FACE RECOGNITION WITH AND WITHOUT MASK USING MACHINE LEARNING APPROACH

BY

Md. Ashikur Rahman
ID: 191-15-2469

Mizanur Rahman Masum
ID: 191-15-2502

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. S.M Aminul Haque
Associate Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
25 JANUARY, 2023

APPROVAL

The Department of Computer Science and Engineering at Daffodil International University has accepted the Project by Md. Ashikur Rahman and Mizanur Rahman Masum titled "Face Recognition with mask and without mask using machine learning approach." for partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering. The presentation has been held on 25 January, 2023.

BOARD OF EXAMINERS

______________________ **Chairman**

Dr. Touhid Bhuiyan
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

______________________ **Internal Examiner**

Dr. Md. Atiqur Rahman
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Shayla Sharmin 25.1.23 **Internal Examiner**

Shayla Sharmin
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

25-01-23 **External Examiner**

Dr. Dewan Md Farid
Professor
Department of Computer Science and Engineering
United International University

DECLARATION

Daffodil International University Department of Computer Science and Engineering Associate Professor, Dr. S.M. Aminul Haque, has supervised this work. Moreover, we affirm that we have not previously submitted this work, in whole or in part, for credit toward another degree.

Supervised by:

Dr. S.M Aminul Haque
Associate Professor
Department of CSE
Daffodil International University

Submitted by:

Md. Ashikur Rahman

Md. Ashikur Rahman
ID: 191-15-2469
Department of CSE
Daffodil International University

Mizanur Rahman Masum

Mizanur Rahman Masum
ID: 191-15-2502
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

We would want to begin by giving God the glory for making it possible for us to finish our senior year project.

Dr. S.M. Aminul Haque, Associate Professor, Department of CSE, Daffodil International University, Dhaka, has been an invaluable resource for us, and we are very appreciative and obliged to him. Deep Knowledge & keen interest of our supervisor in the field of Machine Learning to carry out this project. This endeavor would not have been feasible without his unending patience, intellectual direction, continuous encouragement, frequent and vigorous supervision, constructive criticism, helpful counsel, reading of many poorer drafts, and corrections at every level.

Our deepest appreciation goes out to Dr. Touhid Bhuiyan, Professor and Head, Department of CSE, at Daffodil International University, for all of his support and guidance during the course of this project. We appreciate everyone who contributed to this discussion throughout our time at Daffodil International University.

At the end of the day, we have to give thanks to our parents for all the love and patience they've shown us.

ABSTRACT

The global spread of COVID-19 has had an immediate effect on our everyday lives by disrupting international commerce and transportation. Mask use is highly recommended to prevent the spread of illness. Consequently, covering one's face with a mask has become mandatory. When everyone is hiding their identities behind masks, facial recognition software can't pick out a single face. Many businesses and organizations providing public services mandate that customers and the employees should wear protective masks. Therefore, identifying masks and who is the person behind the mask are essential in serving the global community. To solve this problem, we provide a face recognition method that is able to tell the difference between masked and unmasked faces and can also recognize the person under the mask. In this research, we propose combining MobileNetV2, VGG19 with the HOG method for fast and accurate recognition. The proposed method is tested extensively and shows promising results (94% accuracy in training and testing).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction

People are aware that Covid-19 is stepping into a new world since it has produced a completely new frequency. We must act swiftly to meet the new requirements that are all around us, despite the fact that our society is changing quickly right now. Everyone's first priority will be to create a risk-free atmosphere in order to increase the significance of life. We need to take precautions to ensure the safety of individuals who are returning to the workforce, as well as ourselves and our loved ones. New methods are created every day to ensure legality. It is critical to be alert at all times in order to maintain a safe atmosphere favorable to public safety. Masks have established a new standard in everyday life in this scenario.

1.2 Motivation

A worldwide epidemic COVID-19 conditions arose in a worldwide outbreak of hazardoussickness. Millions of people are ill in a single day. This virus can spread from person to person viadroplets and the air. The people's carelessness and lack of consciousness were the main contributors to the virus infection. Everyone should spread awareness during this disaster period and naturally should engage in some self-care exercises. Therefore, in order to protect one another,everyone has to put on a face mask and make sure it fits properly whenever they go outdoors. Also,there are security issues. Constant surveillance is exceedingly tough and time-consuming. This study primarily contributes to finding a solution and assisting people in self-defense as well as maintaining security.

1.3 Objective

We need to do something about the COVID-19 epidemic. We built a face mask detector using a convolutional neural network. In order to train, validate, and test the model, we used unmasked photos of faces. The data set contains 4000 such photos. These images are derived from a variety of datasets, including RMFD and Kaggle. The model was created using images and real-time video feeds. We looked at measures like accuracy, precision, and recovery to determine the optimal base model, and the MobileNetV2 architecture had the greatest performance with 96% accuracy and 99% recall. Additionally, MobileNetV2 is computationally efficient, which facilitates the installation of the model in embedded devices. This face mask detector can be utilized in a variety of settings, such as shopping malls, airports, and other crowded locations, to monitor the general public and prevent the spread of illness by recognizing who is and is not adhering to the basic criteria. Examples of these settings include: malls, airports, and other crowded places.

1.4 Expected Outcome

As we said before, our research will make easier ways for people to detect masks on their face as well as who is the person behind the mask. We are trying our best to improve the accuracy of our project. The transfer learning CNN model used to identify faces hidden behind the mask will be detected by our neural network. People will generally walk through in front of the camera and the system will automatically recognize the person.

1.5 Report Layout

In chapter 1, we attempted to address the fundamental ideas of "Face Recognition with and without Mask Using Machine Learning Approach," as well as discuss the rationale, aim, and expected outcome of our study.

In chapter 2, concentrate on similar works, a quick explanation of the topic, and the obstacles.

We discuss research methods in Chapter 3.

The specifics of the experimental results are described in Chapter 4.

In the last chapter, we concluded about my assessment results as well as some additional elements that may be incorporated in future works to improve the quality of my research job.
In the final section of the report, we add

- References
- Appendix
- Plagiarism report
- List of figures

CHAPTER 2
BACKGROUND STUDY

2.1 Introduction

This chapter includes the discussion about the other researches about Face Recognition with and without mask. In this research we found out several Transfer Learning based Models are being used. Those are:

1. VGG16
2. MobileNetV2
3. ResNet50

We used python as a programming language and the Transfer Learning based CNN model.

2.2 Literature Review

This study [1] covers machine learning for masked face identification in the context of the COVID-19 epidemic. The globe is now being plagued by a contagious illness caused by the corona virus (COVID-19). Facial recognition systems struggle to pick out an individual's disguised face in a crowd. To deal with this problem, they have introduced a masked face recognition system that employs the support vector machine and radio frequency based recognition technologies. One primary objective of the proposed masked face recognition system is to accurately identify the concealed face. The dataset was built using 28 classes and included 1470 pictures for training and testing. On average, SVM achieves a recognition accuracy of 97% for a task involving 28 classes, whereas random forest achieves an accuracy of 98.2%.

In this research, we apply deep learning to identify facial masks in real time [2]. As a consequence of the COVID-19 pandemic, everyone is aware of the need of regularly wearing a face mask. It is possible to disseminate the SARS COV-2 (Severe Acute Respiratory Syndrome) virus by coughing and sneezing. The virus is contagious and easily spreads via direct contact with an infected individual or by contact with contaminated surfaces. In an effort to reduce the spread of illness, the World Health Organization and local authorities have advocated for the widespread use of protective face coverings. In real time, this gadget can tell whether someone is donning a disguise. In this research, we look at how MobileNetV2 performs in comparison to VGG16. Whether or not a person is donning a mask may be determined by this system thanks to the use of real-time technology. During training, it obtained 98% accuracy using a dataset of around 4000 images with a width and height of 224x224. To determine which CNN is most suited for this task, we trained and combined this model with two others.

A Deep Learning-based Approach for Real-time Facemask Detection is the topic of the following study [3]. A worldwide public health emergency is being caused by the COVID-19 pandemic. Public areas must be maintained free of the terrible outcomes of this pandemic. Many nations have embraced the practice of mandating the use of face masks during public health emergencies. It's becoming harder to visually confirm, in real time, whether a large group of people are indeed all wearing face masks. This research makes use of deep learning to accomplish its aims (DL). The suggested procedure consists of two phases. The first step in creating a DL model that can recognize and locate facemasks and judge whether they are being worn appropriately occurs off-line. edge computing is used to find masks in real time using a deployed DL model. In this research, we propose leveraging MobileNetV2 for automatic face mask recognition. Several tests have been conducted, and the results are promising for the suggested method (99.1% accuracy in both training and testing). The MobileNetV2 also outperforms state-of-the-art models in terms of training time and accuracy, as shown by a number of side-by-side comparisons with networks likeResNet50, DenseNet, and VGG16.

Recent efforts at MFR that make use of deep learning methods are summarized in this paper [4]. This research analyzed the standard MFR pipeline used in recent times and identified the most recent developments that have improved the efficiency of the MFR method. Important topics presented that have direct impact on MFR systems include picture preprocessing, feature extraction, face recognition and localization, face unmasking and restoration, identity matching, and verification. The most essential inference to be drawn is that the MFR job will be thoroughly studied, and that frequent proposals for new studies and operational actions will be made available in the literature. However, when using conventional FR methods for MFR, there is sometimes a major performance impact. In addition, the performance of hybrid deep neural networks in performing many tasks simultaneously is crucial to the accuracy of the MFR. This includes tasks like mask detection and face reconstruction. Learning from metrics will make ID checks more accurate.

Methods of Finding Faces Through Masks in a Large Group of People The concept of "Federated Learning" [5] serves as the central focus of this investigation. In order to accomplish multiscale face localization, the authors of this research construct the DRFL network and the SRNet20 network, respectively, to recognize faces hidden by masks. They define federated learning as a privacy-preserving method of cooperative training that makes use of several data sources from different parties. To start, Wider Face is a public dataset used by the generic face localization network. The method provides a solid foundation from which to build when attempting to identify people. There are a total of 158,989 faces annotated in the training set and 39,496 faces annotated in the validation set, with a total of 32,203 pictures and 393,703 faces in total. Second, the masked face classification network is trained with a unique dataset. The training set includes 18,000 images, including 9,000 with faces obscured by masks and 9,000 without. In all, the test set consists of 1,751 images; 656 of them include people wearing masks, while the remaining 1,095 do not.

Using publicly accessible masked face datasets and state-of-the-art face recognition techniques, this study [6] focuses on the Masked Face Recognition model. We employed both synthetic masked face recognition data sets and real-world labelled face datasets. They retrained the FaceNet model using transfer learning using the Inception ResNet and ResNet50 architectures, and the training set accuracy was 99.98%. They employed hyperparameter adjusting to avoid overfitting on validation sets. While expanding the model to the validation set, they encountered a few issues, all of which were eventually resolved.

This article is called Convolutional Neural Network Face Mask Detection [7]. (CNN).

COVID-19 has disrupted worldwide commerce and travel. Protective face masks are becoming standard. Face recognition and identification will fascinate biometric models. Global civilisation needs face mask identification. They use TensorFlow, Keras, OpenCV, and Scikit-Learn. This project uses rapid picture pre-processing with the mask over the central faces. Our model was trained on photos of individuals with and without face masks. Picture preparation, image information extraction, and image classification comprised our work. Features extraction and convolutional neural networks identify masked people. 99.1% accuracy. Face masks should be used throughout this epidemic.

Machine learning (ML) methods might be used to implement the aforementioned health criteria. In order to effectively carry out the two (2) processes stated, Face Masks Detection and Face Recognition, this research [8] use Convolution Neural Networks (CNN) and Histogram ofOriented Gradients (HOG) feature descriptors using a linear SVM machine learning algorithm. Inthis study, we'll show you how to use a Convolution Neural Network (CNN) to identify whether or not someone is covering their face, and we'll also show you how to add an extra parameter to help you tell whether they're covering their face enough. The Histogram of Oriented Gradients (HOG) feature descriptor is used in conjunction with a linear Support Vector Machine (SVM) machine learning technique to accomplish the task of face detection.

In this study, we first introduce the MAFA dataset [9], which consists of 35,806 masked faces and 30,811 Internet photos. Each face in the dataset is partially obscured by a mask, and the masks' orientations and opacity levels vary. Using this dataset as a basis, we also offer LLE-CNNs for masked face identification, each of which is comprised of three parts. In order to identify likely face areas in the input picture and describe them using high-dimensional descriptors, the Proposal module combines two pre-trained CNNs. By using dictionaries learned from a large dataset of synthetic examples of normal faces, masked faces, and non-faces, the Embedding module transforms these descriptors into a similarity-based descriptor. It seems that the issue of facial recognition has not yet been fully resolved, despite the rapid development of machine learning algorithms in this area. The facial recognition technology proposed in While the speed of certain face detectors may reach up to 35 FPS or even 400 FPS, on the public image benchmark AFW, it achieves an average accuracy of 98.0% by using cascaded Convolutional Neural Networks.

In this article, **[10]** discusses how COVID-19 has had a major influence on the world. According to studies, wearing a face mask is one of the steps to limit the risk of viral transmission. In addition, a lot of public spaces and service providers only permit patrons to utilize their facilities or visit their locations provided they correctly don masks. As a result, it is hard to manually track the consumer to determine whether or not they have the mask. As many detectors are utilized in conjunction with CNN architecture, different parts of an object can be encapsulated. These structures have also been used to obtain cutting-edge results in the field of fine-grained identification, such as identifying a dog breed, a bird species, or a vehicle type. The trained model's accuracy grows in proportion to the quantity of the data set, which is crucial for improving object detection. Over 90% accuracy will be attained by the model, which is continuously being improved. In this pandemic crisis, where everyone wants to resume normal routine, this technique will be useful for monitoring the use of face masks at work.

This paper [11] is about Face Mask Detection using Machine Learning. There must be action taken to slow the COVID-19 pandemic's spread. A convolutional neural network was deployed to model a face mask detector. We used the data set, which comprises 1916 photographs of masked faces exposed in 1919, model training, validation, and testing These images are derived from a variety of datasets, including RMFD and Kaggle. The model was created using images and real-time video feeds. We looked at measures like accuracy, precision, and recovery to determine the optimal base model, and the MobileNetV2 architecture had the greatest performance with 96% accuracy and 99% recall. Additionally, MobileNetV2 is computationally efficient, which facilitates the installation of the model in embedded devices. This face mask detector may be utilized in a number of settings, including malls, airports, and other congested environments, to monitor the general public and prevent disease transmission by recognizing who is and is not following basic norms.

This article [12] is about Detecting Masks in Human Faces using Machine Learning Approach. It is necessary to wear a mask because to Covid19. Trying to figure out if someone is wearing a mask or not is difficult. The lack of a properly annotated dataset to train our algorithms makes it a difficult task as well. In our strategy, they favor deep learning built on CNN. This algorithm has undergone extensive research and covers a wide range of face detection. The proposed gadget makes use of CNN for face detection and K-means for later-stage mask detection. The technology can identify people using masks as well as those who are not wearing masks or who have worn them improperly. To improve the accuracy of our model because we don't have a suitable dataset, they manually produced the "facial mask" dataset. After testing our suggested methodology on our dataset, we have obtained a satisfying result with sufficient precision. Detecting face masks is one of the most challenging tasks in the real world.

The goal of this research is to detect masks on faces that have been rotated and to keep tabs on people whose faces are partly obscured by clothing or other objects. There are several deep learning algorithms for object detection, including YOLO iterations, Fast R-CNN, Faster R-CNN, Region based R-CNN, SSD, and others. This study [13] refined the strategy by doing many rounds of YOLO in an effort to fill the knowledge gap. The authors created their own data collection called COVID-19-Mask. If individuals are hiding their identities behind masks, the technology will be able to tell the difference between similar masks in real life. We analyzed our dataset to see how YOLOv2, YOLOv3, YOLOv4, and OLOv5 fared in comparison. Our research led us to conclude that our suggested approach can accurately differentiate mask-wearers and non-mask-wearers in a variety of scenarios, including when the face is covered by a hand and when persons are wearing Diversified face masks, as well as when the mask is partly concealed and rotated.

This study is about [14] Modern Face Recognition Using Deep Learning's New Masked Face Dataset. The COVID-19 disease outbreak has significantly altered people's daily life. When speaking with an infected person, coughing, sneezing, or touching an infected object, COVID-19 can be disseminated. People abide by the rules by keeping their distance from one another and routinely donning a protective face mask in order to prevent viral infection. In this study, we create a fresh dataset of simulated masked faces that may be applied to masked face recognition tasks. We additionally restricted the dataset using the ArcFace-based system, one of the most well-liked modern facial recognition techniques, in order to assess the dataset's usability. This technique's accuracy is 99.11%. (FW).

The Deep Learning Strategy for Masked Face Recognition is discussed in this study [15]. We introduce a deep transfer learning approach to facial recognition using masks. Our masked face training data allowed us to find appropriate pretrained network models from which to build a deep neural network to solve an unrelated picture classification challenge. The training of the deep learning model was accelerated and its overall performance was enhanced by making effective use of transfer learning. We evaluated six different CNN models on this dataset. The models excel in terms of both learning speed and accuracy in identifying faces. Among the available models are SqueezeNet, GoogleNet, AlexNet, ResNet-50, VGG-16, and MobileNet-V2. The six models' validation accuracies ranged from 97.8 percent to a perfect one hundred. All six models, when tested on unseen data, accurately recognized each masked face with a high degree of certainty, showing a recognition rate of one hundred percent.

Real-Time Detection of Face Masks The integration of OpenCV with Deep Learning is the subject of this study [16]. The epidemic of COVID-19 has devastated healthcare systems throughout the world. Stopping the virus from spreading is now urgent. The practice of keeping one's distance from others, regularly washing one's hands, and wearing protective masks has gained widespread attention. The World Health Organization (WHO) suggests that anybody who wants to prevent the spread of the new coronavirus should wear a mask that encloses their mouth and nose. In this research, we use classifiers from the OpenCV library and the Viola-Jones method (also known as the Haar-Cascade algorithm) to identify people wearing masks. The collection included 3835 images of human faces, some of which were disguised. The results show that the trained model can identify face masks with a success rate of 98%.

This research impacts real-time face mask recognition software in busy places including schools, airports, and public events. This study [17] uses deep metric learning with FaceMaskNet-21 to recognize masked faces. Everyone must use masks to stop COVID-19 from spreading worldwide. The mask hides a person's nose, lips, and other visible features, making it hard to identify them and rendering security facial recognition systems useless. Our deep metric learning algorithm and FaceMaskNet-21 deep learning network recognize faces from still pictures, real-time video streams, and prerecorded video files. Testing accuracy of 88.92% was obtained with an execution time of less than 10 ms. The real-time masked face recognition capabilities of the system make it suitable for identifying persons in surveillance footage from public locations like shopping malls, banks, and ATMs. Our technology has the potential to be utilized in schools and universities for attendance as well as banks and other high-security institutions to enable only authorized persons to enter without having them to remove their masks.

Project [18] "Face Mask Detection Using Machine Learning" seeks to create a system that can analyze a person's face in a photograph and determine whether or not they are hiding their identity behind a mask. A face mask is necessary for protection from COVID. Now that the nation is getting ready to slowly reopen, face masks are a necessary part of our everyday existence. People and business activities need the use of face masks. Therefore, this program uses an embedded camera to identify masks. As a result of the rapid spread of COVID-19, international commerce and travel have been severely disrupted. Protection with face masks has advanced. In the near future, many establishments providing public services will mandate that customers wear masks before they may enter the building. Recognizing people behind masks is becoming more important in maintaining global civilization.

Using popular machine learning libraries like TensorFlow, Keras, OpenCV, and Scikit-learn, a concise strategy for achieving this objective is outlined in this paper [19]. The proposed approach detects the presence or absence of a mask over a given face in a picture. While monitoring, it can tell the difference between a live face and a mask. The method achieves up to 95.77% and 94.58% accuracy on two separate datasets, respectively. There is a need to detect masks while avoiding overfitting, thus we look at what values for the Sequential Convolutional Neural Network model's parameters provide the best results. The global health situation is dire due to the COVID-19 coronavirus epidemic. One of the best preventive methods is wearing a face mask in public, says the World Health Organization (WHO).

This research [20] aims to create a hybrid model for face mask identification by bringing together deep learning and conventional machine learning techniques. The proposed framework has two distinct sections. The first part extracts features using Resnet50. The second part classifies face masks using ensemble methods, decision trees, and vector machines for support (SVM). Three datasets with the faces obscured were utilized for this research. There are three different types of datasets: the Authentic Masked Face Collection (RMFD), the Synthetic Masked Face Dataset (SMFD), and the Labeled Faces in the Wild (LFW). The SVM classifier in RMFD achieved a testing accuracy of 99.64 percent. In SMFD tests, it performed at 99.49% accuracy, while LFW tests showed it to be 100% accurate.

2.3 Research Summary

This study compares two Convolutional Neural Networks, MobileNetV2 and Hog. With MobileNetV2, the average identification accuracy is around 94%. The suggested technique may be used to provide a secure system for detecting masked faces. We may utilize another dataset with many more photos that can be trained with VGG16 for further study over a longer period of time. This model can be used with alarms, SMS alert system, social distancing system This model may also be evaluated with various optimizers and adaptive learning strategies.

2.4 Tools and Software

We use Python language to continue our research. For coding purposes we use Google Colab Notebook as well as Pycharm IDE. We used pre-trained CNN algorithms. For visual presentation we used the Matplotlib library.

2.5 Scope of the Problem

This research is based on face recognition which will facilitate everyone. But there are lots of problems within. We are working with 15 different peoples for which our model can perfectly detect with greater accuracy. But when the dataset is bigger, then the time to recognize the faces can be decreased in this classification problem. Other than that face recognition under the mask is very complex and sometimes two people can be similar when wearing a mask. So, there can be some issues with recognizing the face properly.

2.6 Challenges

The most difficult part of this research is finding the right dataset. There are many datasets available online, however the majority of them are unbalanced. Even though the chosen dataset is unbalanced, we decided to utilize it anyhow since we wanted to enhance its output. The model's training presented further difficulties. Since our GPU setup was not that powerful, we used GPU acceleration to train our model in Google Colab.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Introduction

Several processes are taken into consideration in this session, starting with the pre-processing of the data, noise reduction, and transfer to the correct TensorFlow format for training. Once the training phase is complete, examine the validated photos and send them for testing. And we haphazardly display the test image.

3.2 System Design

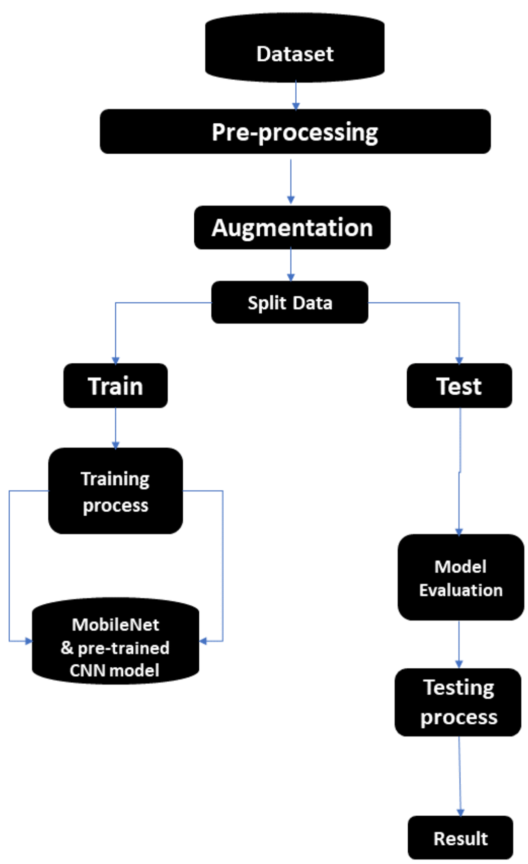

Fig 3.2.1: System design

This figure 3.2.1 represents how we design our system in order to perform our research work. First, we gather the data from different sources then split it according to the size of training a testing size which was 70% for training, 20% for testing and 10% for validation. Then we use Image Augmentation. After pre-processing the data, we trained our model and get the accuracy on the test set.

3.3 Data Collection Procedure

We gathered an image dataset for ten human masked faces and without mask face for face recognition. There are images for the training, validation, and testing for each individual class. The image sizes fall between 1.5 and 4 MB. To ensure that the images meet each network's necessary input dimensions, we resize them. To find the mask in the face, a second dataset was used. It is collected from Kaggle.

3.4 Split the data

The data set has two categories: mask and unmasked images and it contains 4000 images. Accordingly, we cut the dataset into two parts: eighty percent for training and twenty percent for testing. The training folder has 3200 images and the test folder has 800 images.

3.5 Data Pre-processing

Before putting the images from the two datasets through the models, we preprocessed them to give them the shape of 224x224x3. 4000 masked and unmasked images were combined into a balanced dataset as a result of this effort. Although there is representation from a variety of races, genders, and age groups in the dataset, the majority of the images are of male, white, and young people.

3.6 Augmentation

We used the data augmentation approach to enhance the volume of data that was processed. By altering or adjusting the spatial characteristics of the pictures, such as flipping them horizontally or vertically, rotating them, shifting them horizontally or vertically, and zooming them, new data is created. There are various alternative uses for this procedure that don't require more data. We can prevent the over-fitting of data by making several copies of the current data. Additionally, using data augmentation, if we balance the data in each class, then maybe the data in the overall collection will also be balanced.

3.7 Transfer Learning

Transfer learning is a method of machine learning that involves applying an algorithm or model to a new dataset after it has previously been trained with one particular dataset. Basically, this method is utilized when there is not enough data to fully train the model. In such an instance, it would be advantageous if we used a trained version of the model. Here we use a pre-trained CNN model and MobileNetV2 model as well.

3.8 Pre-trained Models

The Fig 3.8.1 describe the network architecture of the model. MobileNetV2 one of the first projects to develop CNN architectures that were easily implementable in mobile apps. One of the major developments is depth-wise separable convolutions, as seen below. A single diffusion kernel may be split into two halves using a technique called separable convolution. For instance, we get a 3x1 and a 1x3 kernel in place of a 3x3 kernel. The input image is scaled from 224 to 128 pixels. Due to the fact that MobileNet-V2 uses global average pooling rather than flattening, you can train it on 224x224 images and then use it on 128x128 images.

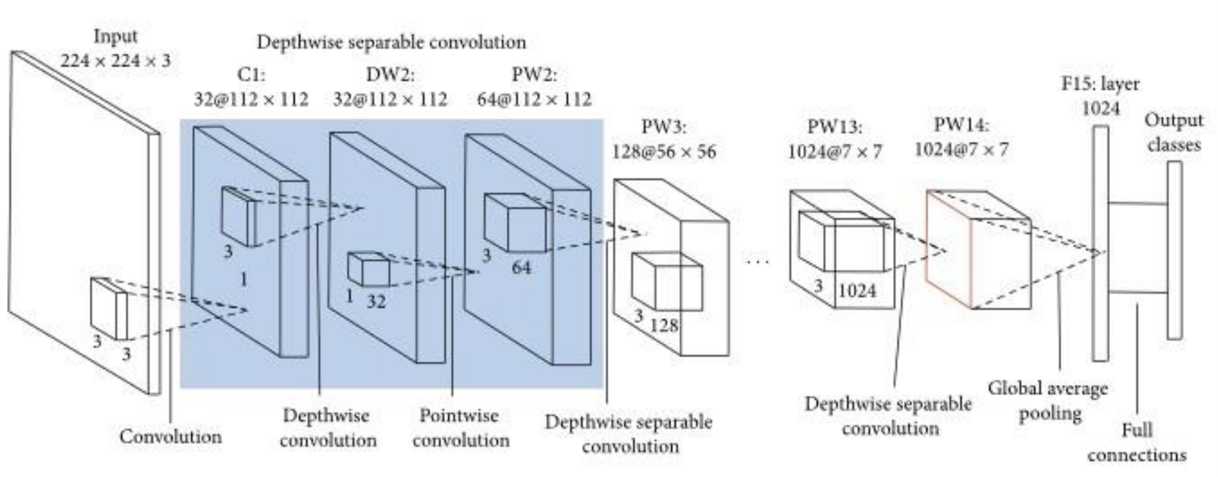

Fig 3.8.1: MobileNetV2 architecture [shorturl.at/fgvJK]

3.9 Training Model

The issue of face recognition may be broken down into a number of distinct but interconnected issues:

1. Find all the faces in the image by first looking at it.
2. Second, study every face carefully and remember that it represents the same person even if the lighting is bad or the angle is off.
3. Third, recognize the unique features of a face, such as the size of the eyes, the length of the face, etc., that may be used to distinguish between a given face and others.
4. The name of the person can then be determined by comparing the distinctive features of that face to all the others you are familiar with.

The human brain was built to automatically and swiftly carry out all of these activities. We are so good at recognizing faces that we frequently attribute them to inanimate things. Because computers are incapable of making such broad generalizations, we must educate them how to carry out each individual stage of the process. We need to build a pipeline in which each face recognition stage is completed separately and the results are passed on to the next. To elaborate, we want to combine a wide variety of machine learning strategies. The initial step in our process is facial recognition.

Face detection is a terrific addition to cameras. When a camera is able to identify people in a group, it may concentrate on getting everyone in the shot. In its place, we'll utilize it to pinpoint certain regions of the picture that should be sent through to the next processing step in the pipeline. Face recognition was not commonly utilized until the early 2000s, when Paul Viola and Michael Jones developed a method that worked quickly enough to be used on inexpensive cameras. However, there are now more trustworthy choices accessible. The method we'll be using is called Histogram of Oriented Gradients (HOG), and it was developed in 2005.

Histogram of Oriented Gradients: HOG is a straightforward yet effective feature descriptor. It is widely used for object detection, including cars, animals, and fruits, in addition to face detection. Because HOG uses the local intensity gradient distribution and edge direction to characterize object shape, it is reliable for object detection. HOG's central concept the picture is divided into small, linked cells. From this output the derived histogram of each cell is calculated and combine all histograms to form a feature vector, which creates a single, distinctive histogram for each face out of all smaller histograms. First, we'll turn our picture into black and white since identifying faces may be done without color information. The next step is a careful examination of the image's pixels, one by one. In this case, we're just interested in the neighboring pixels. Our objective is to determine how much darker a given pixel is compared to its immediate neighbors. After that, an arrow will be drawn in the direction of the opacity loss. By following that method for each pixel, we may replace each one with an arrow. Gradients are the arrows that show how the brightness of a picture gradually decreases toward its darkest areas. Although changing the pixels to gradients might seem random at first, there is a very good reason for doing so. Images of the same person that are extremely dark or extremely light will have entirely different pixel values if we analyze the pixels directly. However, if the brightness is only taken into account in one direction, Images that are incredibly black or highly brilliant will both depict the same thing. Saving the gradient for every pixel, on the other hand, provides far too much information. As a result, we will divide the picture into several 16x16 pixel squares. Within each square, we will count the number of gradients heading in each main direction (up, up-right, right, etc.). The arrow directions with the highest strength will then be utilized to replace that square in the picture.

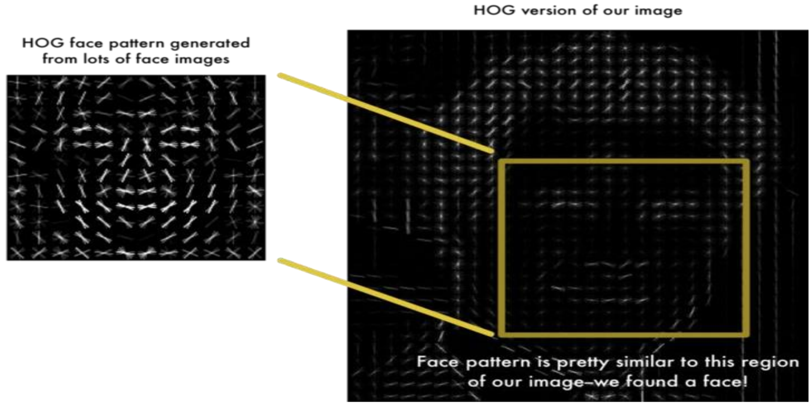

Fig 3.9.1: Simplified Image of Face using HOG [shorturl.at/crNW1]

The figure 3.9.1 represent the original picture is simplified to a representation that accurately depicts a face's essential features. The process of locating faces inside this HOG picture is as easy as pinpointing the area of our image that most closely fits a known HOG pattern that was created from a large number of training faces. This approach allows us to rapidly locate human faces in any image.

Posing and Projecting Faces

When we looked at the photo, we paid close attention to the people's faces. But now we have to deal with the problem that faces in different orientations seem quite different to computers. To make up for this, we'll be distorting each picture such that the mouth and eyes are in the same relative positions in each. Next, we'll be able to compare faces in a far less laborious manner. For this purpose, we'll use a facial landmark estimate method. Although there are several strategies for achieving this goal, we will use a technique created in 2014 by Vahid Kazemi and Josephine Sullivan.

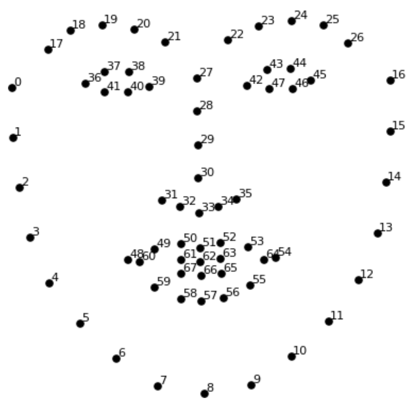

Fig 3.9.2: Facial map

The figure 3.9.2 represents 68 unique points on each person's face (called landmarks), such as the chin, the outside corner of each eye, the inner corner of each brow, and so on. Next, we'll using ML to teach a computer to locate those 68 unique spots on any face.

Since we already know where the eyes and mouth are located, we can simply adjust the image's orientation, size, and shape to put them in the center. Performing complex 3D warps would result in a ruined picture, thus we won't be doing that. Only simple transformations like scaling and rotation will be used, since they are the only ones that maintain parallel lines in the images. Now, we can put the eyes and mouth in the exact same place in the image no matter which way the face is turned. This will considerably increase the precision of our next move.

Encoding Faces

The next step in face recognition is to match the unknown face from Step 2 with all of the annotated photos in our database. We may assume it is the same individual if we locate a similar tagged face. anonymity-protected individual. It turns out, however, that there is a major flaw in such an approach. Due to the sheer volume of data on Facebook, billions of users and trillions of photos, it is difficult to manually check each new picture against the database of tagged faces.

That's just not feasible because of how long it would take. Facial recognition must take minutes rather than hours. There must be a way to measure certain basic characteristics of each face. Then, we can measure our mystery face the same way and see how it stacks up against the closest known face. A person's physical characteristics may be determined by counting features such as the number of ears, the distance between the eyes, the length of the nose, etc.

Researchers have found that letting the computer decide which metrics to collect on its own is the most accurate method. When determining which features of a face should be measured, deep learning outperforms humans. A Deep Convolutional Neural Network must be trained to tackle the challenge. For each face, we'll teach it to produce 128 measures. When training, three face pictures are shown simultaneously:

1. Add a practice face picture of a well-known individual.
2. Open a different photo of the same well-known individual.
3. Upload a photo of a distinct stranger.

After collecting data from the three images, the machine draws conclusions. The neural network is then fine-tuned such that its generated measures for 1 and 2 are somehow closer together, but its measurements for 2 and 3 are much further away. The neural network learns to exactly create 128 metrics for each person after going through this procedure millions of times with millions of photos of thousands of different individuals. The dimensions should be nearly the same for any 10 images of the same individual. Machine learning frequently discusses the notion of converting complex raw input, such as a photograph, into a list of artificially created numbers. Researchers at Google came up with the precise method for faces that we use in 2015, however there are several more methods that are quite similar.

Training a deep neural network to generate face embeddings requires a large data set and powerful processing resources. Even with a high-end NVidia Tesla video card, reasonable accuracy requires about 24 hours of continuous training time. But once trained, the network can provide metrics for just any face, even those it hasn't seen before! Like Fig 3.9.3. Therefore, you need only do this action once. Due to their prior efforts, OpenFace has provided us with a wealth of ready-to-use trained networks. To get the 128 facial measures, we need to feed our face pictures through their neural model. Fig 3.9.3 shows us about the different measurements of a face from an image.

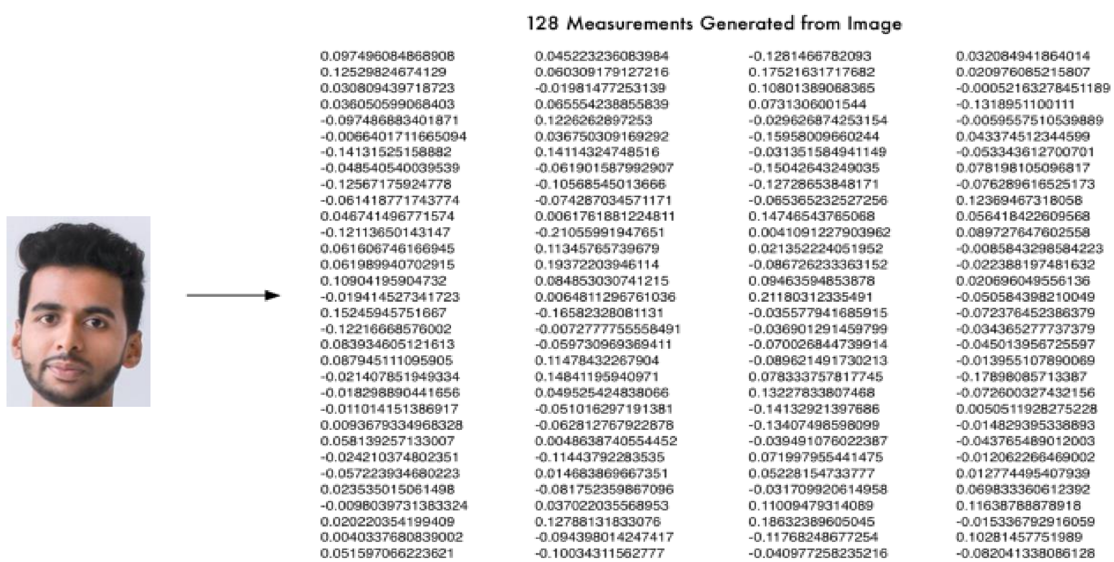

Fig 3.9.3: 128 facial measures from a face

Finding the person's name from the encoding

The process's final step is the one that is truly the simplest. Finding the person in our database of verified individuals whose dimensions are the closest to those in our test image is all that is required. Any simple machine learning classification method will work for this. Many other classification techniques may work, but we'll use a straightforward linear SVM classifier.
All we have to do is train a classifier that can identify the known individual who is the closest match based on measurements from a fresh test picture. It will take milliseconds to run this classifier. The classifier's output is the person's name.

In short, we do the following:

1. In order to make a smaller version of an image, the HOG encoding technique might be used. With this condensed version, you may zero in on the part of the picture that best conforms to a standard HOG modeling of a human face.
2. By identifying the key facial landmarks, determine the facial position. Use those markers to distort the image once we've located them so that the eyes and mouth are in the middle.
3. Use the centered face image as input to a neural network trained to evaluate facial features.
4. Find the individual whose measurements are the most similar to our face's measurements among all the faces we have previously measured. That's our opponent!

For Mask Detection

MobileNetV2 is what we utilize for mask detection. A convolutional neural network architecture called MobileNetV2 was created with mobile devices in mind. The bottleneck layers include residual keys, a concept borrowed from the inverted residual structure. The intermediate expansion layer employs light-depth-desired convolutions to filter out non-linear details. The MobileNetV2 architecture is composed of 19 additional bottleneck layers after the first shared convolutional layer with 32 filters. The method is divided into two sections: the first recognizes the existence of numerous faces in a particular picture or video sequence, and the second determines whether a face mask is there or not. The faces must have their bounding boxes evaluated before being sent to the second phase of the model to determine whether or not a mask is present. The labeled dataset is used to train the model's second component. To train our model, we used TensorFlow and Keras. The matching pictures are also reshaped for the model base in the first phase of training, which also involves storing all image labels in a NumPy array (224, 244, 3). MobileNetV2 with the supplied "ImageNet" weights is the fundamental model in use here. Fig 3.9.4 depicts the main methodology for face recognition process:

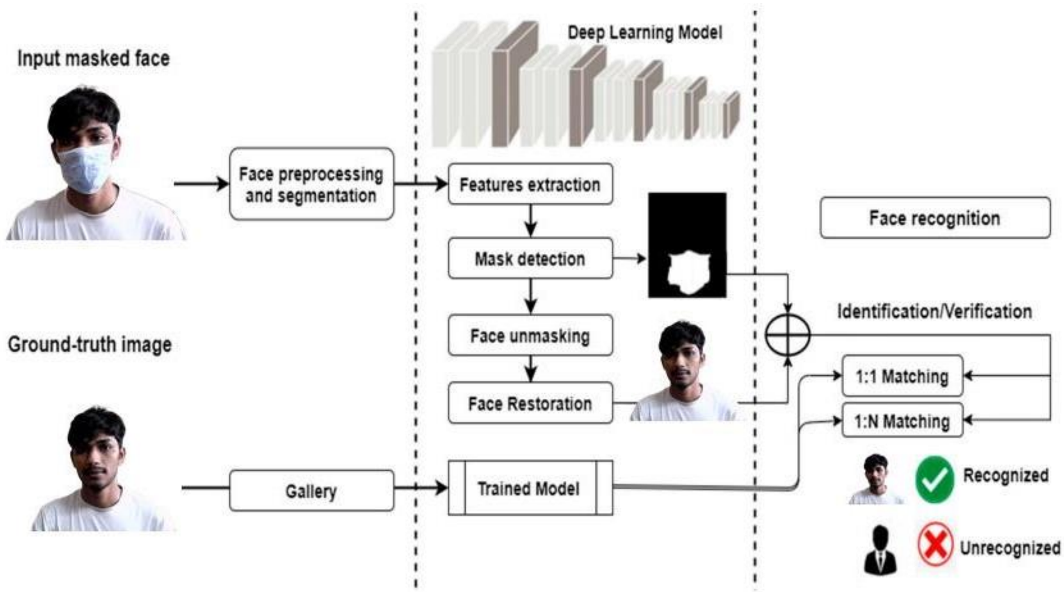

Fig 3.9.4: Methodology of our face recognition system

According to this Fig 3.9.4, the input will be a masked face. In our deep learning model, we extract the feature from the masked face and Restore it using our trained model. In the recognition part, in our database, we have the people’s image. We use SVM classifier that time to find out who was the person.

3.10 Model Architecture

The Fig 3.9.4 describes the MobileNetv2 structure, which is based on an inversion residual structure, with the input and output of the residual block being narrow bottleneck layers, as opposed to the normal residual model's usage of extended representations in the input. To filter features in the middle expansion layer, MobileNet v2 employs shallow depth wise convolutions. To maintain the representational power, non-linearities in the thin layers were also minimized.

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 112, 112, 32  864         ['input_1[0][0]']
                                )

 bn_Conv1 (BatchNormalization)  (None, 112, 112, 32  128         ['Conv1[0][0]']
                                )

 Conv1_relu (ReLU)              (None, 112, 112, 32  0           ['bn_Conv1[0][0]']
                                )

 expanded_conv_depthwise (Depth  (None, 112, 112, 32  288        ['Conv1_relu[0][0]']
 wiseConv2D)                    )

 expanded_conv_depthwise_BN (Ba  (None, 112, 112, 32  128        ['expanded_conv_depthwise[0][0]']
 tchNormalization)              )

 expanded_conv_depthwise_relu (  (None, 112, 112, 32  0          ['expanded_conv_depthwise_BN[0][0
 ReLU)                          )                                ]']

 expanded_conv_project (Conv2D)  (None, 112, 112, 16  512        ['expanded_conv_depthwise_relu[0]
                                )                                [0]']

 expanded_conv_project_BN (Batc  (None, 112, 112, 16  64         ['expanded_conv_project[0][0]']
 hNormalization)                )

 block_1_expand (Conv2D)        (None, 112, 112, 96  1536        ['expanded_conv_project_BN[0][0]'
                                )                                ]

 block_1_expand_BN (BatchNormal  (None, 112, 112, 96  384        ['block_1_expand[0][0]']
 ization)                       )

==================================================================================================
Total params: 2,422,210
Trainable params: 164,226
Non-trainable params: 2,257,984
```

Fig 3.10.1: Model architecture

3.11 Parameter

If we examine our model we find loss parameters. Total parameter in MobileNetV2 is about 2,422,210. Where total trainable parameter = 164,226 and Non-trainable parameter is 2, 257, 984. So, the loss of parameters is very negligible.

3.12 Hyper parameter Tuning

The learning process of a model can be controlled by hyper parameters. It is responsible for the change of performance of the model. Below is a list of hyper parameters we used in our research:

Table 3.12.1: Hyper parameter of our model

No	Hyper Parameters	Tuning
1	EPOCHS	20
2	Batch size	32
3	Width, Height	224, 224
4	Optimizer	Adam
5	Learning rate	1e-4
6	Horizontal and vertical flip	True
7	rescale	1/255
8	rotation_range	20
9	width_shift_range	0.2
10	height_shift_range	0.2
11	shear_range	0.15
12	zoom_range	0.15
13	fill_mode	nearest
14	Dropout	0.5
15	Train, Test split	80%, 20%

The table 3.12.1 represents the Hyper parameter of our created model. Here we used 20 epochs, batch size 32, Image size 224x224, Adam as optimizer and learning rate as 1e-4. We also use different augmentations to increase the data. Our dropout was 0.5 and we split our data to 80% for training and 20% for testing.

CHAPTER 4

EXPERIMENTAL RESULT AND DISCUSSION

4.1 Introduction

The discovery is discussed in this chapter. It is the final step to work with real data. We achieved quite good accuracy with our models.

4.2 Result/Output

Model Evaluation:

One of the most crucial steps in the training process is this one. We can genuinely comprehend how our model is functioning by analyzing the models like VGG16, VGG19. Each model's accuracy during training and validation is shown as an upward trending graph. The Fig 4.2.1 describes the model accuracy where training accuracy 98% and validation accuracy 95% for MobileNet. On the other hand, Fig 4.2.2 describes the model loss of training and validation set.

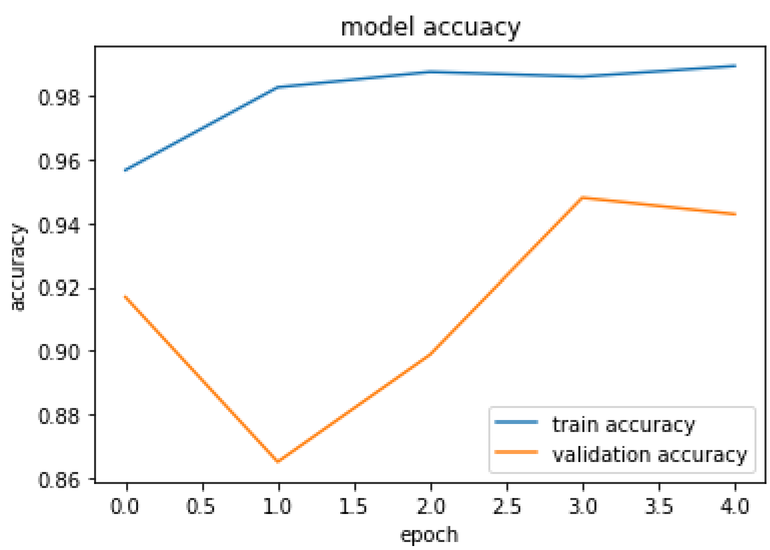

Fig 4.2.1: Model accuracy for MobileNetV2

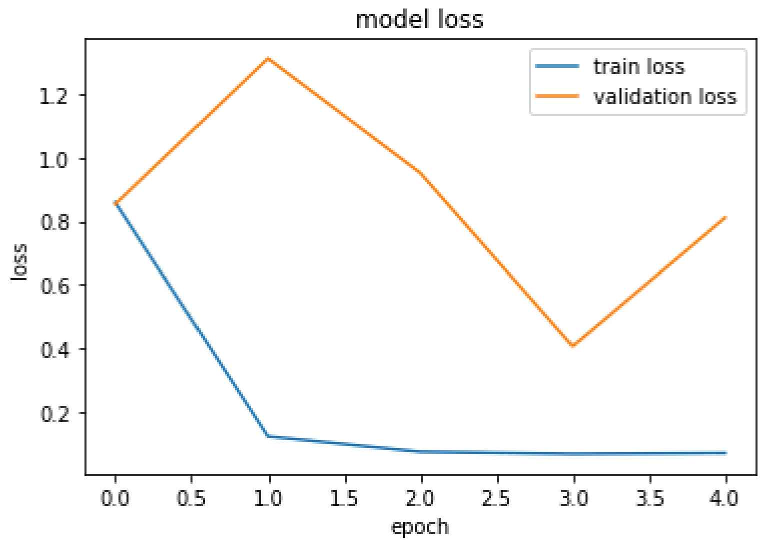

Fig 4.2.2: Model loss for MobileNetV2

The Fig 4.2.3 describes the model accuracy where accuracy 88% for VGG19. On the other hand, Fig 4.2.4 describes the model loss of training and validation set for the model.

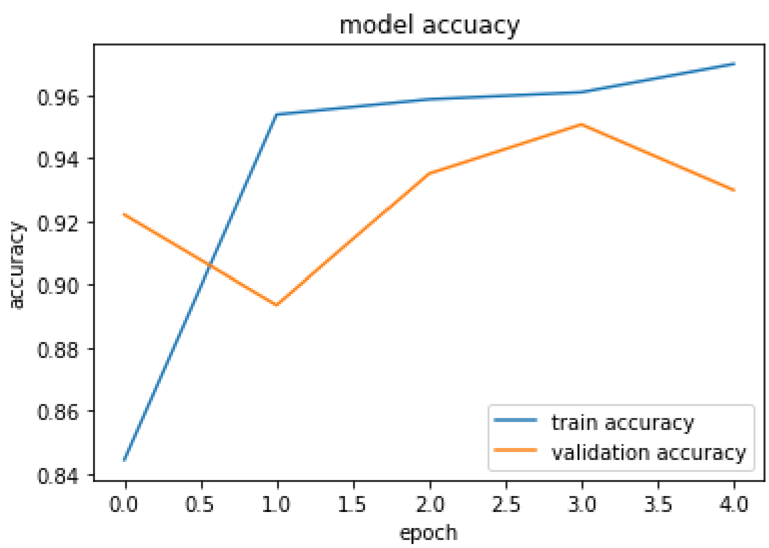

Fig 4.2.3: Model accuracy for VGG19

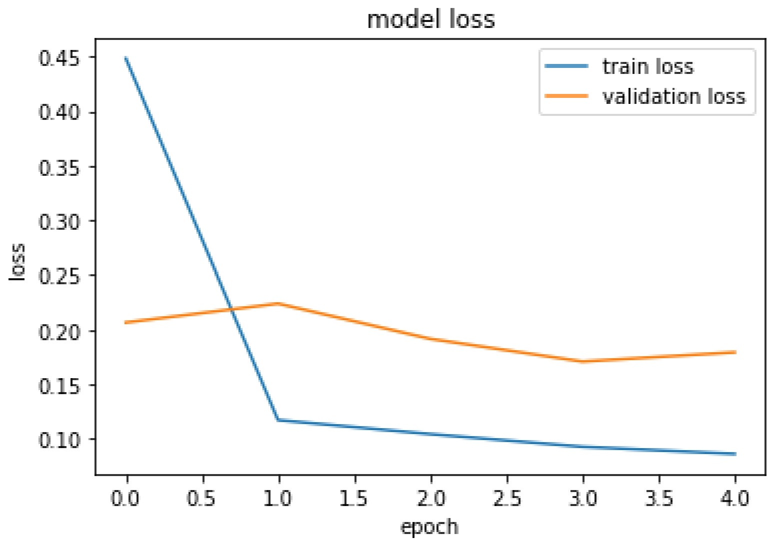

Fig 4.2.4: Model loss for VGG19

Confusion matrix:

Confusion matrix is way to measure the performance of the classifier models. The Fig 4.2.4 shows the confusion matrix of our model. From this, we can easily visualize how much data we identify correctly and how much data we identify wrong.

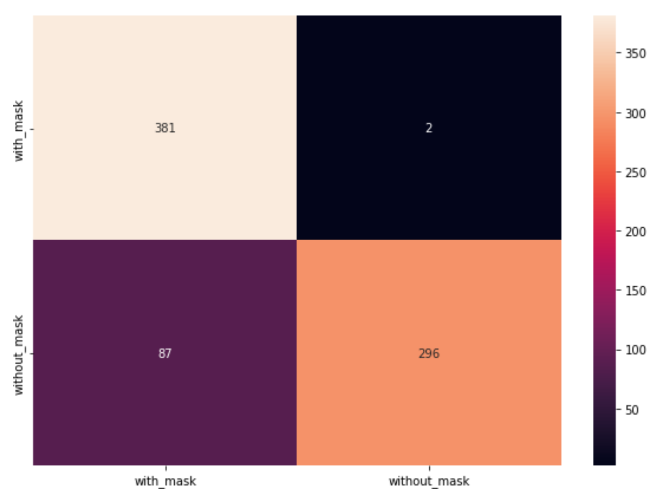

Fig 4.2.5: Confusion Matrix for MobileNetV2

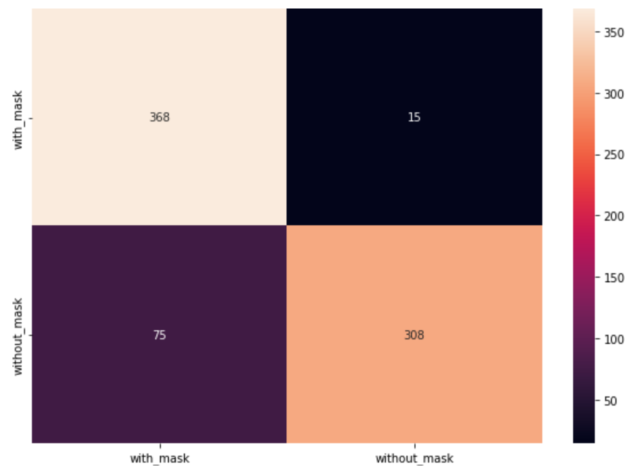

Fig 4.2.6: Confusion Matrix for VGG19

From here we can decide that MobileNetV2 performed better that VGG19 as the accuracy of the model is 94% > 88%. On the other hand, it is easy to use in camera's because of the same size of the model. Also it is a lightweight deep neural network with fewer parameters and higher classification accuracy. Overall it is wise to use MobileNetV2 over VGG19.

Classification Report:

The predictions made by a classification algorithm may be evaluated with the help of a classification report. The Fig 4.2.4 shows the classification report of our model. how many guesses came true and how many didn't? Specific examples of True Positives, True Negatives, False Positives, and False Negatives being used to forecast the metrics of a classification report are provided below:

```
24/24 [==============================] - 35s 1s/step
              precision    recall  f1-score   support

   with_mask       0.94      0.95      0.94       383
without_mask       0.95      0.94      0.94       383

    accuracy                           0.94       766
   macro avg       0.94      0.94      0.94       766
weighted avg       0.94      0.94      0.94       766
```

Fig 4.2.7: Classification report

Precision:

Precision indicates the percentage of correctly anticipated positive classifications.

Predictive Accuracy = True Positive / (True Positive + False Positive).

Recall:

It reflects how much we anticipated accurately out of all the positive classifications. The greater the value, the higher the model's quality.

Recall = True positive / (True Positive + False Negative)

F1-score:

The F-score is useful because it allows us to evaluate both recall and precision simultaneously. The Arithmetic Mean is replaced with the more musically-sounding Harmonic Mean.

F1-score = (2*recall*precision) / (recall + precision)

CHAPTER 5

IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

5.1 Impact on Society

Facial recognition is a biometrics application that analyzes a person's facial patterns, structure, and contours to identify that person. Consider scanning a person's face like a fingerprint. The impact of the work in society are given below:

Businesses can now keep sensitive areas more secure and control access to them by utilizing Masked face recognition system. Airports and military bases, for example, use this technology to ensure that no one enters restricted zones without proper clearance. This also protects these areas from potential terrorist attacks by notifying the appropriate authorities of any individual who may be on a watch list.

5.2 Impact on Environment

As we developed this Research based project on account with COVID-19, it can help us to provide an environment less contaminated with this virus. Because the virus spread rate increases if we don't wear mask. It ensures that everyone should wear mask all the time. Not only for COVID-19, it can help us to prevent other contaminated diseases as well which can be controlled by wearing mask like (Seasonal influenza).

5.3 Ethical Aspects

As facial recognition technology advances, it attracts cyber criminals who seek to exploit these developing systems. Despite the numerous advantages of facial recognition technology, it is critical to weigh the risks and drawbacks in order to keep your personal information safe. Like
a significant disadvantage of facial recognition technology is the threat to individual privacy. People are not happy about having their faces recorded and stored in a database for unknown future use. Databases containing facial recognition data may be compromised. Hackers have gained access to databases containing facial scans collected and used by banks, police departments, and

defense contractors. Facial recognition technology can be used by criminals to commit crimes on defenseless people. To perpetrate identity fraud, they can gather people's private information, including pictures and videos taken via facial scans and kept in databases. With this knowledge, a thief may open bank accounts or credit cards in the victim's name, or even use the victim's identity to commit crimes.

5.4 Sustainability Plan

As we developed this project for the masked face recognition, we want to implement it on various organizations. It is much more convincible for the organization to implement our project in their areas because we deliver both face recognition as well as the masked face. It can help them not only for stopping the viruses but also if there is any incident of crime, they can recognize the criminal as well if the face of the criminals are stored on the police database. So, we provide both the health and financial security in this aspect. That's why we think our plan will successful as well as beneficial for us.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

5.5 Summary of the Study

Using a combination of a convolutional neural network and MoblieNetV2, we're attempting to portray a machine learning-based approach to the detection of masked and unmasked faces. The following is a synopsis of our whole project.

1st stage

- Literature review
- Data collection and preprocessing
- Divided data into train and test set and Feature reduction.

2nd stage

- Select the pre-trained models
- Compile the model and fit the model
- Training the dataset using CNN and MobileNetV2 model.

3nd stage

- Evaluate the model
- Take steps for improvements

Final stage

- Find the predicted test images & accuracy.
- Find tuning the model operators to get better accuracy and solution.

5.6 Conclusion

According to our findings, the average recognition accuracy with MobileNet is over 94%. The suggested technique may be used to provide a secure system for detecting masked faces. We may utilize another dataset with many more photos that can be trained using VGG16 over a longer period of time for additional study. This model can be used with alarms. SMS alert system, social distancing system This model may also be evaluated with various optimizers and adaptive learning strategies.

5.7 Recommendations for Further Study

In this research, we have shown how to recognize the people under the mask and we try to detect masks as well. We are trying to publish a journal and our 80% report has been done. Hope we will be able to publish our journal. And we also want to develop a recognition system in our campus to secure the whole area.

Reference:

[1] Mehreen Fatima1*, S. A. (2022, February 18). Machine Learning for Masked Face Recognition in COVID-19 Pandemic Situation. *International Information and Engineering Technology Association*, *9*(1), 283-289.

[2] Pranad Munjal*, V. R. (2021). Real-Time Face Mask Detection using Deep Learning. *Journal of Technology Management for Growing, 12*(1), 25-31.

[3] Wadii Boulila, M. A. (n.d.). A Deep Learning-based Approach for Real-time.

[4] Ahmad Alzu'bi 1, *. F.-H. (2021, October 31). Masked Face Recognition Using Deep Learning: A Review. *electronics*, *10*(2666), 1-35.

[5] Rui Zhu, 1. K. (2021, October 4). Masked Face Detection Algorithm in the Dense Crowd Based on. *Wireless Communications and Mobile Computing, 2021*, 1-8.

[6] Vibhaakar Sharma, S. G. (n.d.). Masked Face Recognition.

[7] Udit Upadhyay1, 2. D. (2021, November 4). Face Mask Detection Using Convolutional Neural Network (CNN). *EasyChair Preprint*, 1-9.

[8] Anchal Gupta1, A. G. (2022). Face Mask Detector Using Machine Learning Applications. *International Journal of Special Education · February 2022*, *37*(3), 3447-3454.

[9] Shiming Ge1, J. L. (n.d.). Detecting Masked Faces in the Wild with LLE-CNNs. *IEEE Xplore*, 2682-2690.

[10] 1DHEERAJ KD, 2. G. (2021, July). A REVIEW ON FACE-MASK DETECTION USING MACHINE LEARNING. *INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT), 9*(7), 664-669.

[11] Aditya Chichghare1, ,. N. (2021, June). Face Mask Detection using Machine Learning. *INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY*, *8*(1), 1124-1127.

[12] Mishra, A. S. (2021, August). Detecting Mask in human Faces using Machine Learning Approach. *International Journal of New Technology and Research (IJNTR), 7*(8), 73-75.

[13] 1Hiral Talsaniya, 2. V. (2021, JUNE). Face mask detection using Deep Learning Approach. *International Journal for Research in Engineering Application & Management (IJREAM), 7*(3), 234-239.

[14] Vandet Pann, H. J. (2021). Modern Face Recognition using New Masked Face Dataset Generated by Deep Learning. 647-650.

[15] Maad Shatnawi, N. A. (2022). Deep Learning Approach for Masked Face Identification. *(IJACSA) International Journal of Advanced Computer Science and Applications, 13*(6), 296-305.

[16] Hruthik S. Upendra1, S. S. (2021, September 06-07). Real-Time Face Mask Detection using OpenCV and Deep Learning. *nternational Conference on Emerging Technologies: AI, IoT, and CPS for Science & Technology Applications*.

[17] Mehendale1, R. G. (2021, December 24). Masked-face recognition using deep metric learning and FaceMaskNet-21. *Springer Nature 2022*.

[18] Mr.Kalla.Kiran1, B. V. (n.d.). FACE MASK DETECTION USING MACHINE LEARNING .

[19] Arjya Das, M. W. (2020). Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. *India Council International Conference (INDICON).*

*[20] Mohamed Loey a, *. ,. (2020, July 28). A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. 2020 Elsevier Ltd., 1-11.*

APPENDIX

Appendix A: Research Reflection

During our research, we face many kinds of problems. We had to learn Convolutional Neural Networks and deep learning techniques and need to study about. When we work in this research we face this type of problem.

Appendix B: Related issue

Imported Library Functions:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import os
```

Image Augmentation and Preprocessing:

```
aug = ImageDataGenerator(
  rotation_range=20,
  zoom_range=0.15,
  width_shift_range=0.2,
  height_shift_range=0.2,
  shear_range=0.15,
  horizontal_flip=True,
  fill_mode="nearest")
```

Fit the model:

```
print("[INFO] training head...")
H = model.fit(
  aug.flow(trainX, trainY, batch_size=BS),
  steps_per_epoch=len(trainX) // BS,
  validation_data=(testX, testY),
  validation_steps=len(testX) // BS,
  epochs=EPOCHS)
```

Training the model:

```
[INFO] training head...
Epoch 1/20
95/95 [==============================] - 89s 902ms/step - loss: 0.3713 - accuracy: 0.8603 - val_loss: 0.1431 - val_accuracy: 0.9739
Epoch 2/20
95/95 [==============================] - 96s 1s/step - loss: 0.1355 - accuracy: 0.9644 - val_loss: 0.0807 - val_accuracy: 0.9817
Epoch 3/20
95/95 [==============================] - 86s 909ms/step - loss: 0.0944 - accuracy: 0.9756 - val_loss: 0.0651 - val_accuracy: 0.9791
Epoch 4/20
95/95 [==============================] - 86s 904ms/step - loss: 0.0776 - accuracy: 0.9763 - val_loss: 0.0505 - val_accuracy: 0.9870
Epoch 5/20
95/95 [==============================] - 86s 906ms/step - loss: 0.0640 - accuracy: 0.9809 - val_loss: 0.0456 - val_accuracy: 0.9844
Epoch 6/20
95/95 [==============================] - 86s 905ms/step - loss: 0.0589 - accuracy: 0.9835 - val_loss: 0.0500 - val_accuracy: 0.9804
Epoch 7/20
95/95 [==============================] - 86s 903ms/step - loss: 0.0505 - accuracy: 0.9858 - val_loss: 0.0396 - val_accuracy: 0.9870
Epoch 8/20
95/95 [==============================] - 86s 904ms/step - loss: 0.0478 - accuracy: 0.9855 - val_loss: 0.0338 - val_accuracy: 0.9883
Epoch 9/20
95/95 [==============================] - 87s 911ms/step - loss: 0.0493 - accuracy: 0.9862 - val_loss: 0.0405 - val_accuracy: 0.9857
Epoch 10/20
95/95 [==============================] - 86s 907ms/step - loss: 0.0427 - accuracy: 0.9855 - val_loss: 0.0331 - val_accuracy: 0.9909
Epoch 11/20
95/95 [==============================] - 86s 908ms/step - loss: 0.0379 - accuracy: 0.9885 - val_loss: 0.0283 - val_accuracy: 0.9909
Epoch 12/20
95/95 [==============================] - 86s 903ms/step - loss: 0.0347 - accuracy: 0.9908 - val_loss: 0.0345 - val_accuracy: 0.9870
Epoch 13/20
95/95 [==============================] - 86s 903ms/step - loss: 0.0337 - accuracy: 0.9908 - val_loss: 0.0284 - val_accuracy: 0.9909
Epoch 14/20
95/95 [==============================] - 86s 906ms/step - loss: 0.0331 - accuracy: 0.9911 - val_loss: 0.0346 - val_accuracy: 0.9870
Epoch 15/20
95/95 [==============================] - 87s 915ms/step - loss: 0.0277 - accuracy: 0.9898 - val_loss: 0.0232 - val_accuracy: 0.9922
Epoch 16/20
95/95 [==============================] - 87s 913ms/step - loss: 0.0245 - accuracy: 0.9914 - val_loss: 0.0327 - val_accuracy: 0.9857
Epoch 17/20
95/95 [==============================] - 87s 913ms/step - loss: 0.0272 - accuracy: 0.9895 - val_loss: 0.0222 - val_accuracy: 0.9935
Epoch 18/20
95/95 [==============================] - 87s 911ms/step - loss: 0.0299 - accuracy: 0.9911 - val_loss: 0.0321 - val_accuracy: 0.9883
Epoch 19/20
95/95 [==============================] - 87s 912ms/step - loss: 0.0296 - accuracy: 0.9911 - val_loss: 0.0265 - val_accuracy: 0.9909
Epoch 20/20
95/95 [==============================] - 86s 904ms/step - loss: 0.0299 - accuracy: 0.9914 - val_loss: 0.0291 - val_accuracy: 0.9883
```

Test Result:

```
24/24 [==============================] - 190s 8s/step - loss: 0.7631 - accuracy: 0.9426
Accuracy: 0.942558765411377
Loss: 0.7630659937858582
```

Plagiarism

Submission date: 23-Jan-2023 12:41AM (UTC+0600)
Submission ID: 1968327677
File name: Face_recognition_report.pdf (1.04M)
Word count: 9842
Character count: 51773

mask

ORIGINALITY REPORT

15%	12%	7%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
2	Submitted to Daffodil International University Student Paper	2%
3	Submitted to Intercollege Student Paper	1%
4	www.researchgate.net Internet Source	1%
5	link.springer.com Internet Source	1%
6	Rui Zhu, Kangning Yin, Hang Xiong, Hailian Tang, Guangqiang Yin. "Masked Face Detection Algorithm in the Dense Crowd Based on Federated Learning", Wireless Communications and Mobile Computing, 2021 Publication	<1%
7	Shiming Ge, Jia Li, Qiting Ye, Zhao Luo. "Detecting Masked Faces in the Wild with LLE-CNNs", 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017	<1%

Publication

8	iieta.org Internet Source	<1%
9	Submitted to University of Hertfordshire Student Paper	<1%
10	irep.ntu.ac.uk Internet Source	<1%
11	Submitted to Southern University And A & M College Student Paper	<1%
12	Maad Shatnawi, Nahla Almenhali, Mitha Alhammadi, Khawla Alhanaee. "Deep Learning Approach for Masked Face Identification", International Journal of Advanced Computer Science and Applications, 2022 Publication	<1%
13	www.internationaljournalofspecialeducation.com Internet Source	<1%
14	ijream.org Internet Source	<1%
15	www.kdnuggets.com Internet Source	<1%
16	"Advances in Data Computing, Communication and Security", Springer Science and Business Media LLC, 2022 Publication	<1%

17	Submitted to NIIT University Student Paper	<1 %
18	Submitted to University of Bedfordshire Student Paper	<1 %
19	easyprojectmaterials.com.ng Internet Source	<1 %
20	www.ijraset.com Internet Source	<1 %
21	Mengyue Geng, Peixi Peng, Yangru Huang, Yonghong Tian. "Masked Face Recognition with Generative Data Augmentation and Domain Constrained Ranking", Proceedings of the 28th ACM International Conference on Multimedia, 2020 Publication	<1 %
22	Submitted to University of Leeds Student Paper	<1 %
23	ijsrset.com Internet Source	<1 %
24	thesai.org Internet Source	<1 %
25	jglobal.jst.go.jp Internet Source	<1 %
26	www.hindawi.com Internet Source	<1 %

27	repositorio.uam.es Internet Source	<1 %
28	www.degruyter.com Internet Source	<1 %
29	Rahaf Alturki, Maali Alharbi, Ftoon AlAnzi, Saleh Albahli. "Deep learning techniques for detecting and recognizing face masks: A survey", Frontiers in Public Health, 2022 Publication	<1 %
30	bradscholars.brad.ac.uk Internet Source	<1 %
31	ijeer.forexjournal.co.in Internet Source	<1 %
32	scholar.archive.org Internet Source	<1 %
33	www.globalresearch.ca Internet Source	<1 %
34	www.ijitee.org Internet Source	<1 %
35	www.paperdigest.net Internet Source	<1 %
36	"Intelligent Techniques in Signal Processing for Multimedia Security", Springer Science and Business Media LLC, 2017 Publication	<1 %

37	medinform.jmir.org Internet Source	<1 %
38	www.koreascience.or.kr Internet Source	<1 %
39	"Image Analysis for Moving Organ, Breast, and Thoracic Images", Springer Science and Business Media LLC, 2018 Publication	<1 %
40	Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, Nour Eldeen M. Khalifa. "A Hybrid Deep Transfer Learning Model with Machine Learning Methods for Face Mask Detection in the Era of the COVID-19 Pandemic", Measurement, 2020 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		