fast array operations, such as mathematical, logical, shape manipulation, sorting, selecting, basic linear algebra, basic statistical operations, random simulation, and more. Many other popular python packages, like as pandas and matplotlib, are compatible with NumPy. (NumPy 2022.). On the other hand, SciPy is another important python library for scientific and technical computing that is free and open source. It is a set of mathematical algorithms and utility functions based on the python numpy extension. It gives the user a lot of power by providing high-level commands and classes for manipulating and displaying data in an interactive python session. SciPy builds on NumPy, developers do not need to import NumPy if SciPy has already been imported. **(Great Learning Team 2020.)**

3.4 Deep Learning

Machine Learning is a subset of Artificial Intelligence, while Deep Learning is a subset of Machine Learning [32]. Deep learning tries to emulate the human brain, but it falls well short of its capabilities. It enables systems to cluster data and produce extremely accurate predictions. Deep learning is a machine learning technique that teaches computers to perform what people do naturally. It is inspired by the anatomy of the human brain. Deep learning is receiving a lot of attention these days, and for good reason. It's accomplishing accomplishments that were previously unattainable.

When we think of deep learning, the name "Neural Network" comes to mind first because it is such a vital component. Neural networks use a set of algorithms to simulate the human brain. Deep learning algorithms analyze data with a logical structure in order to reach conclusions that are comparable to those reached by humans. Deep learning achieves this by employing a multi-layered computational structure known as a neural network. The neural network's design is primarily inspired by the structure of the human brain. We use our brains to identify styles and classify different pieces of information, and neural networks can be trained to do the same thing with data. Separate layers of neural networks can also be thought of as a type of filter that works from the most extreme to the most extreme, increasing the possibility of detecting and producing an accurate result. We use neural networks to group and classify things. We can use neural networks to group or sort unlabeled data based on similarities between the samples in the data, or we can train the network on a labeled dataset to categorize the samples in the dataset into various categories **[36].** Deep neural networks can thus be considered components of broader machine learning programs involving reinforcement learning, classification, and regression methods. Deep learning and artificial neural networks are extremely powerful and unique in today's market. The fundamental advantage of deep learning over machine learning is the absence of the need for so-called feature extraction. Long before deep learning, flat machine learning techniques like Decision Trees, SVM, Nave Bayes Classifier, and Logistic Regression were used. Normally, these techniques can't be used to raw data like.csv files, images, or text. Feature Extraction, a preprocessing step, is necessary. Feature extraction is typically challenging and requires a deep understanding of the problem domain. The feature extraction process is not necessary for Deep Learning Artificial Neural Networks. The layers can directly and independently learn an implicit representation of the raw input. To conduct and optimize the feature extraction process, deep learning models require little to no manual work. While deep learning was first proposed in the 1980s, it has only lately become relevant for two reasons:

- Deep learning necessitates a large amount of labeled data. For example, the creation of self-driving cars involves millions of photos and thousands of hours of video.
- Deep learning necessitates a significant amount of computing power. The parallel design of high-performance GPUs is ideal for deep learning. This helps development teams reduce deep learning network training time from weeks to hours or less when used in conjunction with clusters or cloud computing.

3.4.1 How deep learning works:

Most deep learning approaches use neural network topologies, deep learning models are sometimes referred to as deep neural networks. The term "deep" refers to the amounts of hidden layers in a neural network. Traditional neural networks have only 2-3 hidden layers, however deep neural networks can have up to 150. Deep learning models are trained using large quantities of labeled data and neural network topologies that learn features directly from the data without the need for manual feature extraction. Neural networks are made up of layers of nodes, similar to how the human brain is made up of neurons. A single neuron in the human brain can receive millions of messages from other neurons. Nodes in neighboring layers are connected to nodes in this layer. The more layers a network contains, the deeper it is said to be. Three layers make up the deep learning model. The deep learning model consists of an input layer, one or more hidden layers, and an output layer.

Input Layer: Input nodes are used to represent all of the input variables. In the artificial neural network's workflow, the input layer is the first phase. After receiving data as input, the computer converts it to bits of binary data so that it can interpret and actualize the information. To be within the same range, input data variables must be either standardized or normalized.

Hidden Layer(s): In the hidden layer, all of the input variables are aggregated across one or more nodes. This essentially adds new features based on the provided inputs. In most cases, all input nodes are connected to all hidden layer nodes. We're now dealing with deep learning if our neural network contains more than one hidden layer. The non-linear processing units for feature extraction and transformation are performed by the layers here. It gradually develops the hierarchy concepts through learning. Each step of the hierarchy transforms the input data into a more theoretical and composite representation.

Output Layer: A prediction or classification is made in the output layer utilizing the nodes in the hidden layer. One output node is used for numerical prediction. C-1 nodes are used for classification, where C is the number of classes that can be created. The activation function transforms the integrated weighted input from the node into the node's activation function in a neural network output layer. Two of the most effective activation functions are relu and sigmoid. It's because the model relu employs is simpler to train and capable of generating superior results.

For each layer, we'll additionally see a Bias Node. A bias node works in the same way as a regression intercept. It enables us to change our learned model. Any zero or negative input would result in a zero output without it.**[43]**

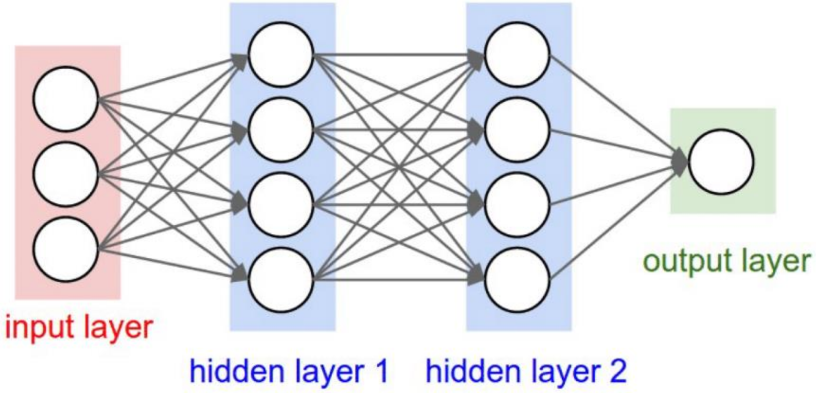


Fig 3.8 Deep learning architecture [43]

3.5 Convolution Neural Network

Artificial neural networks are becoming increasingly used for processing unstructured data, such as images, text, audio, and speech. Convolutional neural networks (CNNs) work well with such unstructured data. When there is a topology related to the data, convolutional neural networks uncover crucial properties. with it. In terms of architecture, CNNs are similar to multi-layer perceptions. CNN employs this technique. By establishing local connection constraints between neurons in adjacent layers, local spatial correlation can be achieved. The heart of convolutional neural networks is data processing via the convolution operation. When one signal is convolutional with another, a third signal is created that may contain additional information in comparison to the original signal.[22]

Deep learning has many branches and convolutional neural network is one of them. It was regarded one of the most powerful tools in the previous several decades and became popular in literature because a great amount of data can be handled. Recently deeper hidden layer has begun to outperform traditional approaches in several disciplines, in particular pattern recognition. CNN is specific sort of profound education architecture inspired by life's visual system. It is a new neural network or sometimes called multi-layered network. The CNN is ideal for the many domains of computer vision and processing of natural language. A detailed overview on all the essential components of CNN is the major emphasis of this section. It also provides an overview of CNN's basis, recent developments in CNN and many significant areas of application. There are several layers of CNN, including convolutional layer, nonlinearity layer, layer of pooling, and fully connected layer. This article proposes a CNN model for the detect of classes of facemask. So, some processes need to be understood before using the CNN algorithm. For instance, the input image is initially stored as a matrix and holds pixel values. It is classified by feeding in multi-level perception. Based on the relevant filters, the CNN can obtain time information. It is understandable to learn a high-dimensional picture decreases dimensionality to improve prediction accuracy without the need for data loss.[23]

3.5.1 CNN Architectures:

Convolutional and pooling (or subsampling) layers are organized into modules in CNN designs of various forms and sizes. Like a regular feedforward neural network, these modules are followed by one or more fully linked layers. Modules are typically piled on top of one another to build a deep model. The network receives an image directly and processes it through multiple convolution and pooling rounds. These operations' output is subsequently fed into one or more fully connected layers. The output layer receives data from the layers above it, performs calculations using its neurons, and computes the result. Despite the fact that this is the most often used base architecture in the literature, several architecture modifications have been proposed in recent years to improve picture classification accuracy or reduce computation costs. The relationship from input layer to output layer is shown in Figure 3.9[24]

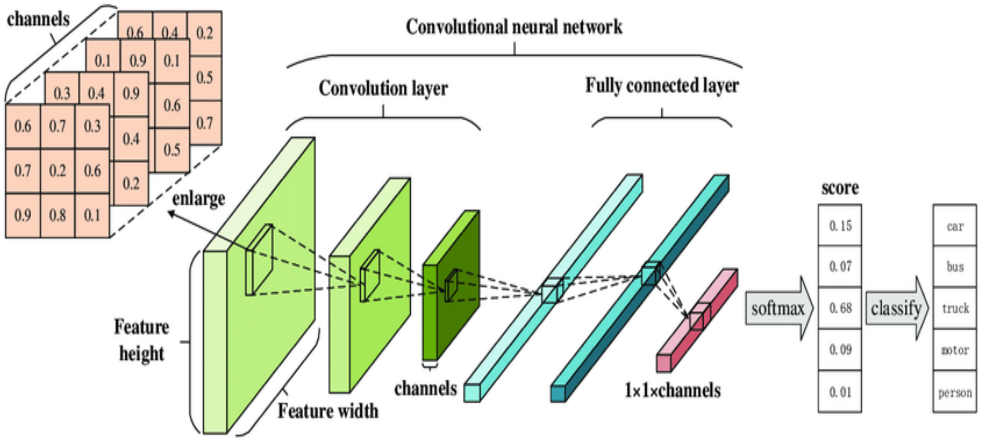


Fig 3.9 CNN architecture (Hands-On Machine Learning with Scikit-Learn and TensorFlow)

3.5.2 Workflow of CNN:

Convolutional neural networks outperform other neural networks using visual, speech, or audio signal inputs due to their superior performance. The three most common types of layers are convolutional layers, pooling layers, and fully connected layers. The convolutional layer is the initial layer of a convolutional network. After convolutional layers, additional convolutional or pooling layers can be added, but the fully connected layer is the last layer. With each layer, the CNN grows more sophisticated, identifying bigger portions of the image. The first layers focus on the most fundamental aspects, such as colors and borders. As the visual data passes through the CNN layers, it starts to recognize larger pieces or features of the item, eventually recognizing the target object.

3.5.3 Convolutional layer

The convolutional layer is the most important component of a CNN because it is where most of the computation takes place. It requires, among other things, input data, a filter, and a feature map. A color image made up of a 3D matrix of pixels, for example, is supplied to the input layer. This means that the input will have three dimensions: height, width, and depth, all of which correspond to a picture's rub color space. A feature detector, also known as a kernel or a filter, will check for the presence of the feature across the image's receptive fields. This method is known as convolution. A two-dimensional (2-D) weighted array that represents a piece of the image is used as the feature detector. The size of the receptive field is also affected by the filter size, which can vary in size. The filter is then applied to a segment of the image, and the dot product between the input pixels and the filter is then shifted by a stride, and the process is repeated until the kernel has swept across the entire image. The ultimate result of a succession of dot products from the input and the filter is a feature map, activation map, or convolved feature.

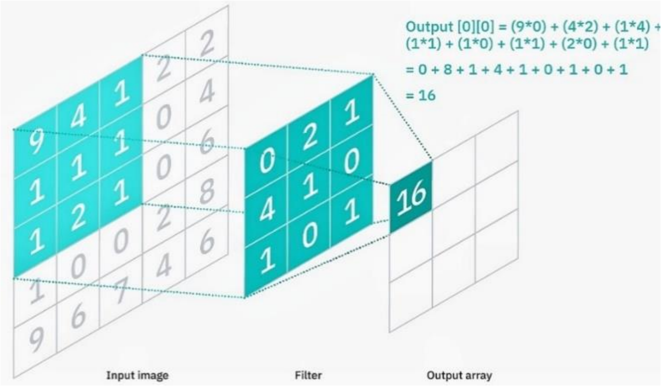


Fig 3.10 Visual representation of a convolutional layers

As shown in Fig 3.10 each output value in the feature map does not have to correspond to each pixel value in the input image. It only needs to be connected to the receptive field, where the filter is used. Because the output array does not have to map exactly to each input value, convolutional and pooling layers are commonly referred to as “partially connected” layers. The weights of the feature detector stay constant as it moves over the image, a technique known as parameter sharing. Some parameters, such as weight values, vary during training due to backpropagation and gradient descent. Three hyperparameters that determine the output volume size must be defined before the neural network can be trained. Among these are the number of filters, stride, and zero padding. The amounts of filters utilized has an impact on the output depth. The stride of the kernel refers to the number of pixels it traverses across the input matrix. A larger stride produces a lesser output, despite the fact that stride values of two or more are unusual. Zero-padding is used when the filters do not fit the input image. Outside of the input matrix, all members are set to zero, resulting in a larger or equivalent output.

3.5.4 Pooling layer:

The features of images with precise positions are brought out by convolutional layers. The feature maps will be altered if the placements vary even slightly for whatever reason. To solve this issue, down sampling must be performed the output of each convolutional layer. Through a sub sampling method that supports local space invariance, this stage allows the initial huge dimension of picture representation to be reduced**. [25]**

This layer’s purpose is to offer spatial variance, which basically implies that the system will be able to recognize an item even if its look changes. **[26].** The activation maps are down sampled by a factor of two in both dimensions using the pooling operation. Max pooling is the most prevalent method for doing this, which merges pixels in adjacent 2x2 cells by taking the highest value among them. Pooling has the virtue of allowing us to reduce the amount of data we have without sacrificing too much information, as well as creating some invariance to translational shift in the original image. Because there are no weights or characteristics to learn, the operation is also relatively inexpensive. After using 27 convolution layer, pooling layer is added. It is applied to feature maps. Basically, it should be smaller than input image. Basically 2×2 filter is used in pooling layer with 2 stride.it cuts image size by two. It implies that each dimension will be half. Sometimes it implies 6×6 filters. [27]. In CNN, two common functions are used, average polling and maximum pooling.

3.5.5 Average Pooling

It is second most used pooling in CNN. Basically, it implies the average of all value and put to the output. The output will be average number correspondingly different color shaded region. Each region will get one output. In figure 3.6, there are 4×4 matrix and uses 2×2 pooling layer. So, the output size will be 2×2. In input layer, there are divided 4 different regions. Each of the region have 4 different number. In average pooling, those number are calculated and get the average number. Then we put it in the output matrix [28]

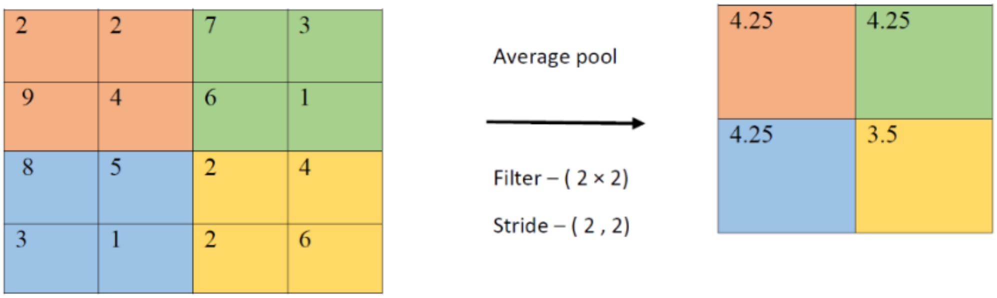


Fig 3.11 Average Pooling

3.5.6 Maximum Pooling (or Max Pooling)

Max pooling is most favorite pooling layer in CNN. Basically, it implies the biggest value and put to the output. The output will be biggest number correspondingly different color shaded region. Each region will get one output. In figure 3.4, there are 4×4 matrix and uses 2×2 pooling layer. So, the output size will be 2×2. In input layer, there are divided 4 different regions. Each of the region have 4 different number. In max pooling, we collect large value. Then we put it in the output matrix. [28]

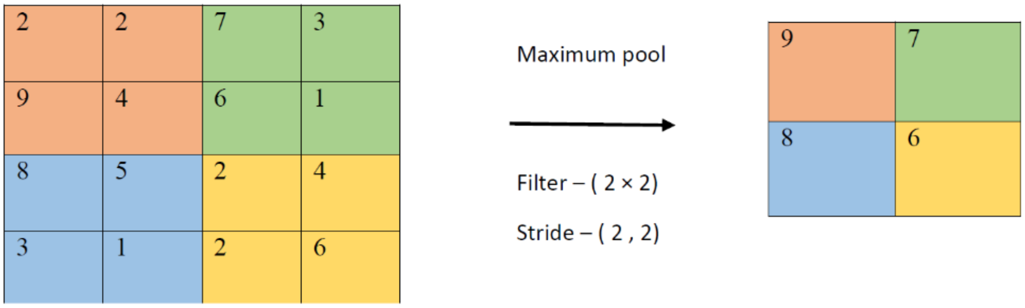


Fig 3.12 maximum pooling

3.5.7 Fully connected layer

The name of the completely connected layer is self-explanatory. Fully linked layers in a neural network are those in which all of the inputs from one layer are connected to each activation unit of the following layer. As previously stated, the pixel values of the input image are not directly connected to the output layer in partially connected layers. In the completely connected layer, each node in the output layer is connected directly to a node in the previous layer. This layer categorizes the features extracted by the previous layers as well as the filters applied to them. Fully connected layers often employ a softmax activation function to produce a probability ranging from 0 to 1, whereas convolutional and pooling layers typically use relu functions to classify inputs.

3.5.8 Different architectures in CNN:

CNN is the most well-known and widely used algorithm in deep learning. The fundamental advantage of CNN over its predecessors is that it detects relevant elements without the need for human intervention. CNNs have been widely used in a variety of domains, such as computer vision, audio processing, and facial recognition, among others. CNNs are like traditional neural networks in that their structure is inspired by neurons in human and animals' brains. There are variety of CNN architectures available, all of which have contributed to the development of algorithms that enable AI today and will continue to do so in the future.

CNN plays an important part in calculating vision connected models in recognizing patterns, as a result allure less computation cost and further the capability of spatial distillation. CNN takes advantage of spiral portions to combine accompanying the basic images in consideration of erase top-level countenance. The kickoff network that is projected in permits the network to receive familiar with the join of kernels. Planning to build a good Convolutional Neural Network construction really remnants as a primary asking. To draw up a lot further interconnected system, K. He and others. proposed Residual Network (ResNet) that can fool traits preparation from the past tier. As item locators are generally sent on lightweight or some entrenched device, place the calculating property are intensely restricted, Mobile Network (MobileNet) is proposed. The MobileNet uses depth-aware convolution to eliminate highlights and channelized convolutions to adjust channel numbers, resulting in a far lower computational cost than networks that use normal convolutions. In Fig 3.13 we have shown a Schematic Diagram for Basic Convolution Neural Network.

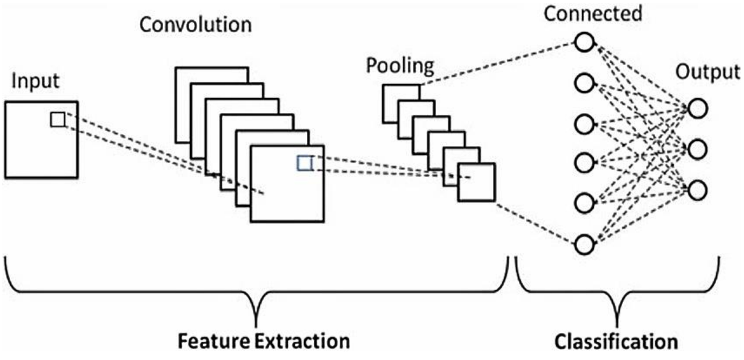


Fig. 3.13 Schematic Diagram for Basic Convolution Neural Network

3.6 Optimization, Activation Function and other layers

3.6.1 Dense layer

The dense layer is a simple layer of neurons in which each neuron receives input from all of the neurons in the preceding layer. Dense Layers are used to identify images based on convolutional layer output. The dense layer conducts matrix-vector multiplication in the background. The values in the matrix are really parameters that backpropagation may be used to train and update. The dense layer produces an 'm' dimensional vector as its output. As a result, the dense layer is mostly employed to alter the vector's dimensions.

3.6.2 Dropout

Dropout is a neural network regularization strategy that assists in reducing interdependent learning among neurons. Because the bulk of the parameters are taken up by a fully linked layer, neurons acquire co-dependency on one another during training, reducing each neuron's distinctive power and resulting in over-fitting of training data. Dropout randomly picks certain nodes and causes them to stop working in the model. The number of nodes is effectively reduced. As a result, variance falls, minimizing overfitting and boosting model accuracy **[39]**

3.6.3 Flatten

The flattening step of a convolutional neural network is simple. It converts the pooled feature map into a one-dimensional vector obtained by the pooling phase. The feature map is flattened into a one-dimensional matrix, which is used as the input layer in the CNN's artificial neural network**. [40]**

3.6.4 Optimization

Optimizers are methods or techniques for decreasing the size of a loss function or increasing the quality of production. Optimizers are mathematical functions that determine the optimum solution depending on the model's learning parameters, such as Weights and Biases. Optimizers can help with knowing how to change the weights and learning rate of a neural network in order to decrease losses **[37]**

Adam is a well-known optimization technique for deep learning. Adam is a learning rate method that estimates individual learning rates based on a variety of aspects in the learning process. Adam gets its name from the fact that it uses estimations of first and second moments of gradient to change the learning rate for each weight of the neural network. The Adam optimizer offers a lot of benefits that contribute to its popularity. It has been established as a standard for deep learning articles and is suggested as the default optimization method **[38].**

3.6.5 Activation Function

The activation function specifies a neuron's output in response to a single or numerous inputs. Forward propagation is the output of the activation function to the next layer, input and output layers, and in deep neural networks, the next hidden layer) (information propagation). A neural network's non linearity transformation is referred to as this.

The use of the ReLU activation function in the hidden layers might help deep neural networks train faster. For deep neural networks, the rectified linear unit is now the standard activation function. Using the ReLU activation function, the vanishing gradient problem may be avoided. This is why the ReLU activation function can boost the deep neural network's learning speed **[41]**

3.7 Transfer Learning

One of the most often used approaches for computer vision tasks such as classification and segmentation is transfer learning. It's the process of sharing weights or knowledge gained from one problem to solve other problems that are similar. When the application domains are closely related, transfer learning reduces training time. The following are two common methods for performing transfer learning.

Using a pre-trained model: A model that has been trained on a large-scale benchmark dataset is referred to as a pre-trained model. Several pre-trained models, such as ResNet, MobileNet, and GoogleNet, have been trained using the Imagenet dataset. These algorithms take color photos as input and classify them into 1000 different categories. When we want to categorize any class defined in the imagenet, this model provides the best accuracy.

Defining a custom output layer: In this way, the pretrained model is considered without the feature extractor output layer. Using a custom output layer, these features can be used for the required categorization task. Finally, to improve classification results, the custom model must be trained.

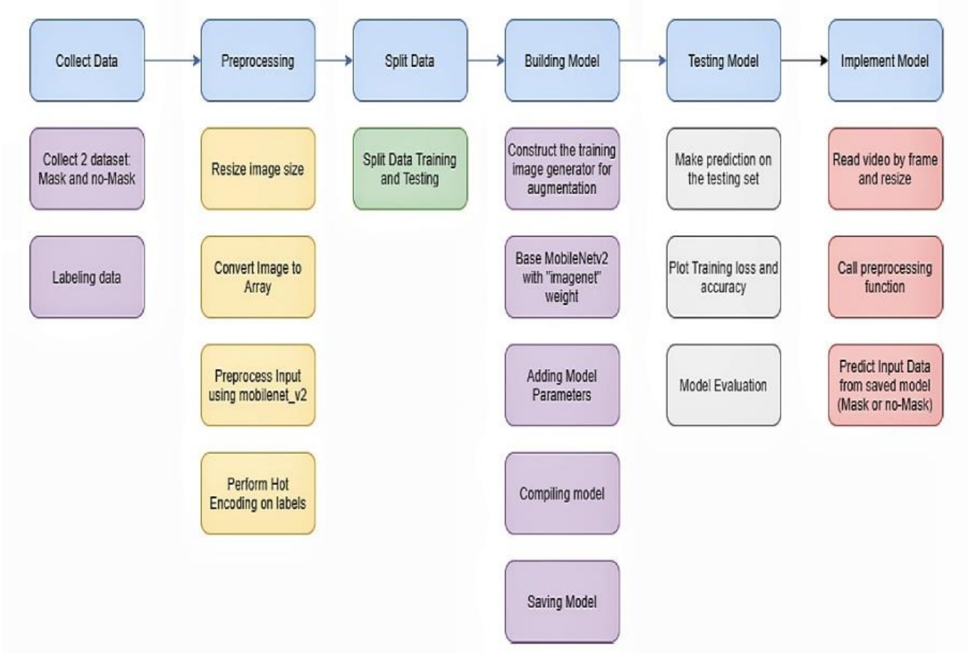

Collect Data
Preprocessing
Split Data
Building Model
Testing Model
Implement Model
Collect 2 dataset: Mask and no-Mask
Labeling data
Resize image size
Convert Image to Array
Preprocess Input using mobilenet_v2
Perform Hot Encoding on labels
Split Data Training and Testing
Construct the training image generator for augmentation
Base MobileNetv2 with "imagenet" weight
Adding Model Parameters
Compiling model
Saving Model
Make prediction on the testing set
Plot Training loss and accuracy
Model Evaluation
Read video by frame and resize
Call preprocessing function
Predict Input Data from saved model (Mask or no-Mask)

Fig 3.14 Steps in building the model

3.7.1 AlexNet

AlexNet, a convolutional neural network was introduced by Alex Krizhevesky and his team in 2012. It is broader and deeper than the LeNet model and successful against hard ImageNet challenges to detect the objects and recognize it. The discovery of AlexNet brought breakthroughs in the area of machine learning. The architecture of AlexNet is depicted in Fig 3.14. To perform max pooling and convolution, the first convolutional layer use 96 filters and the size of the filters are 11*11. The max pooling is accomplished with 3*3 fiters where a size of 2 stride is used as well. In the second layer, the same operations accomplished with 5*5 filters. In terms of the 3rd, 4th and 5th convolutional layers, 384, 384 and 296 featured maps are used respectively with 3*3 filters. Relu is the activation function is used here. With the dropout, 2 fully connected layers are also used with a Softmax activation function at the end. In this architecture, two networks with comparable structures are trained along with the same number of featured maps. The model introduced two new concepts of machine learning and they are Local Response Normalization (LRN) and dropout. LRN can be implemented in two different ways. The first way of LRN applies on the feature map or single channel and it has an N*N patch that is chosen from the feature maps. By depending on the value of neighbors, normalization is applied. In the second way, LRN can be used across several featured maps or channels.

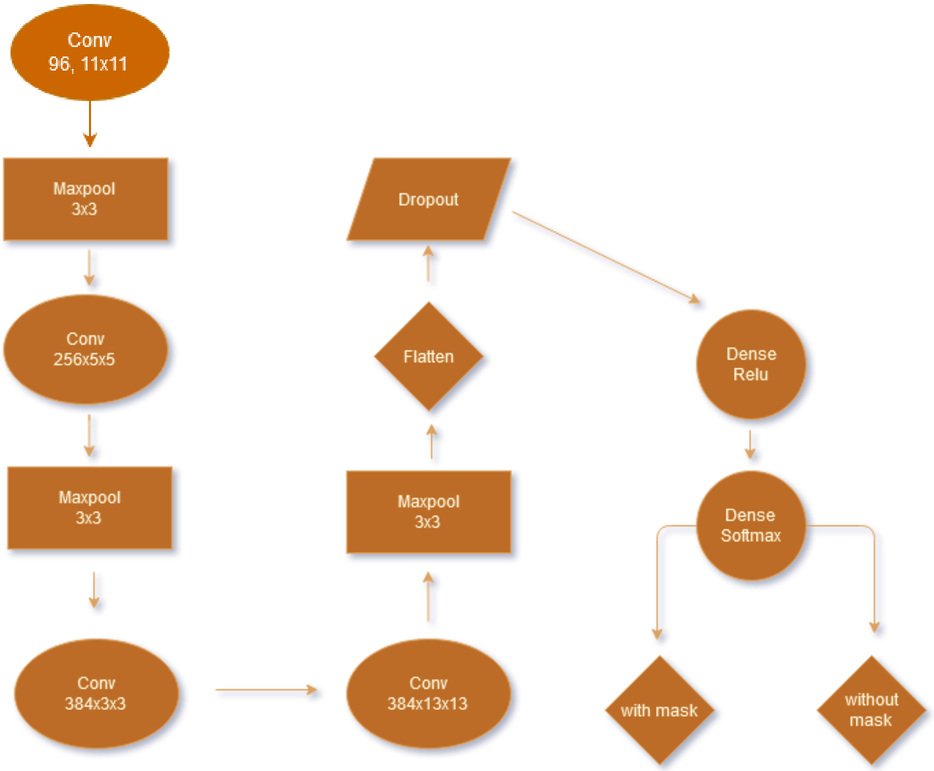


Fig 3.15 AlexNet architecture

For the first layer of the ImageNet dataset, the amounts of parameters for the model can be determined. In the first layer, as the inputs there are 4 strides with the kernel size of 224*224*3 and the output size of the first layer is 55*55*96 or 290400 neurons with 364 weights. So, for the first convolutional layer, there are 290400*364 parameters in this model [15]

Layer	Filter Number and size	Size of feature map
Conv Stride 4	96 11x11	55x55x96
Max Pool Stride 2	3x3	27x27x96
Conv 2 Stride 1, padding 2	256 5x5	27x27x256
Max Pool Stride 2	3x3	13x13x256
Conv 3 Stride 1, padding 1	384 3x3	13x13x384
Conv 4	384	13x13x384

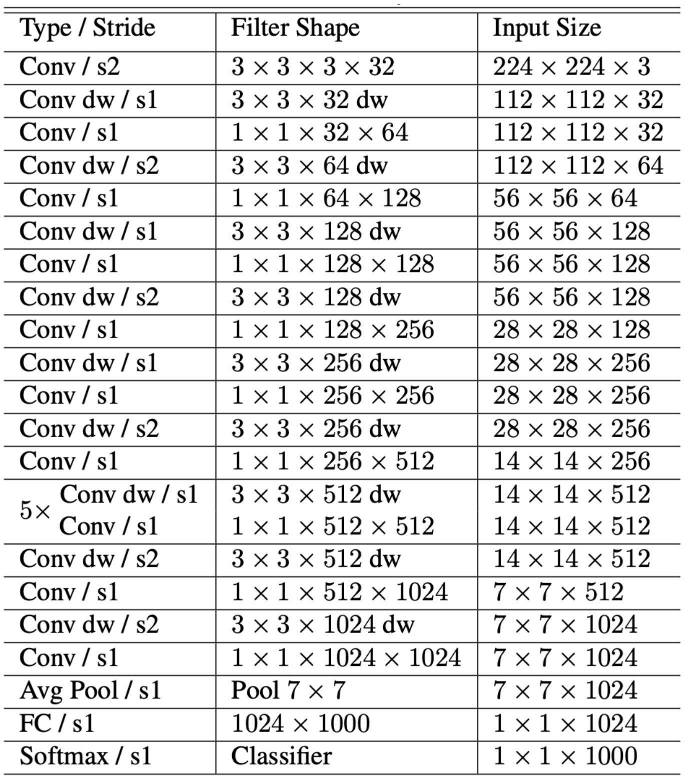
Stride 1, padding 1	3x3	
Max pool Stride 2	3x3	6x6x256
Dropout		6x6x6256

Table 3.2 AlexNet Model

3.7.2 MobileNetV2

In real-world applications, MobileNets are a CNN design that is both efficient and portable. MobileNets are CNNs that can fit on a mobile device and classify photographs or detect objects with low latency. They're usually relatively small CNN architectures, which makes them easy to run in real time on embedded devices like smartphones and drones. The approach has been tested on CNNs with 100-300 layers and has outperformed alternative architectures like as VGGNet. In real-world instances of MobileNets CNN architecture, CNNs embedded in Android phones run Google's Mobile Vision API, which can automatically recognize labels of popular things in images**. [33]**

MobileNetV2 is a sophisticated image classification software. MobileNetV2, a lightweight CNN-based deep learning model, uses TensorFlow to give picture weights. The MobileNetV2 foundation layer is removed first, followed by the addition of a new trainable layer. Our photographs are analyzed by the model, which extracts the most important aspects. In MobileNetV2 [34], there are 19 bottleneck layers. We chose OpenCV as the foundation model, which is built on the ResNet-10 architecture **[34].** OpenCV's Caffe model is used to detect the face and mask from a picture and a video stream. The result face recognized image is sent to the mask detecting classifier. It makes mask detection in video streaming faster and more precise. Overfitting is a common problem in machine learning. Our model was overfitted with the dataset, thus the Dropout layer was utilized to ignore it. We were able to do away with the base layer by using MobileNetV2 (include top=False). Resized images In our trainable model, we apply an average pooling procedure with 128 hidden layers (7,7). Relu is utilized in the secret layer, and SoftMax is used across the linked layer. We used a learning rate of 0.01 to improve accuracy. The Adam stochastic gradient descent approach helps the model understand visual features. Table 3.2 illustrates the MobileNetV2 working layer **[34]**



Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Table 3.3 MobileNetV2 Architecture

3.7.3 VGG-16

Another transfer learning and deep learning model is VGG-16. This network was developed by Simonyan and Zisserman. Back-to-back convolutional layers form the architectural design. A max-pooling layer is then added after that. Then there are two convolutional layers back-to-back and a max-pooling layer. Three convolutional layers are then added, followed by a pooling layer. Three convolutional layers follow, followed by a max-pooling layer that repeats twice. Three thick layers complete the construction. The output layer, containing three neurons, is the final of these dense layers, signifying the three classes of our classification job **[35].**

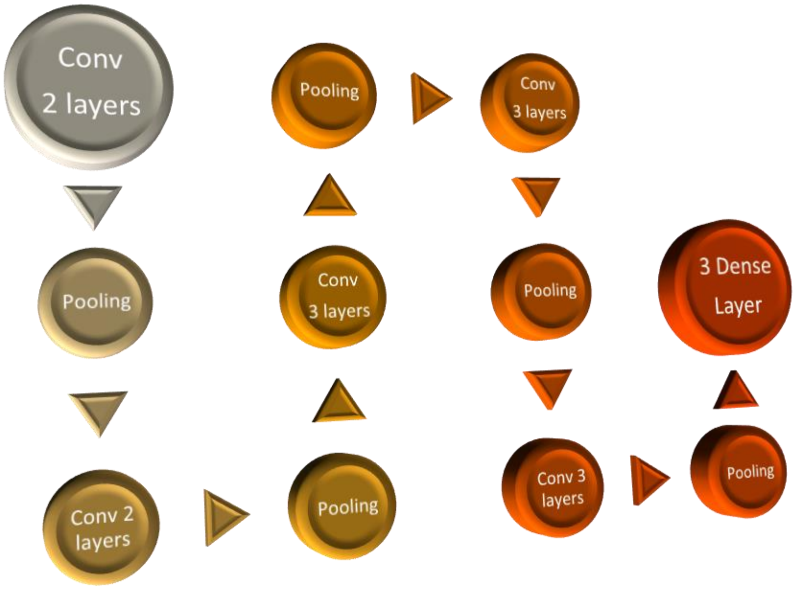


Fig 3.16 VGG16 architectural layers

Layers	Size and dimension
First 2 convolution layer. Stride 1 Padding 1 64 channel	Kernel 3x3 (224x224x64)
Max Pooling Stride 2	2x2 112x112x64
3,4 convolution layer 128 channel	Kernel 3x3 112x112x128
Max Pooling Stride 2	2x2 56x56x128
5,6,7 Convolution	3x3 kernal 56x56x256
max Pooling stride 2	2x2 28x28x256
8,9,10 convolution	3x3 kernal 28x28x512
Max pooling Stride 2	2x2 14x14x512
11,12,13	3x3 kernal

convolution	14x14x512
Max Pooling Stride 2	2x2 7x7x512

Table 3.4 VGG16 Model

3.7.4 Proposed Model

Data augmentation, dropout, normalization, and transfer learning are all concepts used in the proposed research. This approach can be employed in places like hospitals, malls, transportation hubs, restaurants, and other places where surveillance is required CNN's implementation. The proposed model has 11 convolutional layers with 4 average pooling layers. Also, there is 6 dense layer and a flatten layer with dropout. Input shape is 224x224x3.

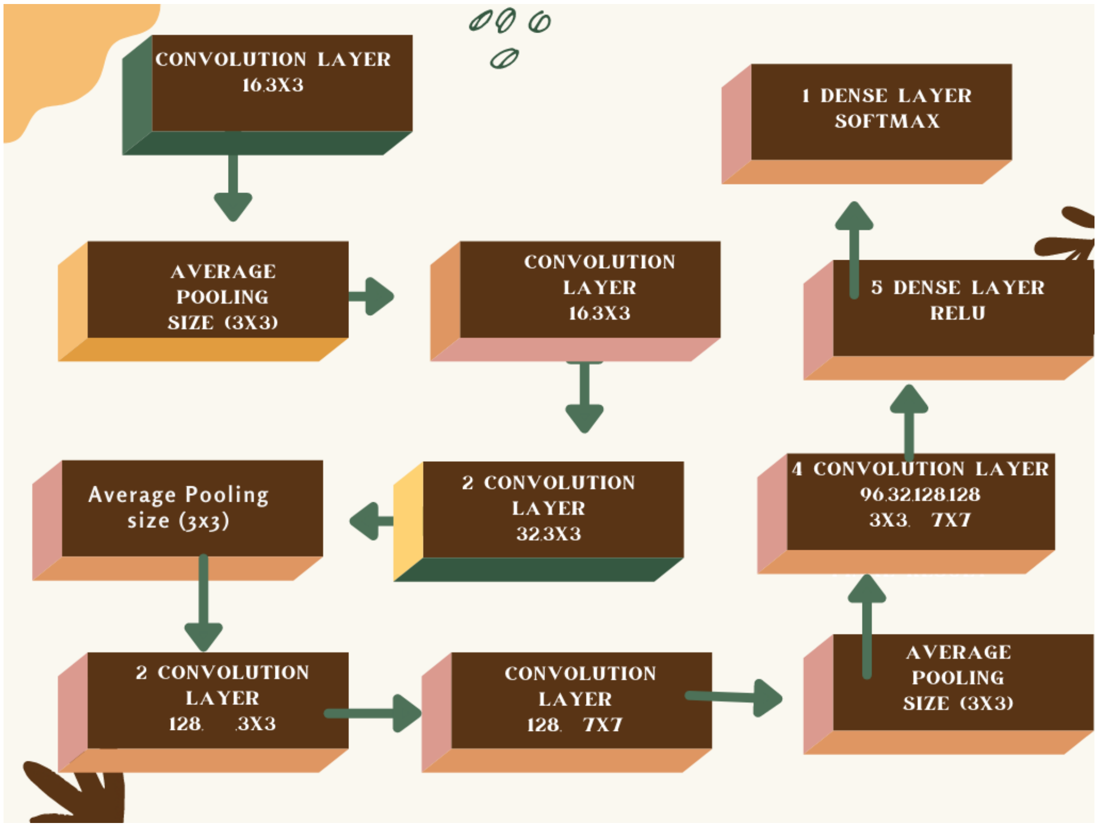


Fig 3.17 Graphical Structure of Proposed Model

Convolution layer have the filter size of 16,32,128 and 96, stride is (1,1) but the kernel size is either (3,3) or (7,7). Relu has been used as the activation function for the convolutional layers. at the very last dense layer the activation function is Softmax as the augmentation is for multiclass categorical data elsewise other dense layers has the Relu activation function.

Layers	Size and Dimension	Activation Function	Padding
Convolutional	16 , 3x3	Relu	valid
Average Pool (2x2)	3x3		
Convolutional	16 , 3x3	Relu	valid
2 Convolutional	32 , 3x3	Relu	Valid Same
Average Pool (2x2)	3x3		
2 Convolutional	128 , 3x3	Relu	Valid Same
Convolutional	128 , 7x7	Relu	Valid
Average Pool (2x2)	3x3		
Convolutional	96 , 3x3	Relu	Valid
Convolutional	32, 3x3	Relu	Same
Convolutional	128, 3x3	Relu	Valid
Convolutional	128, 7x7	Relu	Same
Average Pool (2x2)	3x3		
5 Dense		Relu	
Dense		Softmax	

Table 3.5 Propose Model CNN layer table

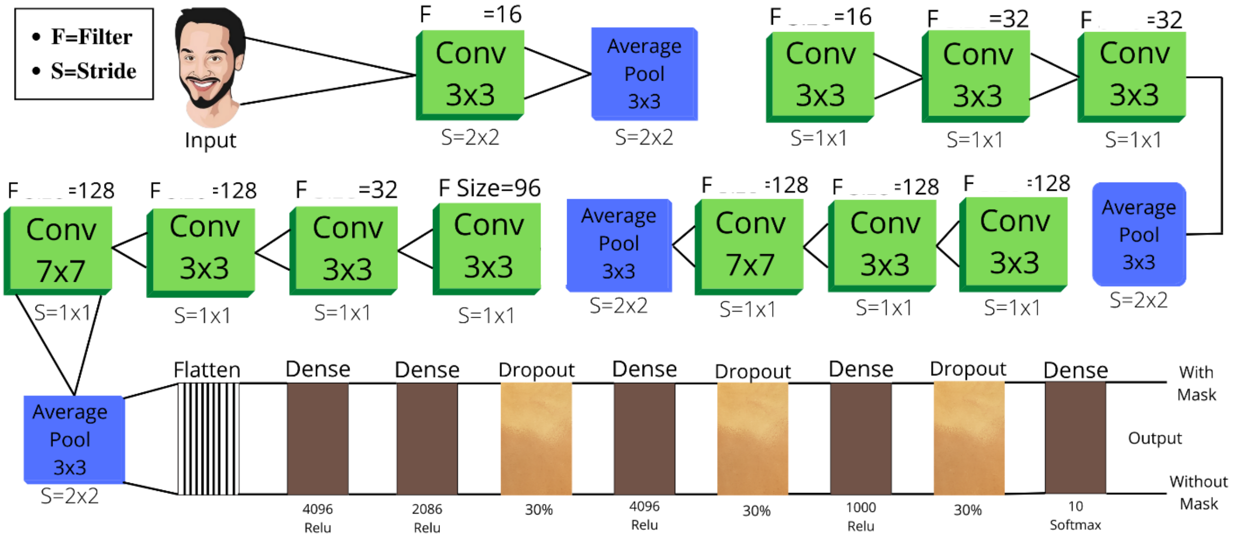


Fig 3.18 Detailed Architecture of Proposed Model

CHAPTER FOUR

Experimental Result and Analysis

The result section includes the train-test accuracy, confusion matrix, classification report for each of the used models.

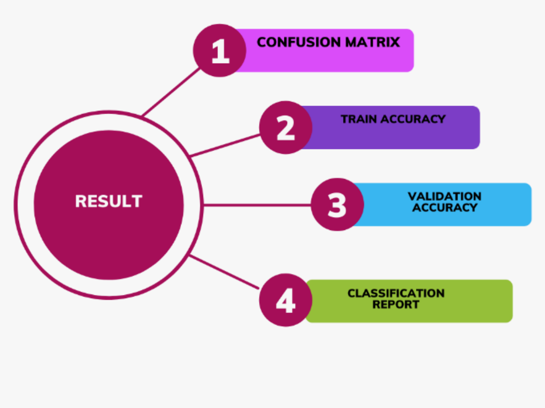


Fig 4.1 Parts of Result

AlexNet has been implemented through 3 different techniques.

- Model Fit without augmentation
- Model Fit with Augmentation and Change of Parameters
- Model Fit with Augmentation

Each the implemented way provides different accuracy for both validation and training. The highest accuracy is provided by the second implemented process. Not only for AlexNet but also for the MobileNetV2 architecture 2 different process of implementation is applied in this paper. The difference between two process is

- Image load and labeling
- Loss Function

We have used sparse categorical cross entropy and binary cross entropy loss function. Sparse categorical is used for multiclass and binary cross entropy is used for binary output.

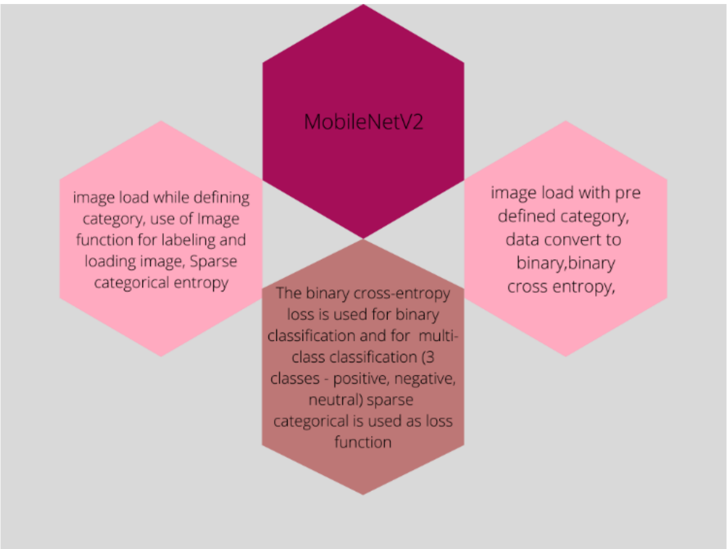

MobileNetV2
image load while defining category, use of Image function for labeling and loading image, Sparse categorical entropy
image load with pre defined category, data convert to binary,binary cross entropy,
The binary cross-entropy loss is used for binary classification and for multi-class classification (3 classes - positive, negative, neutral) sparse categorical is used as loss function

Fig 4.2 Two categorical functions for MobileNetV2

4.1 Training and Validation Accuracy

When training a machine learning model, overfitting is one of the most important things to avoid. This occurs when a model fits the training data well but is unable to generalize and generate correct predictions on new data. To determine if the model is overfitting, data scientists employ the cross-validation approach, in which they divide their data into the training set and the validation set. The training set is used to train the model, whereas the validation set is used exclusively to evaluate the performance of the model. Metrics on the validation set allow you to evaluate the quality of your model. Like how well it can generate predictions based on data. Consequently, train loss and train acc represent loss and accuracy on the training set, whereas val loss and val acc represent loss and accuracy on the validation set.

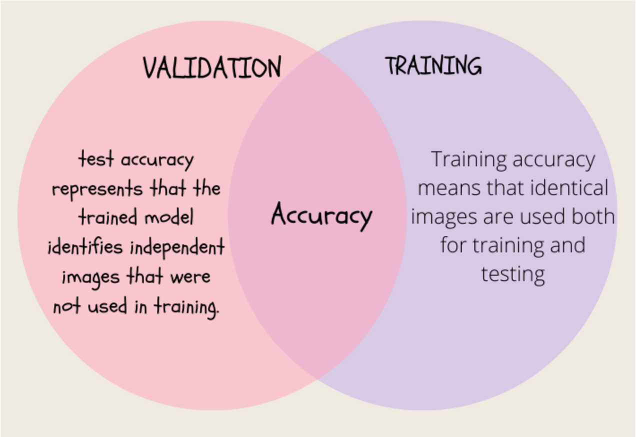

VALIDATION
TRAINING
test accuracy represents that the trained model identifies independent images that were not used in training.
Accuracy
Training accuracy means that identical images are used both for training and testing

Fig 4.3 Type of Accuracy

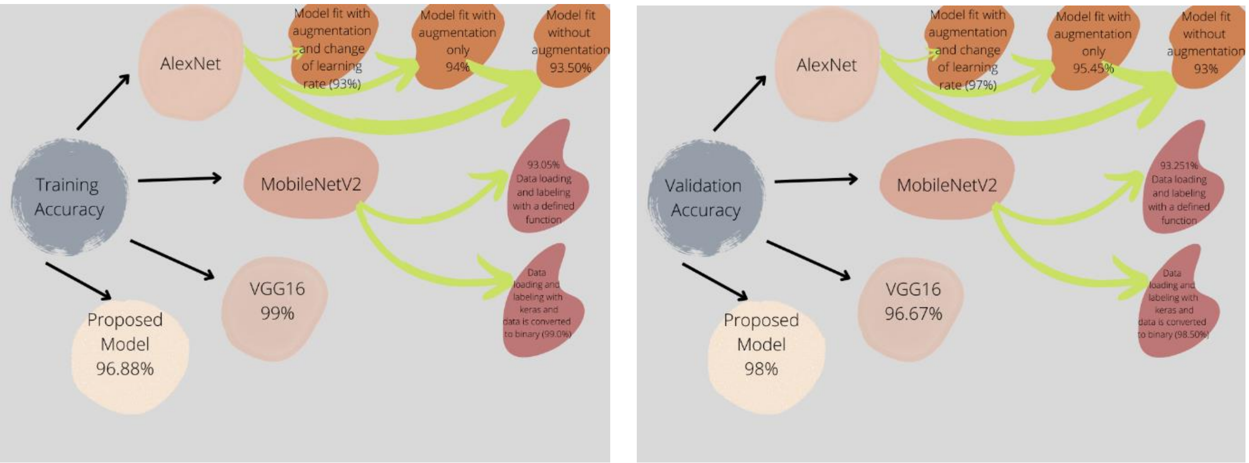

Training Accuracy
AlexNet
Model fit with augmentation and change of learning rate (93%)
Model fit with augmentation only 94%
Model fit without augmentation 93.50%
MobileNetV2
93.05% Data loading and labeling with a defined function
Data loading and labeling with keras and data is converted to binary (99.0%)
VGG16 99%
Proposed Model 96.88%
Validation Accuracy
AlexNet
Model fit with augmentation and change of learning rate (97%)
Model fit with augmentation only 95.45%
Model fit without augmentation 93%
MobileNetV2
93.251% Data loading and labeling with a defined function
Data loading and labeling with keras and data is converted to binary (98.50%)
VGG16 96.67%
Proposed Model 98%

Fig 4.4 Training and Validation Accuracy of all models

In Fig 4.5, we have shown Model training accuracy/loss curves. Parameters with a learning rate (initial) of INIT_LR = 1e-4, batch size BS = 32 and the number of epoch EPOCHS = 27. The graphs of loss nearly tended to zero and the graphs of accuracy showed that after 30 training epochs, the model maintained a high accuracy 98.50% without overfitting.

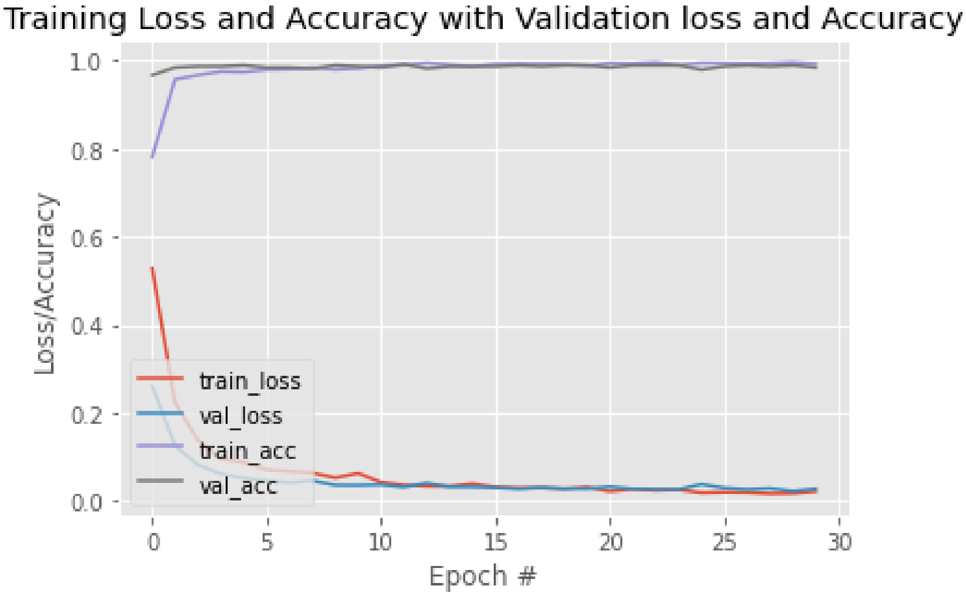
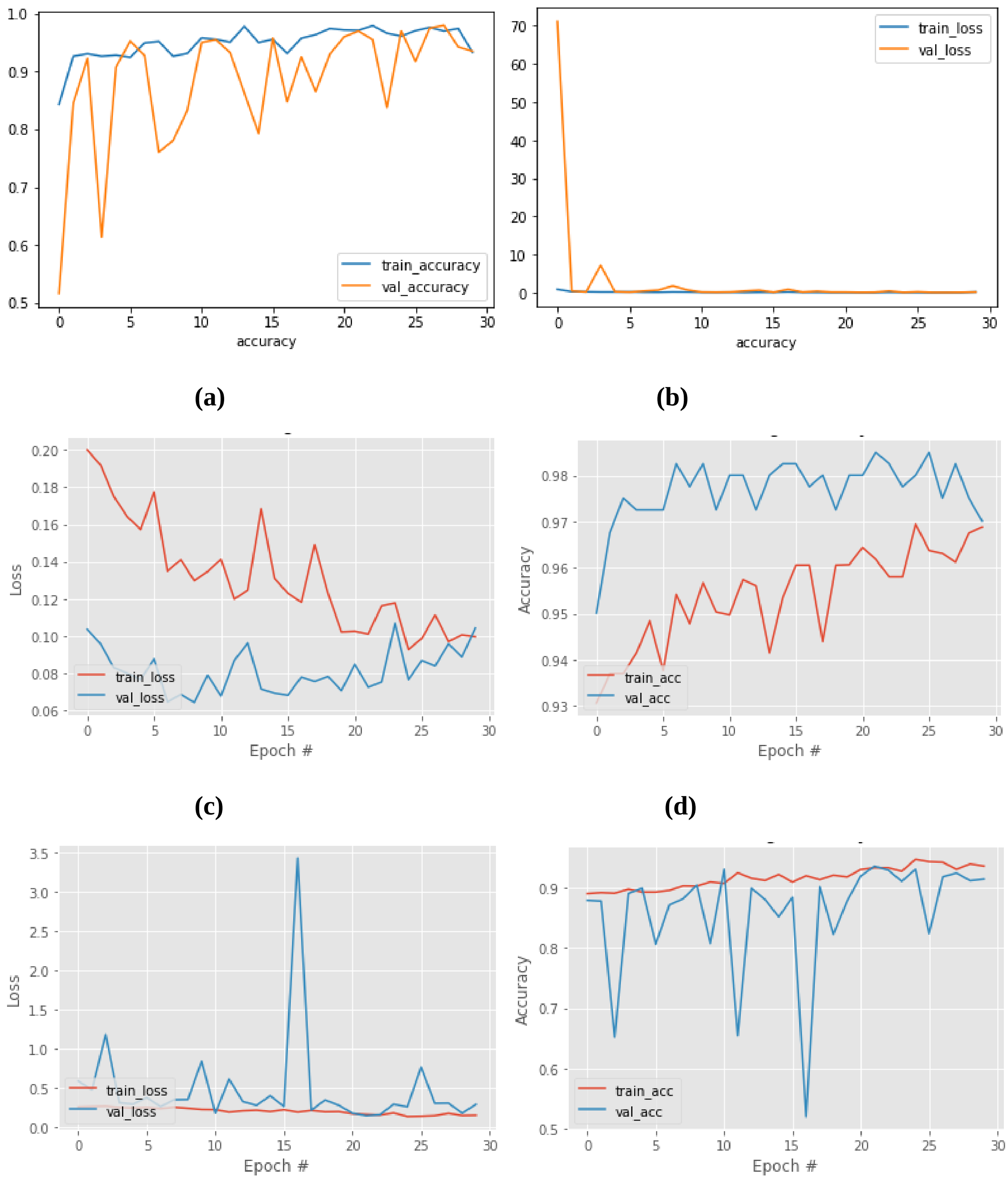


Fig 4.5 Training and Validation Accuracy-Loss graph of MobileNetV2 (binary cross entropy)



(a) **(b)**

(c) **(d)**

(e) **(f)**

Fig 4.6 AlexNet training and validation accuracy-loss graph. (a,b) for without augmentation model fit, (c,d) for with augmentation and changed parameter, (e,f) for with augmentation fit but without changing parameter

Y-axis denotes loss/accuracy. In Fig 4.6 - (a), (d) and (f) the accuracy of train and validation fluctuates but at the end it increased from 0 to nearly 1. However, the loss decreases in (b), (c) and (e). Even though all the graphs are from same algorithm but the fluctuation of the points are different. Because their implementation process is different.

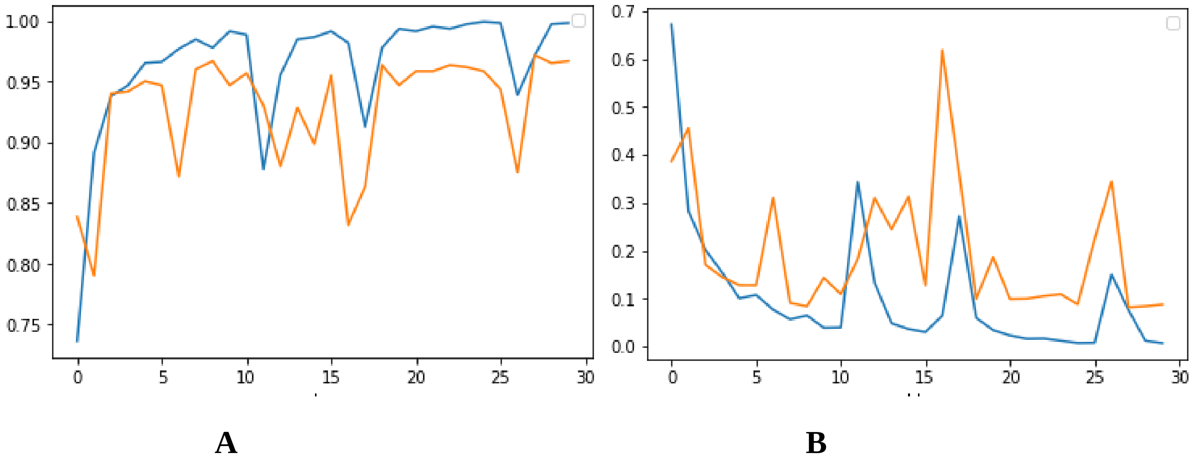


Fig 4.7 A- Training and Validation Accuracy, B- Training and Validation Loss

VGG16

Graph A represents the training and validation accuracy of VGG16 model and graph B represents the training and validation loss. The accuracy graph of both train and validation fluctuates but it tended to provide the accuracy nearly 1. The orange one is for training whereas the blue denotes the validation one.

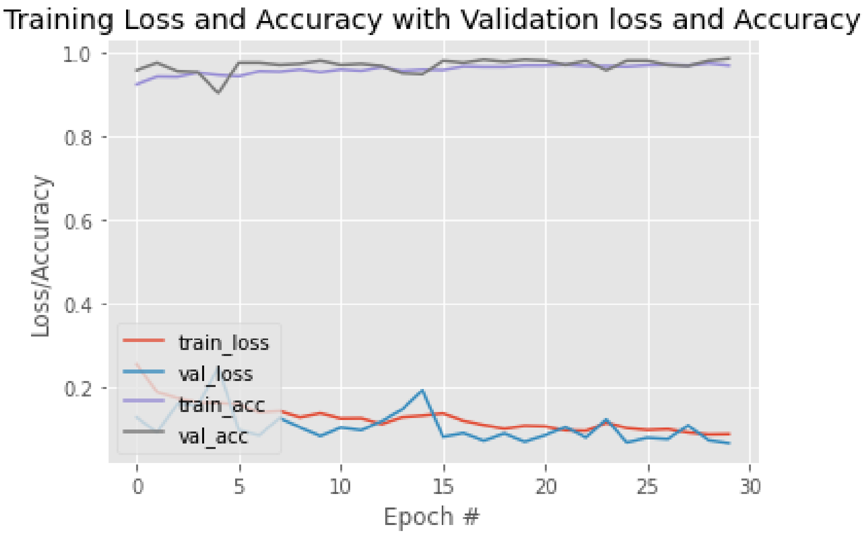


Fig 4.8 Proposed Model Output graphs

Fig 4.8 denotes the training-validation accuracy-loss graph for the proposed model. For this model values don't fluctuate much. Training accuracy is the accuracy which provides how well the model is trained. The validation one is to check if the training is done properly.

4.2 Classification Report

The performance of the models is measured using precision, recall, f1-score, and accuracy after completing the training and testing phase. The formulas that we used are as follows:

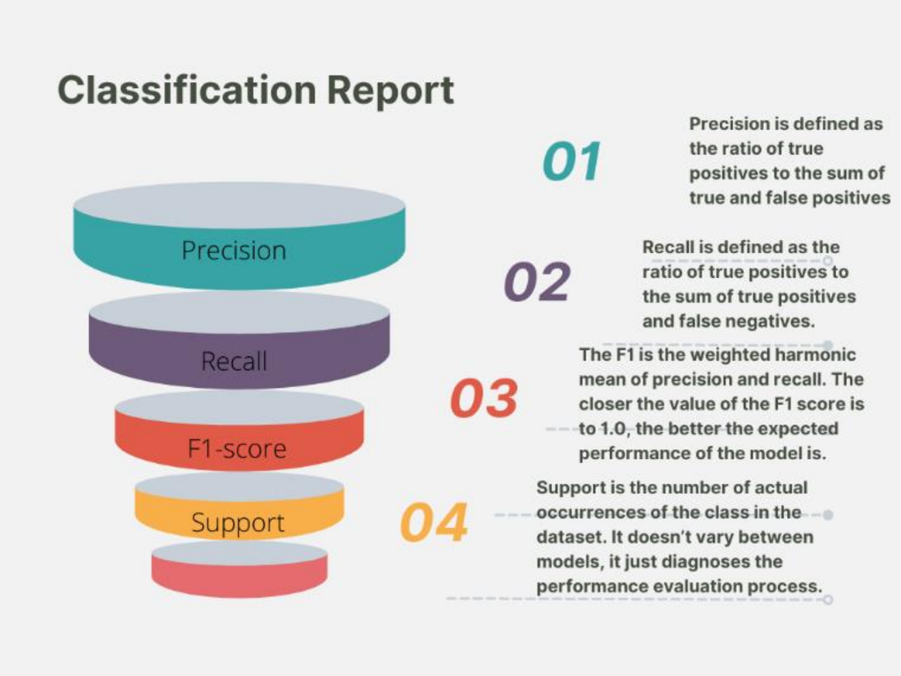
Accuracy = (*TP* + *TN*)/ (*FN* + *TP* + *TN* + *FP*) (1)

Categorical cross entropy in equation 2 is used as a metric for this work. A perfect classifier gets the log loss of 0.

$$logloss = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \qquad (2)$$

Precision = *TP*/(*TP* + *FP*) (3)

Recall = *TP*/ (*FN* + *TP*) (4)



Classification Report

01 Precision is defined as the ratio of true positives to the sum of true and false positives

02 Recall is defined as the ratio of true positives to the sum of true positives and false negatives.

03 The F1 is the weighted harmonic mean of precision and recall. The closer the value of the F1 score is to 1.0, the better the expected performance of the model is.

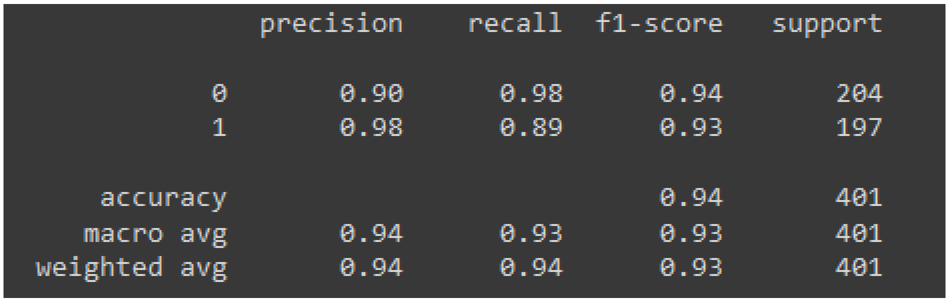
04 Support is the number of actual occurrences of the class in the dataset. It doesn't vary between models, it just diagnoses the performance evaluation process.

Fig Classification report category

f1 Score = 2 x (Precision Recall) / (Precision + Recall) (5)

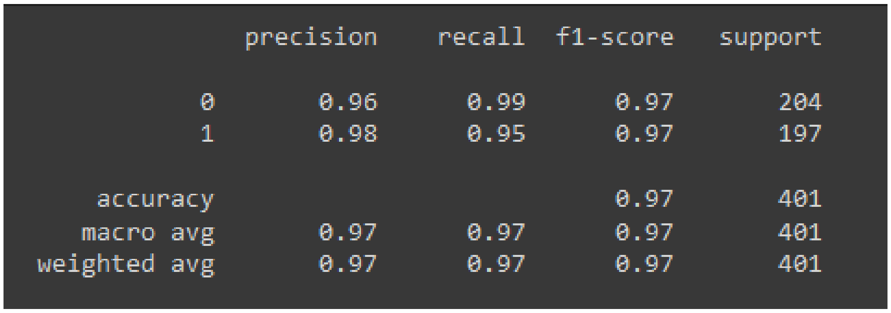
Where TP = True positive, TN = True negative, FP = False positive, FN = false negative,

When we define a positive example as a "person with a mask", recall and sensitivity are same, but precision and specificity are not. Precision is also known as PPV (Positive Predictive Value). All of the metrics (precision, recall, and specificity) provide valuable insight into how effectively our categorization model works. It is important to examine all of them.



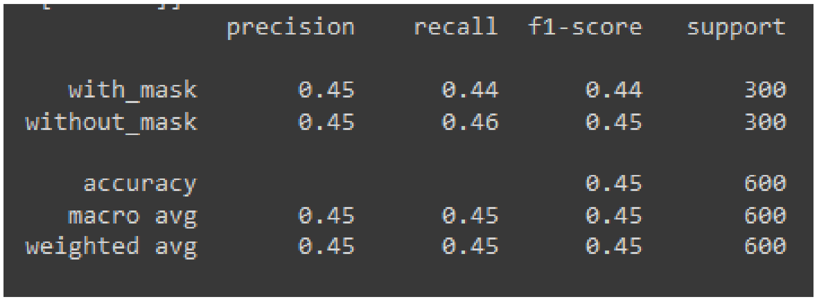
```
              precision    recall  f1-score   support

           0       0.90      0.98      0.94       204
           1       0.98      0.89      0.93       197

    accuracy                           0.94       401
   macro avg       0.94      0.93      0.93       401
weighted avg       0.94      0.94      0.93       401
```

(a)



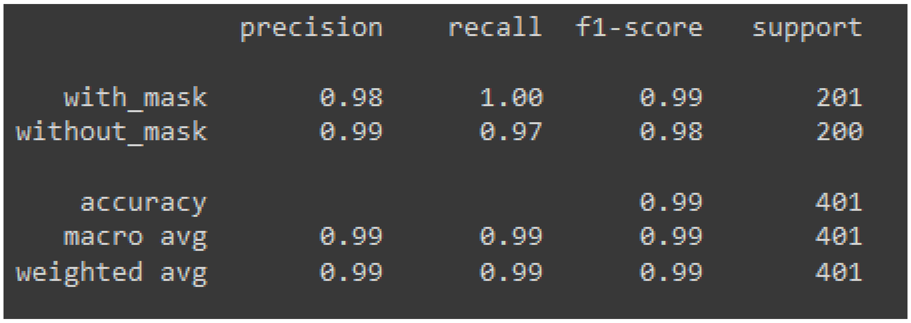
```
              precision    recall  f1-score   support

           0       0.96      0.99      0.97       204
           1       0.98      0.95      0.97       197

    accuracy                           0.97       401
   macro avg       0.97      0.97      0.97       401
weighted avg       0.97      0.97      0.97       401
```

(b)



```
              precision    recall  f1-score   support

   with_mask       0.45      0.44      0.44       300
without_mask       0.45      0.46      0.45       300

    accuracy                           0.45       600
   macro avg       0.45      0.45      0.45       600
weighted avg       0.45      0.45      0.45       600
```

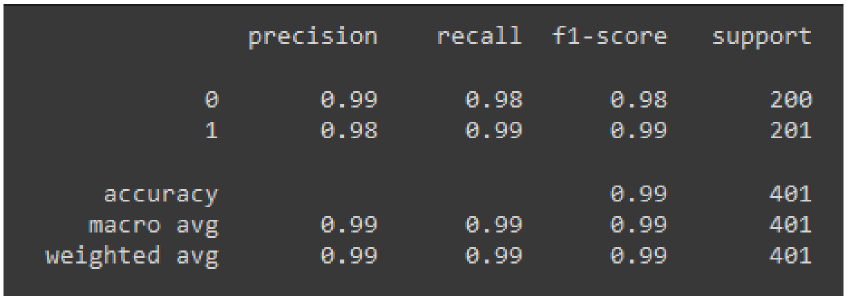
(c)



```
              precision    recall  f1-score   support

   with_mask       0.98      1.00      0.99       201
without_mask       0.99      0.97      0.98       200

    accuracy                           0.99       401
   macro avg       0.99      0.99      0.99       401
weighted avg       0.99      0.99      0.99       401
```

(d)

Fig 4.9 AlexNet Classification graph. a-without augmentation model fit.
b- with augmentation model fit and changed parameters. (c) denotes VGG16 classification report and (d) Represents the Classification Report of MobileNetV2 (binary)



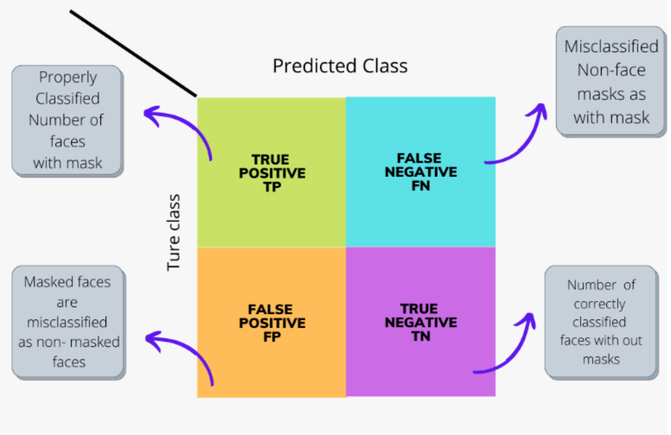
```
              precision    recall  f1-score   support

           0       0.99      0.98      0.98       200
           1       0.98      0.99      0.99       201

    accuracy                           0.99       401
   macro avg       0.99      0.99      0.99       401
weighted avg       0.99      0.99      0.99       401
```

Fig 4.10 Proposed Model Classification Report

In these classification reports, 0 is for without mask and 1 is for with mask. The ratio of true positives to the sum of true and false positives is known as precision. The ratio of true positives to the sum of true positives and false negatives is known as recall. The weighted harmonic mean of accuracy and recall is the F1. The closer the F1 score is to 1.0, the better the model's projected performance will be. The amounts of actual instances of the class in the dataset is known as support. It does not differ between models; rather, it diagnoses the process of performance evaluation. The low false positive rate is related to high accuracy. Accuracy is an excellent statistic, but only when you have symmetric datasets with almost identical false positive and false negative values. The F1 score considers both false positives and false negatives. Although it is not as intuitive as accuracy, F1 is frequently more useful than accuracy, especially if the class distribution is unequal. When false positives and false negatives have equivalent costs, accuracy works well. It's best to look at both Precision and Recall if the cost of false positives and false negatives is considerably different.

4.3 Confusion Matrix

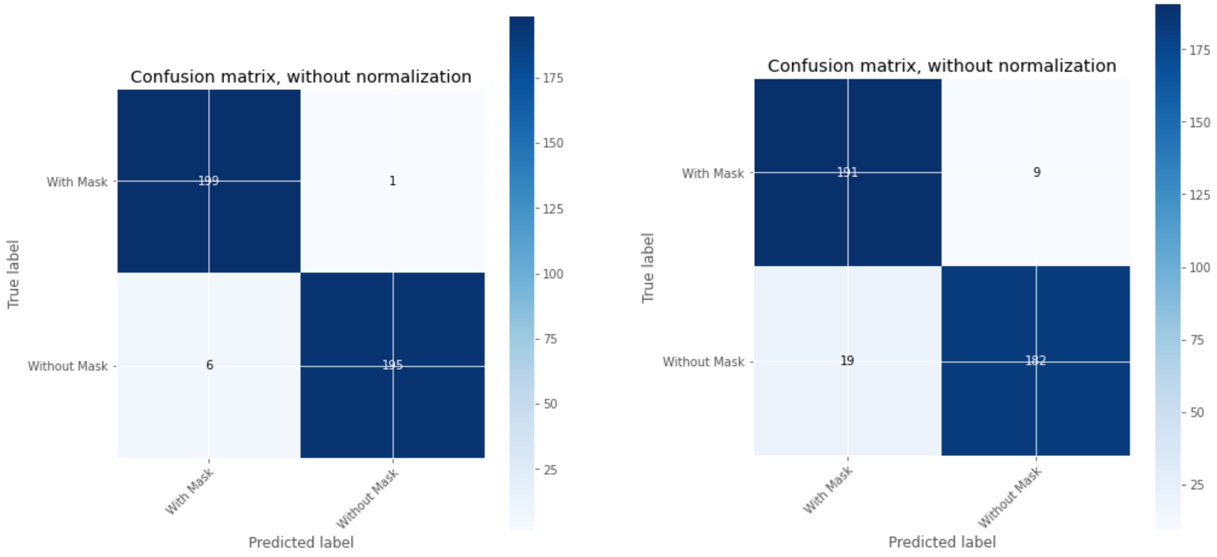
After the classification, the confusion matrix is used to evaluate the performance of the approaches.



Predicted Class
Ture class
TRUE POSITIVE TP
FALSE NEGATIVE FN
FALSE POSITIVE FP
TRUE NEGATIVE TN
Properly Classified Number of faces with mask
Misclassified Non-face masks as with mask
Masked faces are misclassified as non- masked faces
Number of correctly classified faces with out masks

Fig 4.11 Confusion Matrix

True positive values refer to images which were labelled true and after prediction by model gave true result. Likewise, for True negative refers to images which were labelled true but after prediction resulted in false result. False positive refers to images which were labelled false and after prediction resulted in false hence false positives. False negative refers to images which were labelled false and after prediction resulted in true hence false negatives. These evaluation metrics were chosen because of their ability to give best results in balanced dataset.

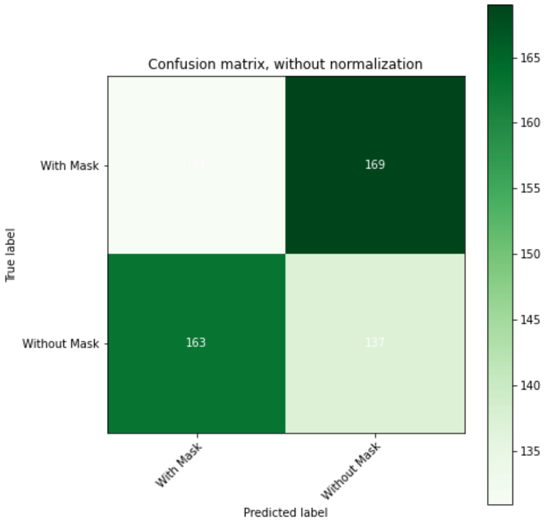


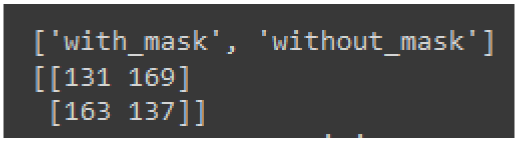
(binary categorical)

(Sparse Categorical)

Fig 4.12 Confusion Matrix of MobileNetV2

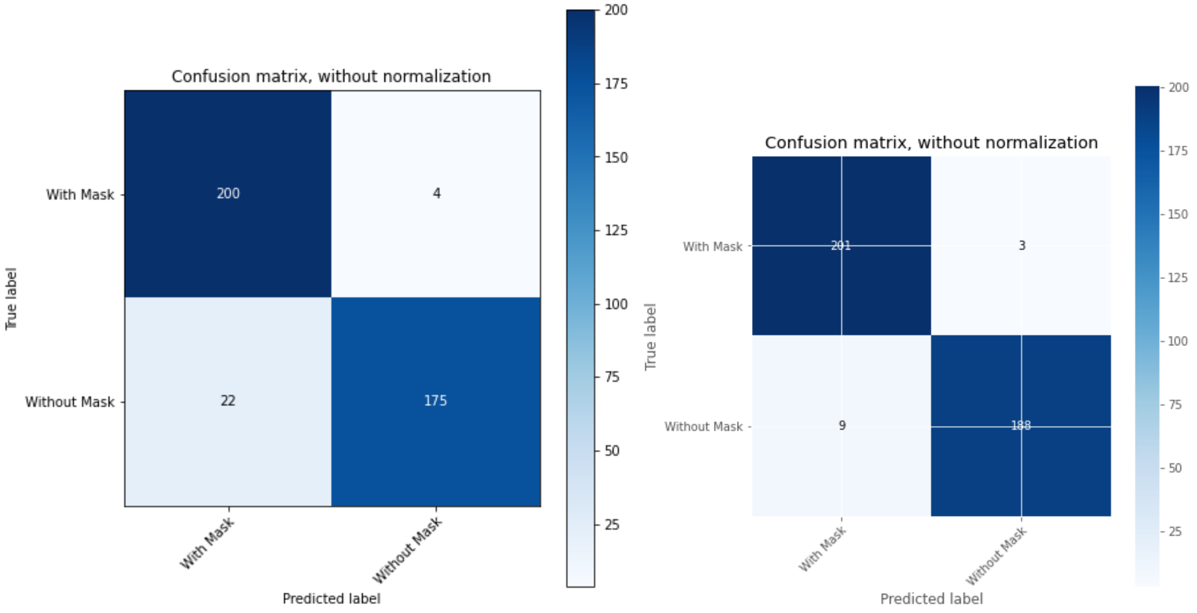
The confusion matrix shown in fig 4.12 depicts a form to compare the labels, model prediction, and actual labels it was supposed to predict. It is showing where the model is getting confused. The confusion matrix is plotted with the help of heatmap showing a 2D matrix data in graphical format. It has successfully identified 199 true positives, 1 false negative in binary categorical. 6 false-positive, and 195 true negative. In Sparse Categorical identified 191 true positives, 9 false negatives in binary categorical. 19 false-positive, and 182 true negative. In figure 4.13 Confusion matrix of VGG16 identified 129 true positives, 169 false negatives, 163 false-positive, and 137 true negative. Confusion matrix of AlexNet A- Without augmented model fit identified 200 true positives, 4 false negatives, 22 false-positive, and 175 true negative.





```
['with_mask', 'without_mask']
[[131 169]
 [163 137]]
```

Fig 4.13 Confusion matrix of VGG16



A **B**

Fig 4.14 Confusion matrix of AlexNet

A- Without augmented model fit

B- With augmented model fit and changed parameter

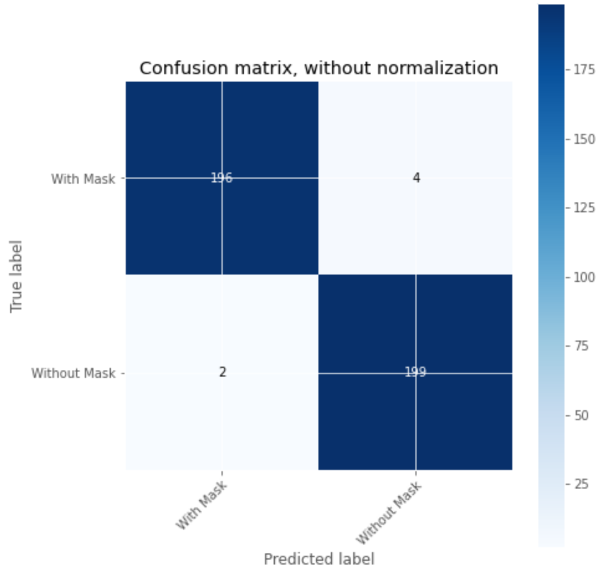


Fig 4.15 Confusion Matrix of Proposed Model

In fig 4.11 it is marked how the confusion matrix defines classified and misclassified data

4.4 Real Time Detection

To detect people with mask in real time, we have implemented a code to access the camera of computer/laptop which will take live photos and will detect the mask. After taking a picture as input it will first work on its' region of interest such as eyes, mouth and nose. Then it will detect if there is any mask with the help of trained model. With mask will show a green rectangular box, whereas there will be red for without mask.

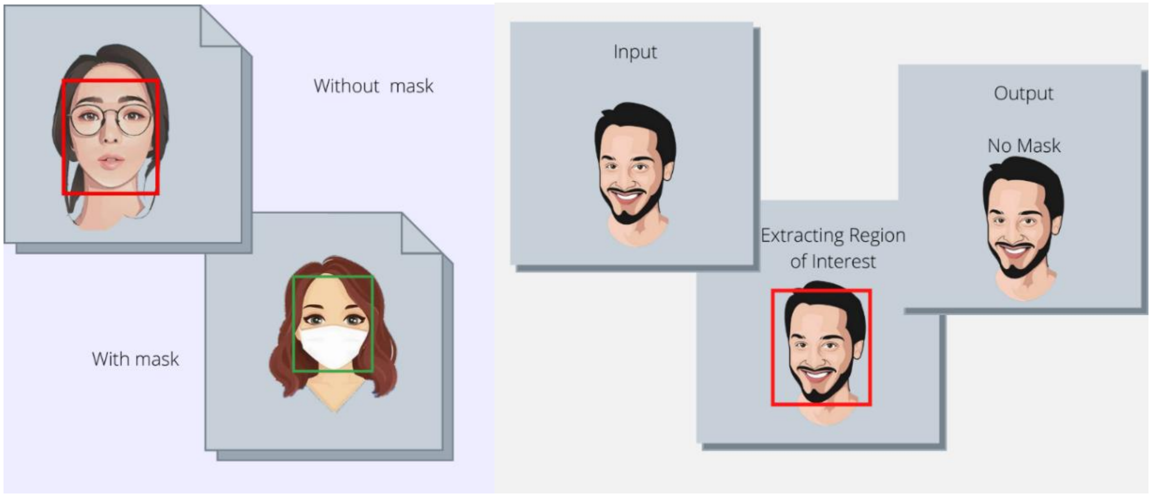
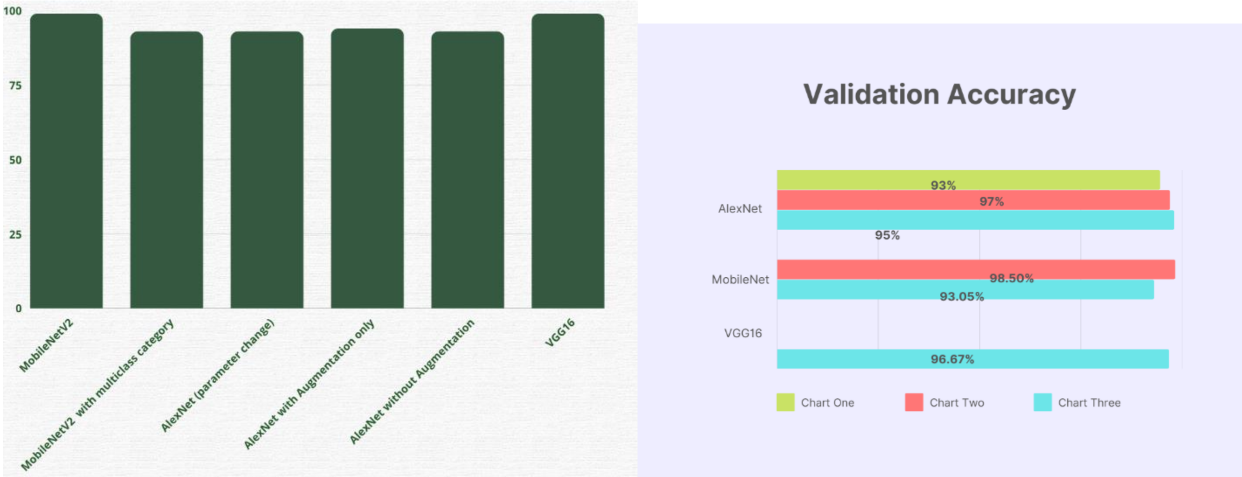


Fig 4.16 Face mask detection in Real Time

4.5 Comparison



Training Accuracy

Validation Accuracy

Fig 4.5.1 Comparison of training and Validation Accuracy among VGG, AlexNet and MobileNetV2

Using the transfer learning method, the face mask detection algorithm is found. In Fig 4.5.1 it is clearly shown that MobileNetV2 with binary cross entropy has the highest training accuracy. On the other hand, for the validation accuracy VGG16 and MobileNetV2 (binary cross entropy) is almost 99%. The more dataset the model will get, it will give more accuracy. But the dataset needs to be good, have a good resolution. Moreover, the proposed architecture has the accuracy of 98% with the same dataset. MobileNetV2 is the fastest algorithm among these models. It trains these models quickly with a better accuracy than others.

CHAPTER FIVE

Conclusion and Future work

5.1 Conclusion

The majority of countries have made wearing a face mask mandatory due to an epidemic of COVID-19. In busy places, manual inspection of the face mask is necessary. As a result, researchers have been inspired to automate the face mask detecting method. A color image is sent into the MobileNet, which outputs a multi-dimensional feature map. The feature map is transformed into a feature vector of 64 features by the global pooling block used in the proposed model. Finally, the softmax layer uses the 64 characteristics to achieve binary classification. On two publicly available datasets, we tested our proposed model. On DS1 and DS2, our suggested model attained 99 percent and 100 percent accuracy, respectively.

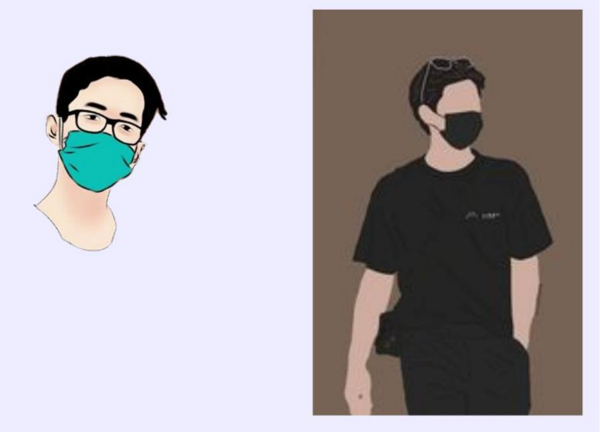


Fig 7.1 Face Mask Detected

In this paper, we used CNN architectures MobilenetV2 architecture, VGG16 and AlexNet . Our main goal was to propose a suitable, high-accuracy model that would make mask identification straightforward throughout the epidemic. We can try to add more models to compare with Mobilenetv2 and link this model with IoT to detect humans without masks automatically in order to analyze performance with a larger dataset.

The goal of this thesis was to construct a face mask detector, which was accomplished by implementing the concept using cutting-edge technologies such as opencv, mobilenet, machine learning, and deep learning. Masks are getting increasingly popular in recent years. Masks are one of the few means to guard against the corona virus in the absence of immunization, and they play a vital role in preserving people's health from respiratory infections. This project can be integrated with embedded technology and deployed in a variety of public venues, including airports, train stations, offices, schools, and public areas, to protect human safety.

5.2 Future Work

In the future, physical distance integration could be introduced as a feature, or coughing and sneezing detection could be added. It will compute the distances between each participant and look for any coughing or sneezing. If the mask is not properly worn, a third class that marks the image as 'improper mask' might be added. Researchers could also recommend a better optimizer, enhanced parameter setting, and the usage of adaptive models. The short dataset size, the experimental findings demonstrate that using transfer learning to identify faces with and without masks is a good strategy. We can recommend this approach to be used in practice to detect faces with masks and faces without masks, which can potentially contribute to public healthcare, because the main goal of all research is to obtain good results and achieve an effective recognition system, because the combination of MobileNet-V2 model with default parameters got an excellent performance (98.50%).

There are many more different cases in which this model can be integrated for the safety of the public:

- Identify a person if he is doing any crime by wearing face mask.
- Identify what type of mask is the person wearing.
- Coughing and Sneezing Detection.
- Temperature Screening

This paper emphasized flaws such as retaining image resolution during the detection process, a lack of a large dataset, categorical classifications, and others. It also defined potential scopes, such as a variety of datasets and facemask types, diverse facemask wearing conditions, masked face reconstruction, and so on. This in-depth analysis will help the research community better grasp current facemask identification techniques. Researchers will propose unique techniques to cover those gaps by examining the weaknesses and future challenges in this field.

References

1. D. Hunt, “Pathogenesis of tissue injury in the brain in patients with systemic lupus erythematosus,” in *Systemic Lupus Erythematosus*. El- sevier, pp. 341–348.
2. WHO, “World health organization,” 2020, accessed 17 Oct 2020.

   [Online]. Available: https://www.who.int/health-topics/coronavirus
3. Woods, A., BDaily News, Jun 2020, Britain faces an anxiety crisis as people return to work,https://bdaily.co.uk/articles/2020/06/22/britainfaces-an-anxiety-crisis-as-people-return-to-work.

4. Howard, J., Huang, A., Li, Z., Tufekci, Z., Zdimal, V., van der Westhuizen, H., von Delft, A., Price, A., Fridman, L., Tang, L., Tang, V., Watson, G.L., Bax, C.E., Shaikh, R., Questier, F., Hernandez, D., Chu, L.F., Ramirez, C.M., Rimoin, A.W., Face Masks Against COVID-19: An Evidence Review, Preprints, 2020, 2020040203, (doi:10.20944/preprints202004.0203.v1).
5. Verma, S., Dhanak, M., & Frankenfield, J. (2020), Visualizing the effectiveness of face masks in obstructing respiratory jets, Physics of fluids (Woodbury, N.Y. : 1994), 32(6), 061708. https://doi.org/10.1063/5.0016018
6. N. H. Leung, D. K. Chu, E. Y. Shiu, K.-H. Chan, J. J. McDevitt, B. J. Hau, H.-L. Yen, Y. Li, D. K. Ip, J. M. Peiris et al., “Respiratory virus shedding in exhaled breath and efficacy of face masks,” Nature medicine, vol. 26, no. 5, pp. 676–680, 2020.
7. Nowrin, Afsana, et al. "Comprehensive review on facemask detection techniques in the context of covid-19." *IEEE access* (2021).
8. Rao TS, Devi SA, Dileep P, Ram MS. A Novel Approach To Detect Face Mask To Control Covid Using Deep Learning. European Journal of Molecular & Clinical Medicine. 2020;7(6):658-68.
9. H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, in Proceedings of the IEEE conference on computer vision and pattern recognition (2015), pp. 5325{5334
10. P. Khandelwal, A. Khandelwal, S. Agarwal, Using computer vision to enhance safety of workforce in manufacturing in a post covid world, arXiv preprint arXiv:2005.05287 (2020)
11. M. Jiang, X. Fan, Retinamask: A face mask detector, arXiv preprint arXiv:2005.03950 (2020)
12. P. Viola and M. J. Jones, “Robust real-time face detection,” *Int. J.Comput. Vision*, vol. 57, no. 2, pp. 137–154, May 2004.
13. Satapathy, Sandeep Kumar, et al. "Deep learning based image recognition for vehicle number information." *International Journal of Innovative Technology and Exploring Engineering* 8.8 (2019): 52-55.
14. Pathak, Mrunal, Vinayak Bairagi, and N. Srinivasu. "Multimodal Eye Biometric System Based on Contour Based E-CNN and Multi Algorithmic Feature Extraction Using SVBF Matching."*International Journal of Innovative Technology and Exploring Engineering*
15. 20Ravi, Sunitha, et al. "Multi modal spatio temporal co-trained CNNs with single modal testing on RGB–D based sign language gesture recognition." *Journal of Computer Languages* 52 (2019): 88-102.
16. Patel, Ashok Kumar, Snehamoy Chatterjee, and Amit Kumar Gorai. "Development of a machine vision system using the support vector machine regression (SVR) algorithm for the online prediction of iron ore grades." Earth Science Informatics 12.2 (2019): 197-210.
17. D. Matthias, M. Chidozie, Face mask detection application and dataset

18. N.-C. Ristea and R. T. Ionescu, ``Are you wearing a mask? Improving mask detection from speech using augmentation by cycle-consistent GANs,'' 2020, *arXiv:2006.10147*. [Online]. Available: http://arxiv.org/abs/2006.10147
19. Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Pei, et al., Masked face recognition dataset and application, arXiv preprint arXiv:2003.09093 (2020)
20. M. Inamdar and N. Mehendale, ``Real-time face mask identification using facemasknet deep learning network,'' India, Jul. 2020. [Online]. Available: https://ssrn.com/abstract=3663305
21. V. V. V. Vinitha, ``COVID-19 facemask detection with deep learning and computer vision,'' *Int. Res. J. Eng. Technol.*, vol. 7, no. 8, pp. 3127_3132, 2020.
22. IBM Cloud Education. 2020. *What are Convolutional Neural Networks?* Available at: https://www.ibm.com/cloud/learn/convolutional-neural-networks. Accessed 10 February 2022.
23. Kumar, A. 2022. *Different Types of CNN Architectures Explained*. Available at: https://vitalflux. com/different-types-of-cnn-architectures-explained-examples/. Accessed 15 February 2022.
24. "CNN | Introduction to Pooling Layer," GeeksforGeeks, 2019. [Online]. Available: https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/. Accessed: 2021-04-27
25. "Convolutional neural networks," [Online]. Available:https://ml4a.github.io/ml4a/convnets/. Accessed: 2021-05-10
26. K. I. Lin Dong, "Diagnosis of Breast Cancer from Mammogram Images Based on CNN," vol. 8, 2020.
27. "Convolutional neural networks," [Online]. Available: https://ml4a.github.io/ml4a/convnets/. Accessed: 2021-05-10
28. Face mask detection using MobileNet and Global Pooling Block
29. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. "Mobilenetv2: The next generation of on-device computer vision networks. URL Https://Ai. Googleblog. Com/2018/04/Mobilenetv2- next-Generation-of-on. 2020.
30. Sharma, S. 2021. *TensorFlow for Beginners with Examples and Python Implementation.* Available at: https://www.analyticsvidhya.com/blog/2021/11/tensorflow-for-beginners-with-examples-and-pythonimplementation/. Accessed 25 February 2022.
31. Simplilearn. 2021. *The Best Introductory Guide To Keras*. Available at: https://www.simplilearn. com/tutorials/deep-learning-tutorial/what-is-keras. Accessed 2 March 2022.
32. "https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac".Kumar, A. 2022. *Different Types of CNN Architectures Explained*. Available at: https://vitalflux. com/different-types-of-cnn-architectures-explained-examples/. Accessed 15 February 2022.

33. Kumar, A. 2022. *Different Types of CNN Architectures Explained*. Available at: https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/. Accessed 15 February 2022.

34. An automated System to limit covid 19 using facial mask detection in smart city network(2020, IEEE) https://ieeexplore.ieee.org/document/9216386

35. Reza, Ahmed Wasif, et al. "ModCOVNN: a convolutional neural network approach in COVID-19 prognosis." *International Journal of Advances in Intelligent Informatics* 7.2 (2021): 125-136.

36. "https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac".

37. https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0

38. https://www.hindawi.com/journals/scn/2021/9956773/

39. https://ieeexplore.ieee.org/abstract/document/8013796 - Modulation Format Recognition and OSNR Estimation Using CNN-Based Deep Learning.

40. https://arxiv.org/abs/1412.6980 “Gradient Descent Optimization in Deep Learning Model Training Based on Multistage and Method Combination Strategy.

41. H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2684-2691, doi: 10.1109/IJCNN.2017.7966185.