5-2021

Facial Recognition and Face Mask Detection Using Machine Learning Techniques

Mira M. Boulos
Montclair State University

ABSTRACT

Facial recognition, as a biometric system, is a crucial tool for the identification procedures. When using facial recognition, an individual's identity is identified using their unique facial features. Biometric authentication system helps in identifying individuals using their physiological and behavioral features. Physiological biometrics utilize human features such as faces, irises, and fingerprints. In contrast, behavioral biometric rely on features that humans do, such as voice and handwritings. Facial recognition has been widely used for security and other law enforcement purposes. However, since COVID-19 pandemic, many people around the world had to wear face masks. This thesis introduces a neural network system, which can be trained to identify people's facial features while half of their faces are covered by face masks. The Convolutional Neural Network (CNN) model using transfer learning technique has achieved remarkable accuracy even the original dataset is very limited. One large Face mask detection dataset was first used to train the model, while the original much smaller Face mask detector dataset was used to adapt and fine-tune this model that was previously generated. During the training and testing phases, network structures, and various parameters were adjusted to achieve the best accuracy results for the actual small dataset. Our adapted model was able to achieve a 97.1% accuracy.

Keywords: Biometrics, Facial Recognition, Face Mask Detection, CNN, Transfer learning.

MONTCLAIR STATE UNIVERSITY

Facial Recognition and Face Mask Detection Using Machine Learning Techniques

by

Mira M. Boulos

A Master's Thesis Submitted to the Faculty of

Montclair State University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

May 2021

College of Science and Mathematics

Department of Computer Science

Thesis Committee:

[redacted]

Dr. Michelle Zhu

Thesis Sponsor

[redacted]

Dr. Tianyang Wang

Committee Member

[redacted]

Dr. Weitian Wang

Committee Member

[redacted]

Dr. Jiayin Wang

Committee Member

FACIAL RECOGNITION AND FACE MASK DETECTION USING MACHINE LEARNING TECHNIQUES

A THESIS

Submitted in partial fulfillment of the requirements

For the degree of Master of Science

by

Mira M. Boulos

Montclair State University

Montclair, NJ

2021

ACKNOWLEDGMENTS

First and foremost, glory and grace to the almighty God for His blessings throughout the completion of my dissertation.

I would also like to express my deep and genuine gratefulness to my super advisor, Dr. Michelle Zhu. Thank you for allowing me to do the thesis and providing invaluable guidance throughout this thesis. Without your continuous support, patience, enthusiasm, and immense knowledge, I would not be able to finish my master's thesis.

Additionally, I would like to thank the rest of my thesis committee members: Dr. Tianyang Wang, Dr. Jiayin Wang, and Dr. Weitian Wang for their encouragement and insightful comments.

My sincere appreciations also go to my classmate, Aseel Aloweiwi, for all the unsleeping nights we had while working together. Thank you for being the perfect school partner throughout the completion of the Degree of Master of Science. I am grateful to you for always being there for me, encouraging me, and pushing me further. With you being here, this journey became so much better.

Most significantly, my success, the completion of my master's program, and dissertation would not have been achievable without the encouragement and support of my parents: Mamdouh Boulos and Mariam Rizkalla. Thank you for your love, compassion, and sacrifices for providing me with a better future. I am exceedingly grateful to my husband, Miller Khalil, for his love, sympathetic and continuous support. You are the source of my encouragement to complete my master's program and thesis. Thank you for always believing in me and for showing how proud you are of my achievements. Also, my sincere appreciation goes to my brothers, Peter Boulos and Andrew Boulos, for their help and guidance.

Table of Contents

ABSTRACT ---------- i

ACKNOWLEDGMENTS ----------v

Table of Figures ----------x

CHAPTER 1 ----------1

INTRODUCTION ----------1

1.1 Biometric Authentication System Background ----------*1*

1.2 Advantages of Using Biometric Authentication System ----------*2*

1.2.1 Biometric Authentication Uses Unique Data ----------2

1.2.2 Biometric Authentication Is Convenient to Use ----------3

1.2.3 Biometric Authentication Supports Multi-Factor Authentication ----------3

1.3 Facial Recognition as Biometric Security ----------*4*

CHAPTER 2 ----------7

RELATED WORKS ----------7

CHAPTER 3 ----------9

Facial Recognition System ----------9

3.1 Facial Recognition Structure ----------*9*

3.1.1 Face Detection ----------9

3.1.2 Face Alignment ----------9

3.1.3 Feature Extraction ---- 10

3.1.4 Face Recognition ---- 11

3.2 Facial Recognition Accuracy ---- 12

CHAPTER 4 ---- 14

FACIAL RECOGNITION THREATS AND COUNTERMEASURES ---- 14

4.1 Threats ---- 14

4.1.1 3-D Printing Face Mask ---- 14

4.1.2 Makeup Techniques ---- 14

4.1.3 Face Masks ---- 15

4.2 Countermeasures ---- 16

CHAPTER 5 ---- 17

DATA SOURCE ---- 17

5.1 Dataset Description ---- 17

5.1.1 Face Mask Detection Dataset ---- 18

5.1.2 Face Mask Detector Dataset ---- 21

5.2 Data pre-processing ---- 24

CHAPTER 6 ---- 26

METHODOLOGY AND EXPERIMENTAL SETUP ---- 26

6.1 Supervised Learning ---- 26

6.2 Convolutional Neural Network ---- *26*

6.2.1 Convolutional Layer ---- 27

6.2.2 Max Pooling ---- 28

6.2.3 Fully Connected Layer ---- 29

6.3 Training ---- *30*

6.3.1 Adam Optimizer ---- 31

6.3.2 MSELoss ---- 31

6.3.3 Running Epochs ---- 31

6.4 Testing ---- *32*

6.5 Transfer Learning ---- *33*

6.5.1 Training and Fine-tuning ---- 33

6.6 Testing ---- *34*

CHAPTER 7 ---- 36

RESULTS AND DISCUSSION ---- 36

7.1 Matching Predicted Classes with Real Classes ---- *36*

7.2 Pre-trained Model Training vs. Testing Results ---- *38*

7.3 Pre-trained Model vs. Transfer Learning Model ---- *40*

7.4 Predicting Real-World Images ---- *42*

7.5 Experiment 2 ---- *43*

7.5.1 Experiment 2 Pre-trained Model Training and Testing ---- 44

7.5.2 Experiment 2 Transfer Learning Training and Testing Results ---- 45

7.6 Experiment 1 and Experiment 2 Comparison ---------- 45

CHAPTER 8 ---------- 47

CONCLUSION AND FUTURE WORK ---------- 47

Bibliography ---------- 48

Table of Figures

Figure 1: Biometric Types [2] ---------- 2

Figure 2: Multi-Factor Authentication [5] ---------- 4

Figure 3: Law Enforcement using Face Recognition [8] ---------- 5

Figure 4: Airports Using Facial Recognition [10] ---------- 6

Figure 5: Facial Recognition System Structure [17] ---------- 9

Figure 6: Feature Extraction [19] ---------- 10

Figure 7: Identification and Verification in Face Recognition [21] ---------- 11

Figure 8: Percentage of FRR and FAR [23] ---------- 13

Figure 9: 3-D Printing Face Mask [24] ---------- 14

Figure 10: An example of Makeup Techniques to Be Used as a Threat for Facial Recognition [25] ---------- 15

Figure 11: Sample Images of People Who Wear Face Masks (Face Mask Detection Dataset) ---------- 18

Figure 12: Sample Images of People Who Do Not Wear Face Masks (Face Mask Detection Dataset) ---------- 19

Figure 13: Classes Distribution of Face Mask Detection Dataset ---------- 19

Figure 14: Distribution of with_mask Class Image Sizes ---------- 20

Figure 15: Distribution of without_mask Class Image Sizes ---------- 20

Figure 16: Sample Images of People Who Wear Face Masks (Face Mask Detector) ---------- 21

Figure 17: Sample Images of People Who Do Not Wear Face Masks (Face Mask Detector) ---------- 22

Figure 18: Sample Images of People Who Wear Face Masks but Incorrectly (Face Mask Detector) ---------- 22

Figure 19: Classes Distribution of Face Mask Detector Dataset ---------- 23

Figure 20: Distribution of with_mask Class Image Sizes ---------- 23

Figure 21: Distribution of without_mask Class Image Sizes ---------- 24

Figure 22: Distribution of incorrect_mask Class Image Sizes ---------- 24

Figure 23: Example of Splitting Data by 90:10 Ratio ---------- 25

Figure 24: Convolutional Neural Network Architecture ---------- 27

Figure 25: 2x2 Max Pooling [32] ---------- 29

Figure 26: Fully Connected Layers Architecture [33] ---------- 30

Figure 27: Accuracy Results of Testing the Pre-Trained Model ---------- 32

Figure 28: Fine-tuning Steps [37] ---------- *34*

Figure 29: Transfer Learning Accuracy Results ---------- *35*

Figure 30: Example of Class 1 Prediction ---------- *36*

Figure 31: Example of Class 0 Prediction ---------- *37*

Figure 32: Example of Class 2 Prediction ---------- *37*

Figure 33: Pre-trained Model Training vs. Testing Loss ---------- *38*

Figure 34: Pre-trained Model Training vs. Testing Accuracy ---------- *39*

Figure 35: Pre-Trained Model Training Accuracy Results ---------- *39*

Figure 36: Transfer Learning Model Training vs Testing Loss ---------- *41*

Figure 37: Transfer Learning Model Training vs Testing Accuracy ---------- *41*

Figure 38: Real-world Image of Class 0 Prediction ---------- *42*

Figure 39: Real-world Image of Class 1 Prediction ---------- *43*

Figure 40: Real-world Image of Class 2 Prediction ---------- *43*

Figure 41: Experiment 2 Pre-Trained Model Accuracy Result ---------- *44*

Figure 42: Experiment 2 Transfer Learning Accuracy Results ---------- *45*

Table of Tables

Table 1: Epoch Number and Loss Percentage Results ---------- *32*

Table 2: Pre-trained vs. Transfer Learning Loss Comparison ---------- *40*

Table 3: Experiment 1 vs Experiment 2 Comparison ---------- *46*

Table of Equations

Equation 1: Supervised Learning Algorithm Equation ---------- *26*

Equation 2: MSE Loss Function Equation---------- *31*

CHAPTER 1

INTRODUCTION

1.1 Biometric Authentication System Background

Biometric is capable of efficiently recognizing and authenticating people through the use of people's unique biological characteristics. A biometric authentication system collects the user's biometric features and compares with the stored template to confirm that this user possesses the access right [1]. There are two types of biometrics, which are physiological measurements and behavioral measurements. Physiological measurements involve fingerprints, vein patterns, irises, and facial shapes. These measurements are static and do not change over time. Also, the physiological measurements contain biological traits, including DNA, blood, urine, and saliva, which are mainly used by medical staff and police forensics. On the other hand, behavioral measurements contain voice recognition, keystroke dynamics, and signature dynamics. These measurements are considered dynamic and tend to change over time.

While the earliest accounts of biometrics can be dated to 500BC in the Babylonian empire, the primary record of a biometric authentication system occurred in Paris, France in the 1800s. Bertillon developed a technique of fixed body dimensions for the classification and comparison of prisoners. To use this method, he had to use distinctive biological features to authenticate identity. Furthermore, fingerprinting was allowed in the 1880s, it was not only used to identify criminals but also was used as a method of a signature on agreements. It was well-known that a fingerprint was representative of a person's identity, and one can be held accountable by using it [2]. After this era, in 1924, the biometric authentication system was rapidly implemented by law enforcement

and the FBI in the United States of America. Figure 1 lists all types of biometric authentication systems.

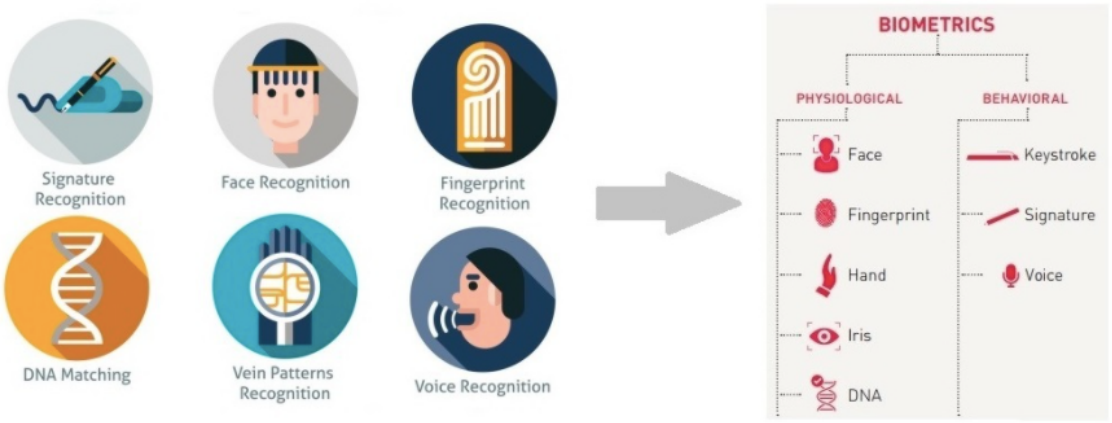


Figure 1: Biometric Types [2]

1.2 Advantages of Using Biometric Authentication System

Given the range of device logins that each enterprise asks their clients and customers to use their username and password combinations to gain access daily, the method of keying in a password repeatedly has grown to be difficult along with raising security risks. From hospitals and corporates to banks and school campuses, the use of the password is becoming a non-operational procedure of login credentials. Not only that passwords are difficult to remember, but hackers also are constantly inventing approaches to steal users' passwords to carry out their illegal actions [4]. Therefore, it is safe to conclude that using biometrics is more appropriate than passwords. Thus, the biometric authentication system is being used in the most sensitive places worldwide, such as airport security, building access, cars, hospitals, schools, and law enforcement.

1.2.1 Biometric Authentication Uses Unique Data

The level of complexity and specificity which can be found in biometrics cannot usually be found in password-based authentication. While human beings can create unpredictable

alphanumeric passwords that appear hard to decipher and tough to copy yet, the concept that passwords are theatrically formed and occur outside of the person can also lead to being easily copied or stolen. Unlike passwords, biometric data refers to a feature that is inherited in each individual. Hence, Biological characteristics became a crucial part of a person's arithmetical identity and are authenticated right where the individual is placed. Unlike passwords that can be forgotten by time, biometric data remain inherent in each individual over the years, as these are his/her physical and behavioral features.

1.2.2 Biometric Authentication Is Convenient to Use

The fact that people have to memorize personal identification numbers can be aggravating. Thus, most people use the same code for different platforms. In the USA, the average user links up to 130 online servers to a single e-mail account [4]. If a single password got stolen, there is a high risk of losing access to more than one account at once. Unlike password-based authentication, a biometric authentication system reduces the need for users to input an extraordinary password on different servers or even risk losing their information due to hacking. When using biometrics data, the user only needs to wait for the biometric scanner to finish scanning for accurate matches. The biometric authentication system permits easy confirmation, which makes it more convenient for users.

1.2.3 Biometric Authentication Supports Multi-Factor Authentication

For people who still appreciate keying a password or drawing lock patterns but want to experience a better layer of security, the use of a biometric authentication system allows them to do so by combining biometrics data with other different modes of verification. Most secured biometric systems can acclimate any combination of physical, behavioral keys, and insignificant codes to present action. For example, an individual can enter a password then continue with a

fingerprint scan to gain access to a requested resource. The use of multi-factor authentication, at least two or more forms of authentications, is mandatory as input before any user becomes accurately authenticated [5]. The more factors that are needed to classify a person commonly make it tougher to imitate them. This procedure ensures a combination of password-based authentication and biometric authentication to give the user a higher protective security system of their private information. Figure 2 shows different identifiers and how using at least two forms of authentication can be beneficial.

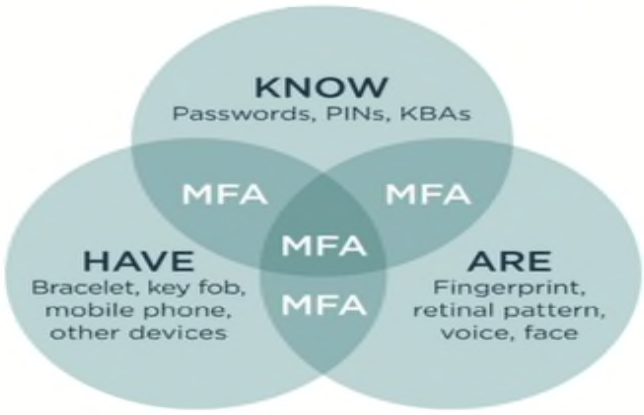


Figure 2: Multi-Factor Authentication [5]

1.3 Facial Recognition as Biometric Security

Face recognition is becoming more popular every day. Various technologies use facial recognition nowadays, such as using Face ID to unlock iPhones. When dealing with unlocking iPhone, a facial recognition system does not need a huge database of an individual's photos to define an individual's identity to be able to grant access to the iPhone. Instead, the system simply detects and identifies one individual as the unique owner of the device where it limits access to other people [7]. Other than unlocking phones, facial recognition is used for many purposes, such as law enforcement. NBC reported that using facial recognition is highly increasing among law

enforcement agencies within the US and other countries as well [8]. Law enforcement uses the technology to take photos of an arrestee to add these photos to databases to be scanned when another criminal search is being carried out by the police. Figure 3 shows how law enforcement uses facial recognition technology to find matches.

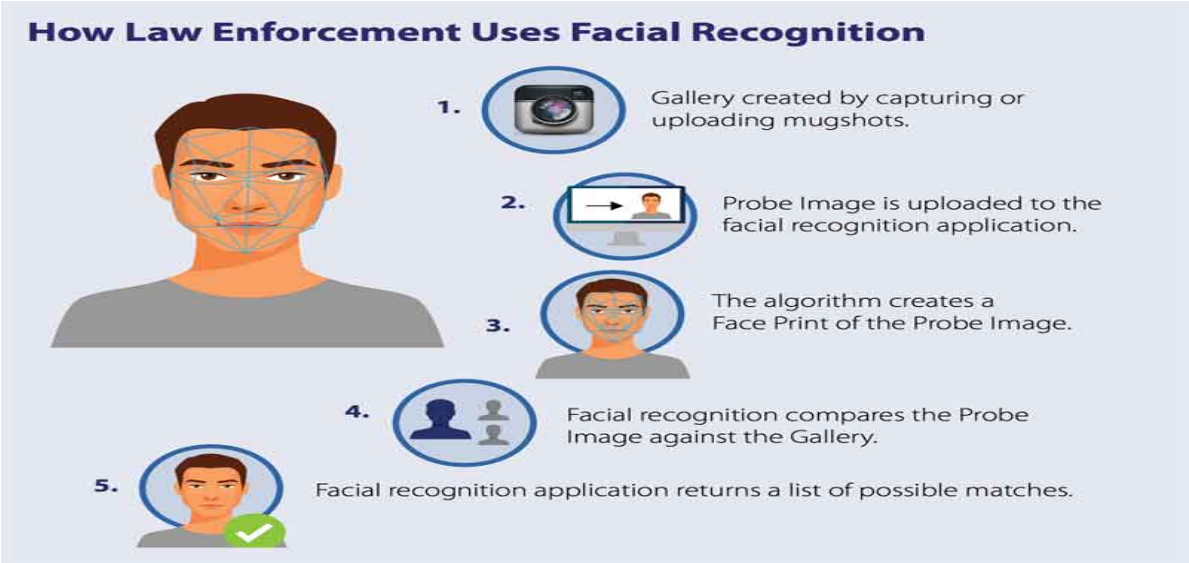


Figure 3: Law Enforcement using Face Recognition [8]

A facial recognition system is becoming a very popular method at many airports and border controls around the world. The US Department of Homeland Security reported that facial recognition procedures would be used on almost 97% of travelers by 2023 [10]. United States authorities and airlines are gradually using facial recognition systems at check-in process, security, baggage drop, and throughout the boarding process. Airlines are actively using the technology to scan travelers' faces to get each individual through the security and boarding processes much faster, which allows travelers who are holding biometric passports to skip long lines by walking through the automated ePassport control to be able to reach their gates quickly. The use of facial recognition not only reduced the waiting times at the airport but also enhanced security. Figure 4

explains how airports in the USA use facial recognition to increase their security system and prevent any possible threats.

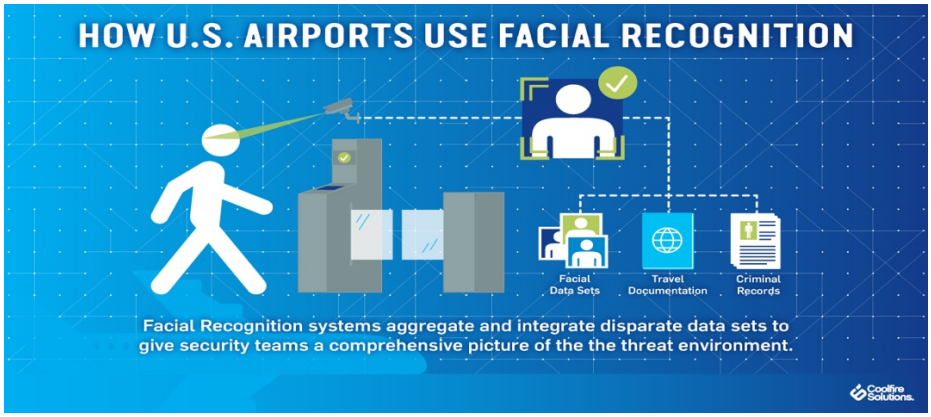


Figure 4: Airports Using Facial Recognition [10]

Another important place that uses facial recognition is online banking. Users do not need to use one-time passwords but, they can approve transactions by looking at their devices instead [12]. Facial recognition security systems helped in decreasing hacking. Even if hackers try to steal an individual's phone database, there is a technique, called liveness detection. Liveness detection is used to determine if the person who is making the transaction is a live human being or a fake representation. This technique prevents many hackers from using unreal representation to proceed with fake transactions.

CHAPTER 2

RELATED WORKS

Preeti Nagrath, Rachna Jain, Agam Madan, and their team in SSDMNV2 proposed an approach using deep learning, Keras, TensorFlow, and OpenCV to perform real-time face masks detection. This team was able to find an open-source dataset, which is called Kaggle's Medical Mask Dataset. The dataset was created by Mikolaj Witkowski and Prajna Bhandary. In the proposed SSDMNV2 model, the team divided the image dataset into two different categories, where the first category included people having masks when the other category included people not having masks on. They were able to classify their images using the MobilenetV2 image classifier. The MobilenetV2 was chosen for easier deployment in real-time even on embedded devices. After many trials, the team achieved 93% accuracy results and 93% as F1 score. Overall, the SSDMV2 model is very useful and can be used during pandemic times by the authorities to deploy in real-time devices [13].

Mohamed Loey, Mohamed Hamed N. Taha, Gunasekaran Manogaran, and Nour Eldeen M. Khalifa created an approach to annotate and localize the medical face mask objects in real-life images. Their proposed model included two components: a feature extraction process using the ResNet-50 deep transfer learning model, and a medical face mask detection using YOLO v2. The authors were able to improve the detection performance by using the mean IoU. This helped in estimating the best number of anchor boxes. With that said, the authors were able to conclude that using Adam optimizer achieved the highest accuracy of 81% [14].

Ge S., Li J., Ye Q., Luo Z built a model by using a dataset to find the unmasked and masked face. The dataset, which is called Masked Faces (MAFA), included 35,806 images of people wearing masks. The authors used a convolutional neural network to propose their model including

three different modules, which are proposal, implementation, and authentication. Their work achieved 76.1% accuracy results [15].

Ejaz Md. S., Islam Md. R., Sifatullah M., Sarker A applied machine learning techniques to distinguish between people wearing face masks versus people not wearing face masks. They used Principal Component Analysis (PCA). The paper accomplished identifying people who are not wearing masks provides a better recognition rate in the Principal Component Analysis. Authors were able to find that extracting features from people wearing face masks is lesser than people who are not wearing face masks. They also found that accuracy has much decreased after classifying people wearing a face mask, which gave an accuracy of 70% [16].

Sammy V. Militante and Nanette V. Dionisio have proposed a study on real-time facemask recognition using deep learning methods and Convolutional Neural Networks (CNN). The authors' study presents defined and rapid results for facemask detection. The study was about distinguishing between people wearing a facemask and people who are not wearing a facemask. Authors were able to use the CNN model to train their model, which helped them to achieve 96% result as performance accuracy. Furthermore, the study was a valuable tool in fighting the spread of the COVID-19 virus by identifying an individual who wears a facemask or not wearing a facemask. The study can help in setting alarms to warn other people if there is an individual who is not wearing a face mask [17].

CHAPTER 3

Facial Recognition System

3.1 Facial Recognition Structure

A face recognition procedure has been developed to recognize if there is any person inside a photo. It also locates an individual's face and identifies who an individual is. The procedure is separated into four stages, which are face detection, face alignment, feature extraction, and face recognition. Figure 5 represents the facial recognition system structure.

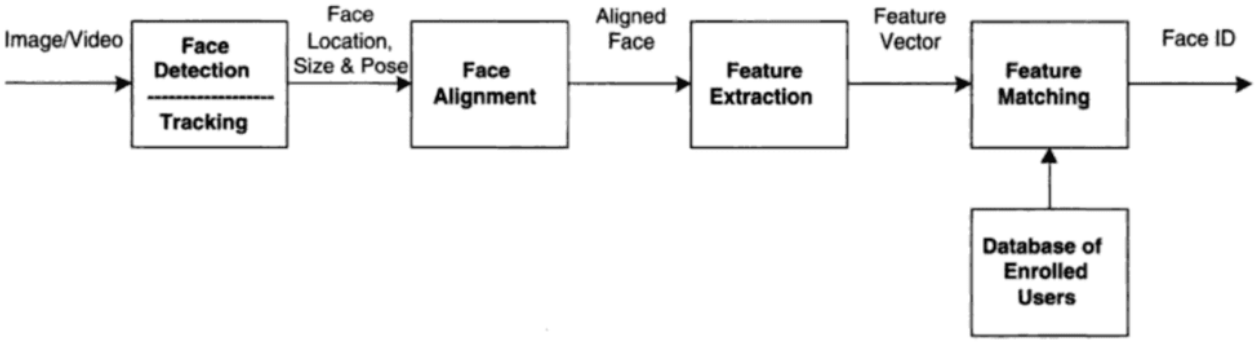


Figure 5: Facial Recognition System Structure [17]

3.1.1 Face Detection

In this very first step, the main purpose is to identify the part of the photo or a given video that represents a face as well as recognizes the location of these faces. Then, the output of this step can transform the given data into patches that contain each face as the input image.

3.1.2 Face Alignment

In this second step, the system normalizes the face to become more consistent with the database by identifying geometry and photometric. In other words, the system captures and analyzes the image of a given face. Usually, most facial recognition systems capture images in 2-D rather than 3-D to make images more suitably matching with public photos and databases. As

the system reads the image, it is looking for key factors, such as the depth of an individual's eye sockets, the shape of an individual's cheekbones, the distance between the eyes, the distance from an individual's forehead to chin, and the outline of an individual's lips, chin, and ears. The face alignment is used to validate the scales, resolution, brightness, levels of zooms, and orientations of the patches performed in the previous step, which serves as preprocessing for face recognition.

3.1.3 Feature Extraction

After normalizing the face, an individual's face patches are extracted from images. The system is responsible for converting the image to data based on an individual's facial features. The system tends to extract the most significant data from these images, where the system should identify the most relevant bits of data, all while ignoring any noises [19]. Feature extractions of facial recognition can achieve information packing, noise cleaning, dimension reduction, and salience extraction. During feature extraction, a unique faceprint is computed for each person. Figure 6 shows an example of the most meaningful features from an extracted image.

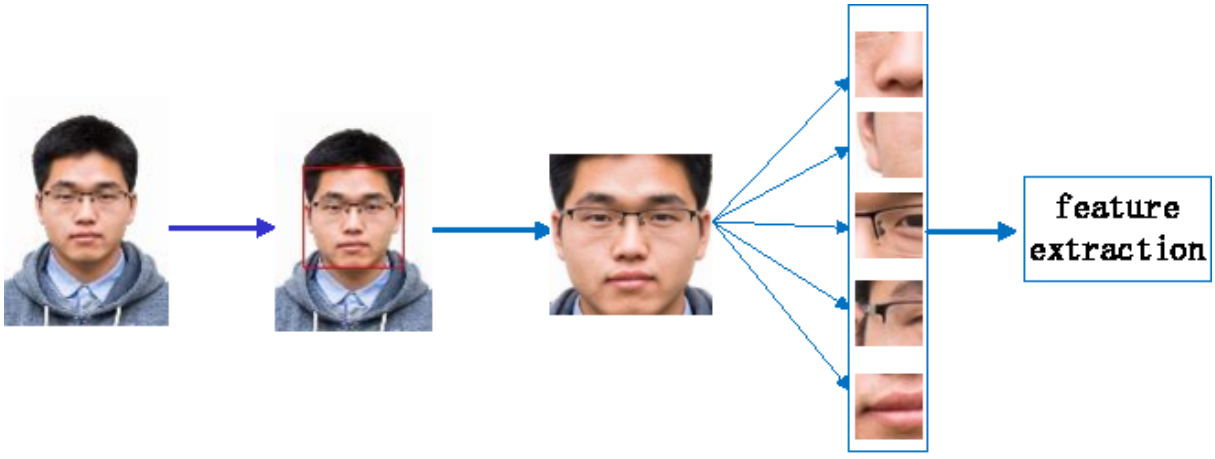


Figure 6: Feature Extraction [19]

3.1.4 Face Recognition

In the very last step, the system is responsible for distinguishing between the identities of different people's faces. For the system to achieve an automatic recognition process, a face database can be built by taking several images of each individual's face, then the features of these images are extracted and stored in the system database. Then, whenever an input image appears, the system performs face detection as well as feature extraction of the input image. Later, the system compares the image features to each stored faceprint in the database to either grant or deny access.

In facial recognition, there are two different applications, which are called identification and verification. For face identification, the system is asked to tell whom the given image is for. On the other hand, for face verification, the system is asked to tell if an identified image, is true or false [21]. Figure 7 indicates the difference between how the system responds to identification versus verification.

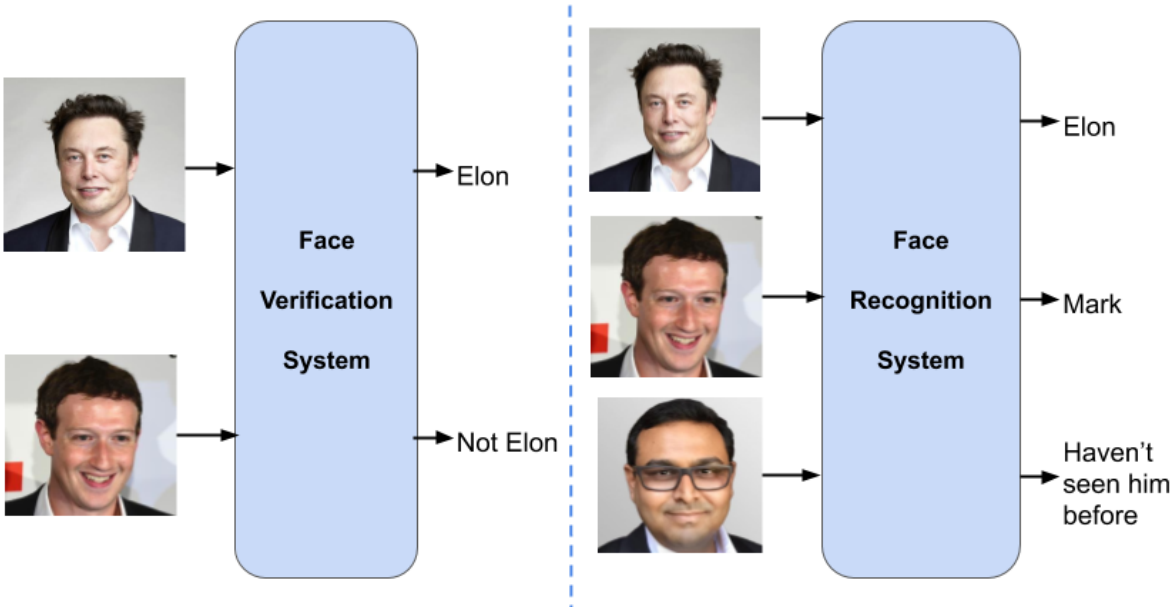


Figure 7: Identification and Verification in Face Recognition [21]

3.2 Facial Recognition Accuracy

As mentioned in the previous section, facial recognition systems tend to use particular computer algorithms to pick up precise and unique details about an individual's face, which helps in distinguishing one face from another. However, various facial recognition systems compute the possibility of matching unidentified individuals with a specific faceprint that was previously stored in the system's database. The purpose of these systems is to return numerous possible matches and rank them in order based on probability of accurate identification instead of returning one match. However, facial recognition systems sometimes face a challenge when identifying people under certain conditions, such as low-quality photos, poor resolution, poor illumination, and suboptimal view angle. All the mentioned conditions can result in producing errors, and when it comes to facial recognition, there are two types of possible errors as false positive and false negative.

A false positive can occur when the system mistakenly identifies the match between two images, but there is no match between them in reality. On the other hand, a false negative occurs when the system fails to identify the true match between two images. With that said, understanding the concept of False Acceptance Rate and False Rejection Rate is very important because these two work as a trade-off. False Rejection Rate and False Acceptance Rate are the core two elements of the facial recognition biometric authentication system threshold. They work hand in hand, which means reducing the threshold level to allow more leniency in the system for accepting discrepancy as well as noise, which will lead to a significant increase in the False Acceptance Rate. Likewise, if it is decided to increase the threshold to create a more protected and secured system, then there is a big chance for the False Rejection Rate to rise as well [23]. Thus, these measurements should be something to bear in mind when a designer decides to create a facial recognition biometric system. The system should be able to provide high-security level as well as user-friendly access

control. For example, if someone is trying to unlock their phones using facial recognition, it is often better if the system calculates False Negative errors than False Positive errors. Failing to identify the right person is better than granting access to ineligible users. Allowing misidentified users to unlock the phone will grant them access to all private data. Figure 8 presents a graph of how the level of security is impacted by the percentage of False Rejection and False Acceptance Rates.

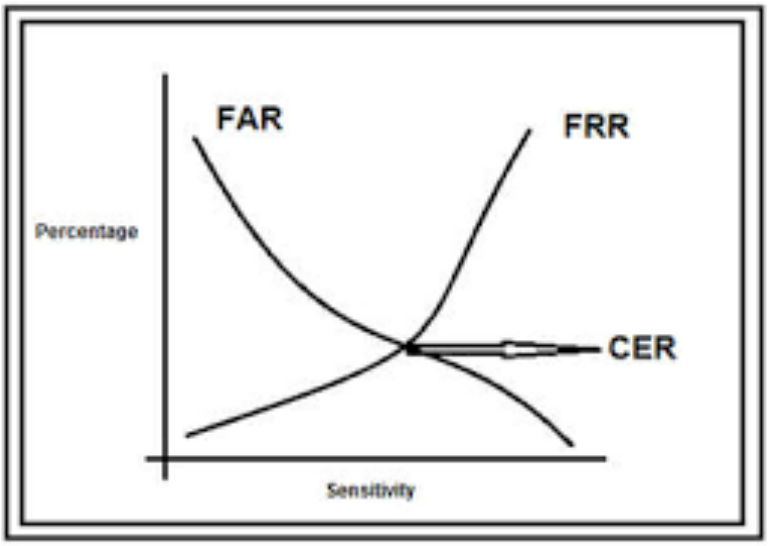


Figure 8: Percentage of FRR and FAR [23]

CHAPTER 4

FACIAL RECOGNITION THREATS AND COUNTERMEASURES

4.1 Threats

The precision and flexibility of facial recognition security systems have been able to secure everything from smartphones to airports. However, nowadays, people have found methods to trick the systems using 3-D printing face masks, makeup techniques, and by wearing face masks.

4.1.1 3-D Printing Face Mask

Nowadays, researchers have found a way to manipulate face recognition technology by using 3-D printing face technology. A pigmented hard resin is created with face features, such as texture, hair, and skin tone [25]. Criminals and people who are hiding from surveillance can use this tool to compromise the facial recognition system by providing a fake face. Figure 9 shows an example of someone who is wearing the 3-D printing face mask.

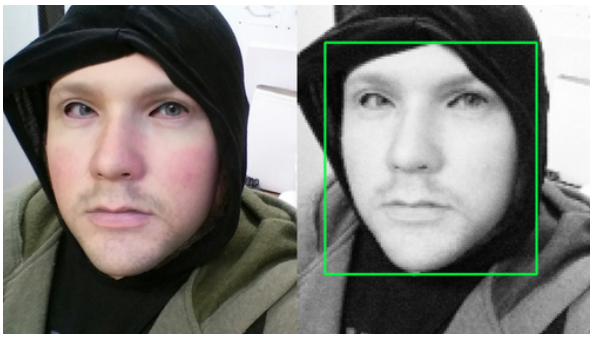


Figure 9: 3-D Printing Face Mask [24]

4.1.2 Makeup Techniques

Facial recognition systems can also be compromised by people who are applying maximalist makeup techniques. If someone covers his/her face with senseless makeup, it will be very challenging for the system to recognize the true identify. The key features of the human faces, including the shape of their forehead, the bridge of their nose, and the depth of their eyes are hidden

[26]. Likewise, people who wear jewelry, as well as other imposing objects on their faces, would make another challenge. In figure 10, an example of makeup techniques is shown.

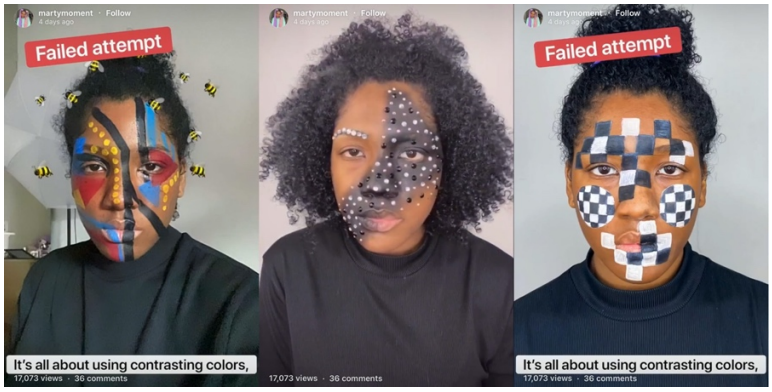


Figure 10: An example of Makeup Techniques to Be Used as a Threat for Facial Recognition [25]

4.1.3 Face Masks

Since the spread of COVID-19, people started to cover their faces to reduce the spread of the virus, which caused a challenge to facial recognition systems to recognize and classify their faces. The NIST has found an error rate of up to 50% of matching between face masked images and unmasked images for the same person, even with using the best facial recognition algorithms. [27]. When people are wearing face masks that sufficiently cover the mouth and nose causes an algorithm rate error of facial recognition. In the NIST study, it was found that wearing black masks causes errors more than wearing blue masks. Also, the more of the nose is covered by the mask, the harder the facial recognition systems to identify the face. While facial recognition algorithms work by computing the distances between an individual's facial features, wearing face masks lowers the accuracy of the system's algorithms because most of the key identification features are being removed or hidden by the mask.

4.2 Countermeasures

In the previous section, three threat types were introduced that can be challengeable to facial recognition systems. However, the companies that create facial recognition systems are rapidly adapting to the new world and its challenges. Scientists are already boosting their systems to recognize individuals with half-covered faces where the identification accuracy reached almost 90%. The new adaptive systems work by focusing on the uncovered areas of the face, such as eyes, eyebrows, hairline, and forehead. Another way to overcome wearing a face mask challenge, many companies are planning to provide face masks with customers' faces printed on them. These face masks can help people to unlock their smartphones without having to take their face masks off. These companies are not only working on helping people to unlock their smartphones but also, they are working on calculating the ability of Artifactual Intelligence systems to collect and match images of people wearing different types of face masks in international airports from around the whole country [28]. The US Department of Homeland Security (DHS) was able to identify people who are wearing masks with an identification accuracy of 77% [29]. Even if this is not a 100% accuracy result, yet this result is a promising finding. These results may decrease risks for many travelers as well as provide them the ability to travel without having to remove their masks at airports. Thus, to overcome the rising challenges of facial recognition security system, a CNN model which can classify between individuals who are wearing a face mask, not wearing a face mask, and those who are wearing face mask incorrectly was proposed in this paper.

CHAPTER 5

DATA SOURCE

5.1 Dataset Description

Machine learning models depend mainly on data. Without applying high-quality training data even, the most well-functioning computer algorithms can be impractical. Hence, no other element is more fundamental in machine learning techniques other than quality training data. Training data indicates the original data that is used to improve the model. The model uses the training data to build and improve itself. The quality of this data has deep effects on the model's succeeding development, which helps in setting a powerful model for any future applications that may use the same training data. Training data in machine learning involves a human contribution to examine and develop the data for machine learning procedures.

In this thesis, supervised learning was used to train the model with labelled training data. When the data is labeled, the dataset usually is marked with fundamental identification features. These features are very beneficial for the model to train and learn. Therefore, the accuracy of the model and its ability to identify the outcomes and perform high-quality predictions are largely affected by the extracted features from the dataset as well as the quality of labeling.

Therefore, the main purpose of this thesis is to train a model to classify individuals' faces with face masks. This system could be used to encourage people to wear face masks. In addition, it can perform facial recognition at various places, such as airports and ATM. To achieve the best accuracy results, one large Face mask detection dataset was first used to construct the model. Then this model is adapted to fit our actual smaller dataset using transfer learning.

5.1.1 Face Mask Detection Dataset

This dataset, which is available on Kaggle, has two classes which include people who wear masks and people who do not wear masks. The dataset consists of 3725 images of faces with masks, and 3828 images of faces without masks, which gives us a total of 7553 face images. All images in this dataset have three color channels (RGB) [30]. Figure 11 shows sample images of people who wear face masks. Figure 12 shows sample images of people who do not wear face masks. In this dataset, people who wear face masks were labeled with 0, while people who do not wear face masks were labeled with 1. This dataset was used to pre-train the model initially. Figure 13 displays a distribution of the total number of different classes of the dataset. Likewise, figures 14 and 15 present the distribution of each class' image sizes.

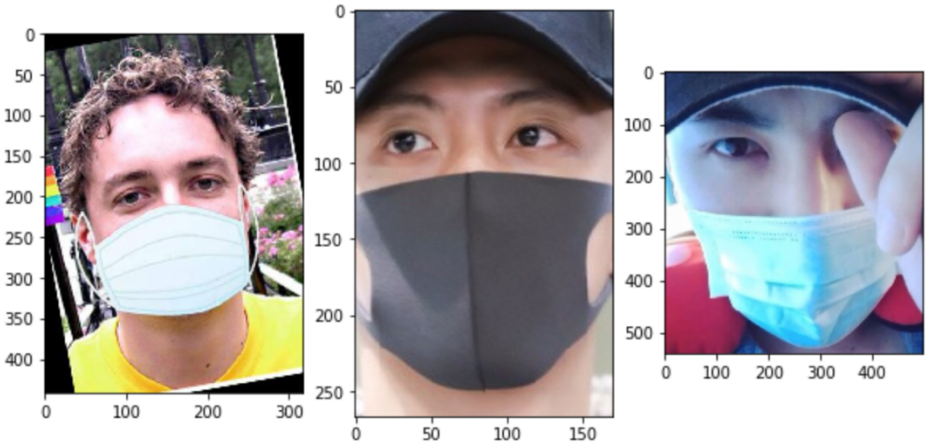


Figure 11: Sample Images of People Who Wear Face Masks (Face Mask Detection Dataset)

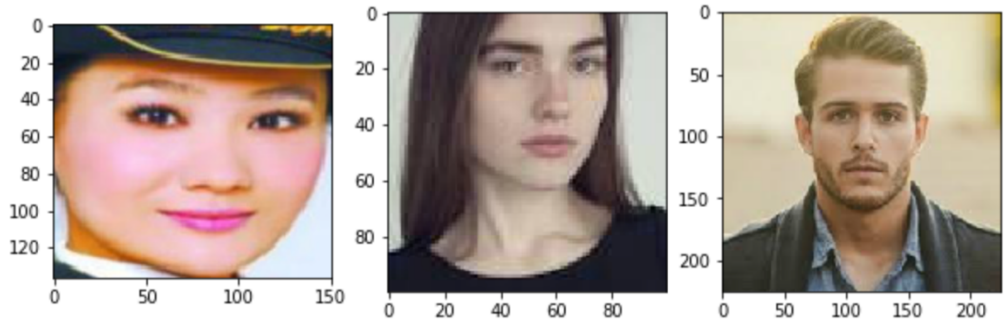


Figure 12: Sample Images of People Who Do Not Wear Face Masks (Face Mask Detection Dataset)

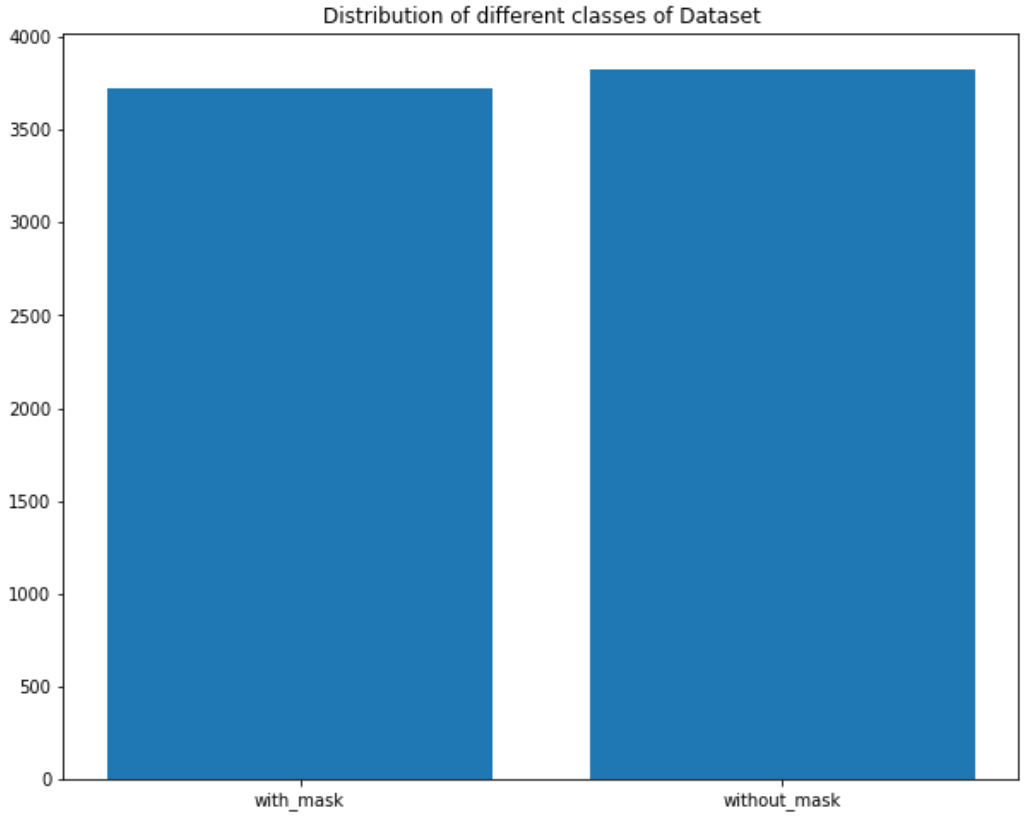


Figure 13: Classes Distribution of Face Mask Detection Dataset

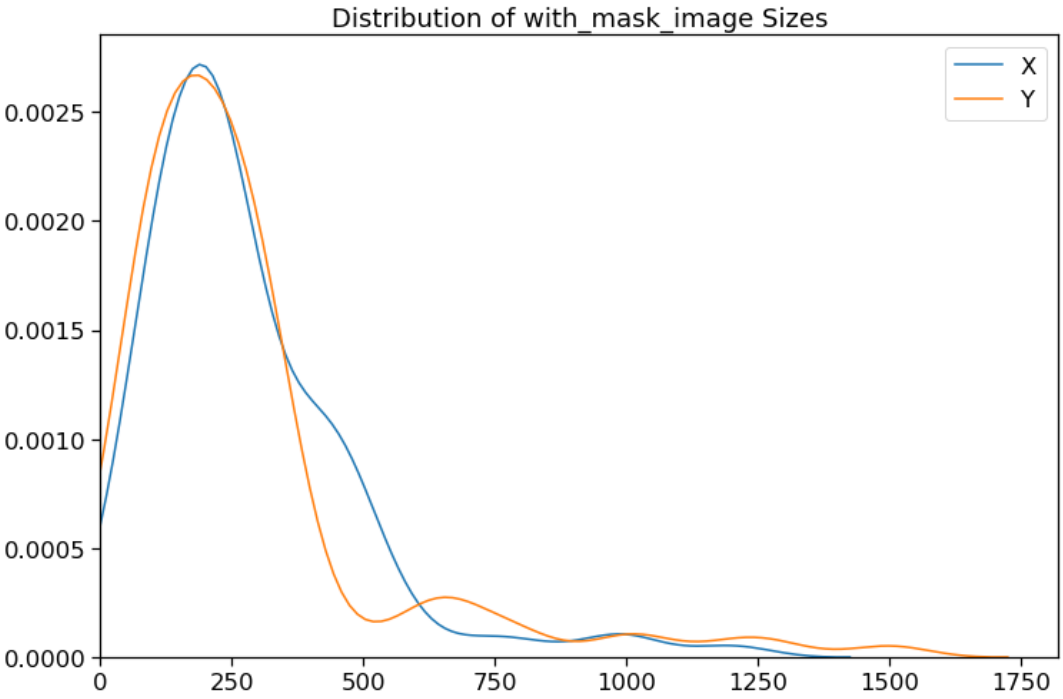


Figure 14: Distribution of with_mask Class Image Sizes

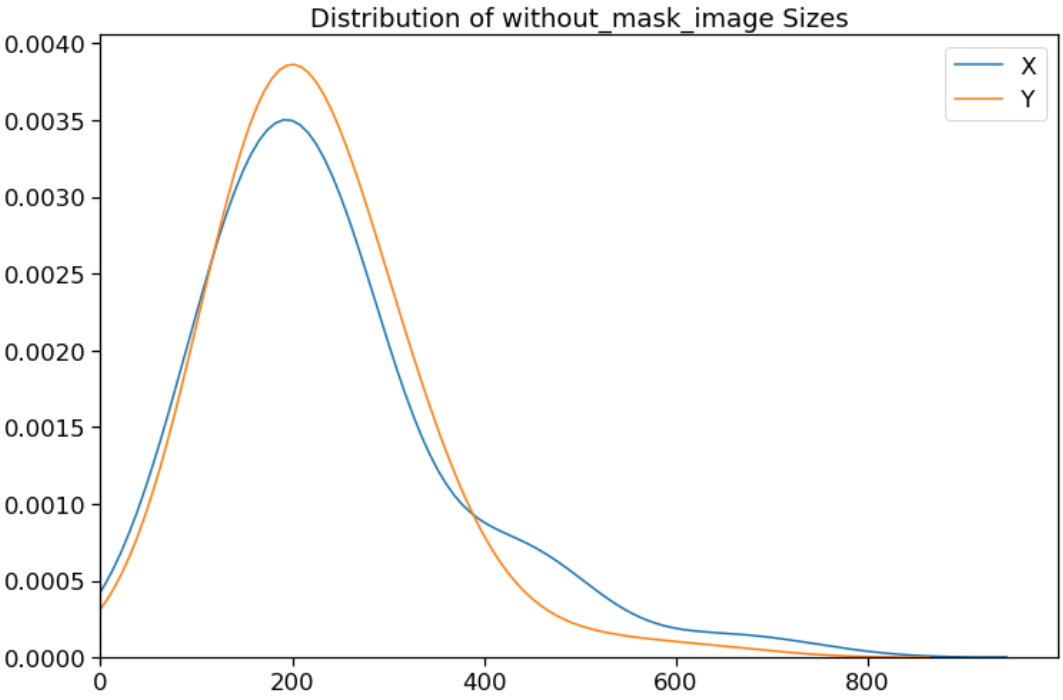


Figure 15: Distribution of without_mask Class Image Sizes

5.1.2 Face Mask Detector Dataset

This dataset, which is available on Kaggle, has three different classes which include people who wear masks, people who do not wear masks, and people who wear masks but in an incorrect manner. The dataset consists of 703 images of people who inappropriately wear a mask, 690 images of people wear masks, and 686 images of people who do not wear a mask at all. The total number of images for this dataset comes up to 2079 face images [31]. Figure 16 shows sample images of people who wear face masks. Figure 17 shows sample images of people who do not wear face masks. Figure 18 shows sample images of people who wear face masks but in an incorrect manner. In this dataset, people who wear face masks were labeled with class 0, while people who do not wear face masks were labeled with class 1 and people who wear face masks incorrectly were labeled with class 2. In this paper, this dataset was used for applying the transfer learning method. Figure 19 displays a distribution of the total number of different classes of the dataset. Likewise, figures 20, 21, and 22 present the distribution of each class' image sizes.

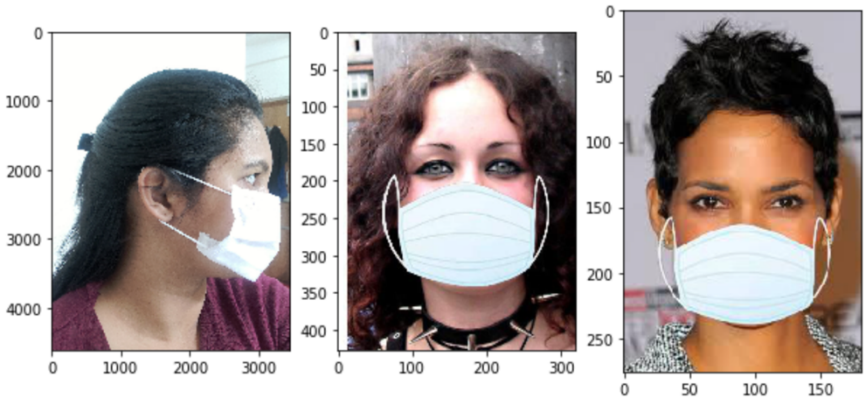


Figure 16: Sample Images of People Who Wear Face Masks (Face Mask Detector)

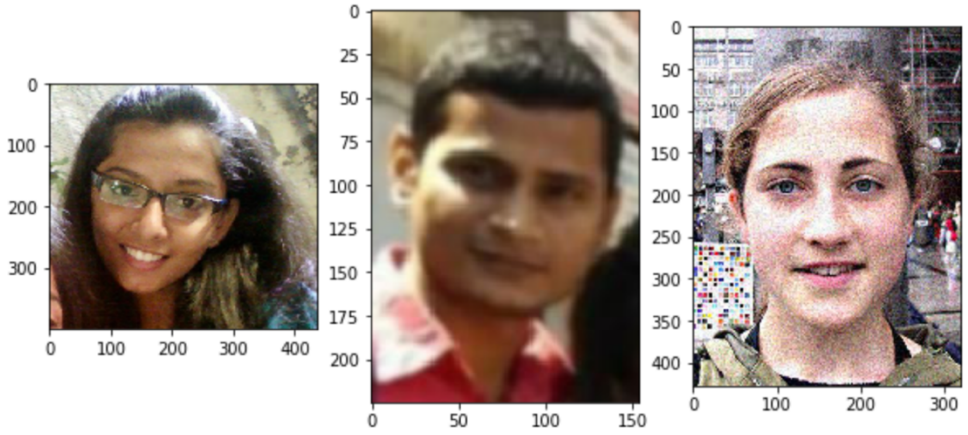


Figure 17: Sample Images of People Who Do Not Wear Face Masks (Face Mask Detector)

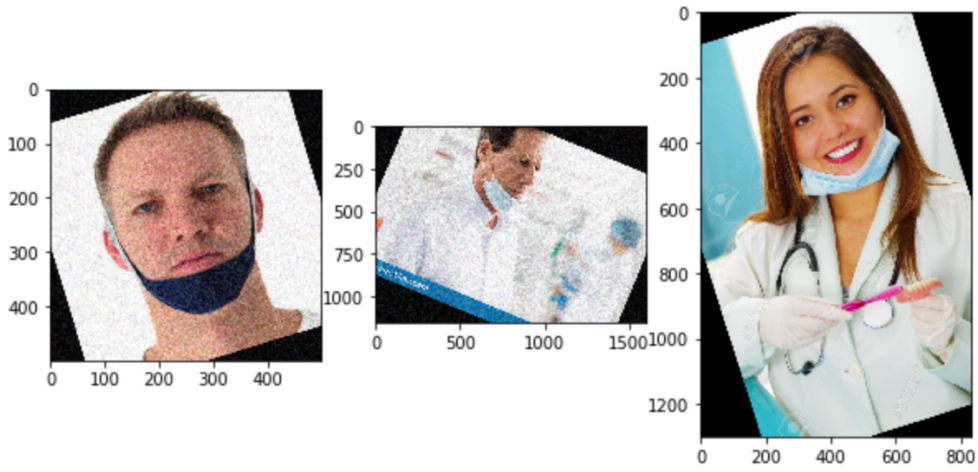


Figure 18: Sample Images of People Who Wear Face Masks but Incorrectly (Face Mask Detector)

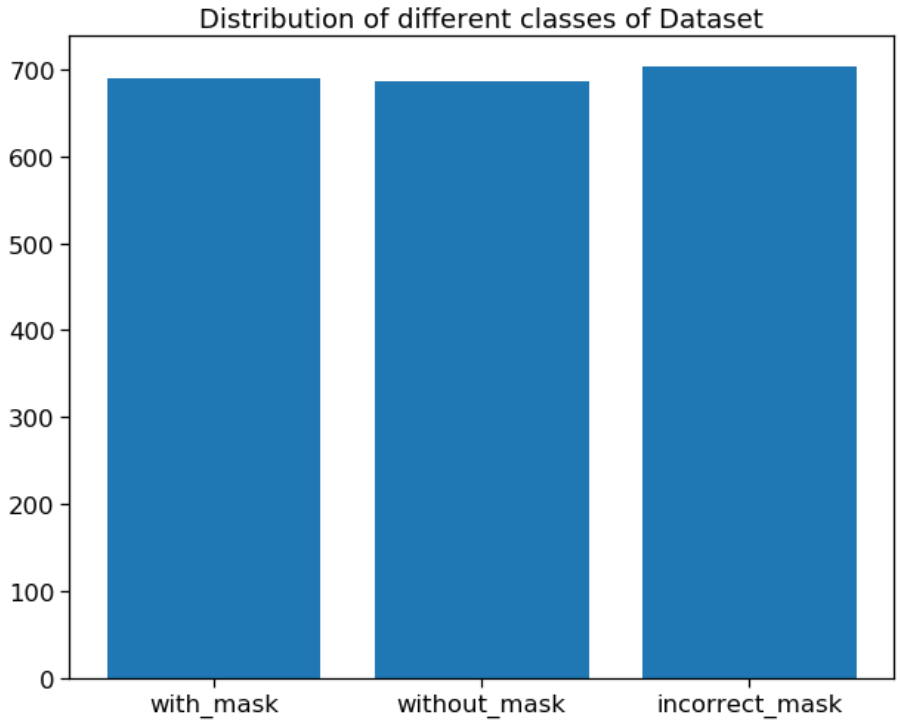


Figure 19: Classes Distribution of Face Mask Detector Dataset

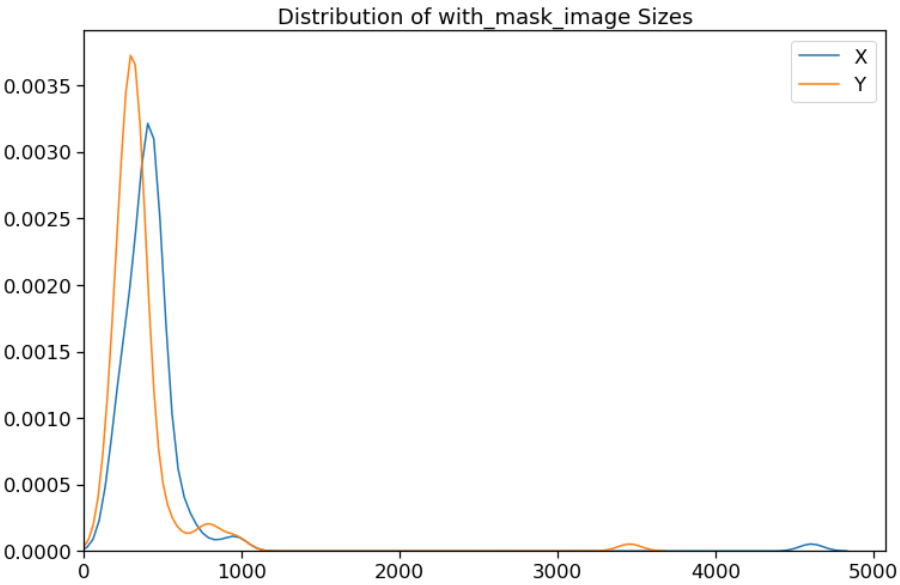


Figure 20: Distribution of with_mask Class Image Sizes

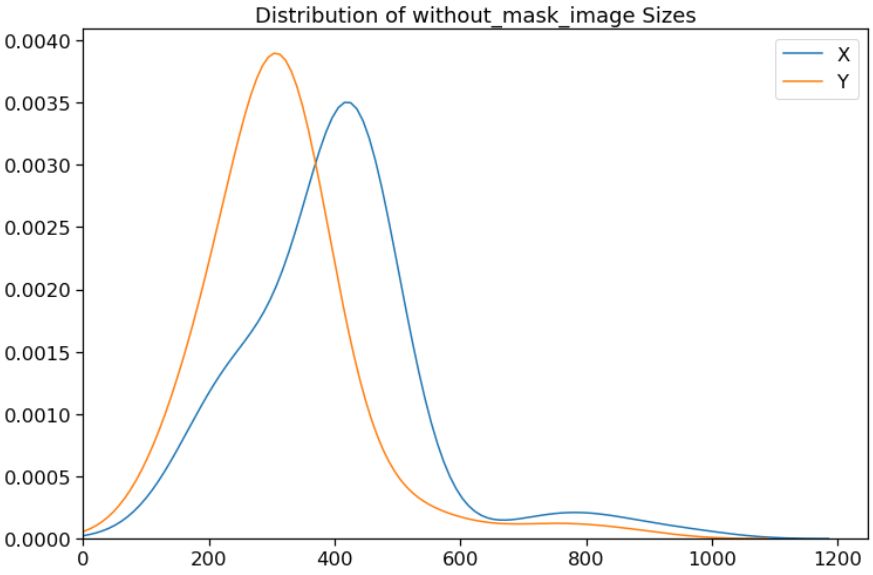


Figure 21: Distribution of without_mask Class Image Sizes

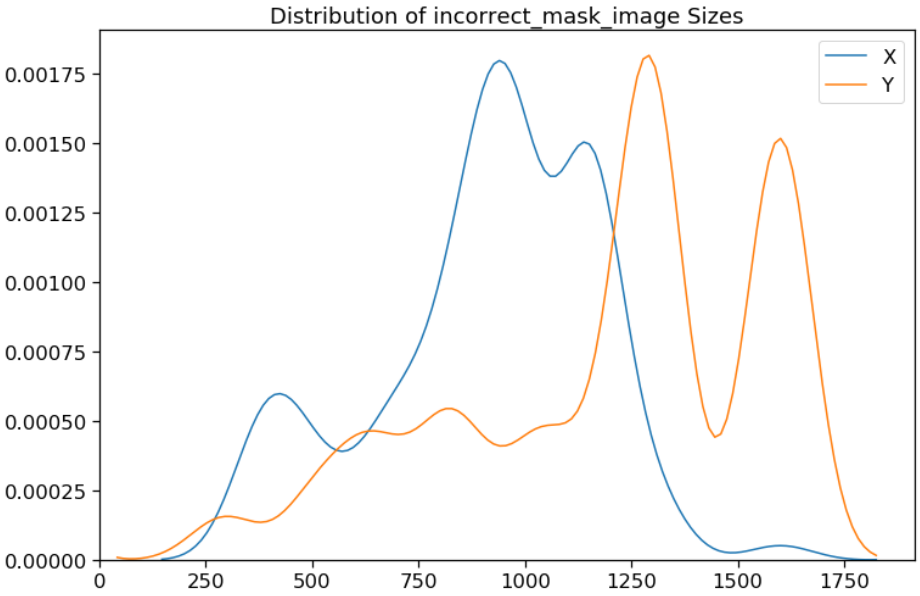


Figure 22: Distribution of incorrect_mask Class Image Sizes

5.2 Data pre-processing

In machine learning, performing data pre-processing is a very significant step that helps in improving the quality of data to help the extraction of important understandings from the data. In data pre-processing, the data is getting cleaned and organized to become proper for building and

training a model. For both datasets, images were converted to grayscale instead of color channels. The main purpose of using grayscale is to simplify the algorithm as well as reducing computational requirements to help in the extracting descriptors process. Also, for both datasets, images were resized to 100x100, resizing images is a very crucial step in the preprocessing step because machine learning models tend to train faster on small-sized images. Also, resized images are easier for the model to deal with since they are in the same dimensions. Likewise, in both datasets, the data was split into two datasets, which are training and testing. The whole dataset was split into a 90:10 ratio, which means 90 percent data take in train and 10 percent data take in the test. This led the testing dataset to become 755 face images and the training dataset to become 6798 face images for Face Mask Detection Dataset. Likewise, the testing dataset became 207 face images, and the training dataset became 1872 face images for Face Mask Detector Dataset. Lastly, the dataset was shuffled to avoid any chance of overfitting and improving the machine learning model quality and predictive performance. Figure 23 shows a code example of applying a 90:10 ratio to split the Face Mask Detection Dataset.

```
import torch
X= torch.Tensor([i[0] for i in training_data_plot]).view(-1,100,100)

X= X/255.0
y= torch.Tensor([i[1] for i in training_data_plot])

VAL_PCT= 0.1
val_size= int(len(X)*VAL_PCT)


train_X=X[:-val_size]
train_y=y[:-val_size]

test_X=X[-val_size:]
test_y=y[-val_size:]

print(len(train_X))
print(len(test_X))
```

```
6798
755
```

Figure 23: Example of Splitting Data by 90:10 Ratio

CHAPTER 6

METHODOLOGY AND EXPERIMENTAL SETUP

6.1 Supervised Learning

Since supervised learning is the best and most common technique of machine learning nowadays, in this paper, supervised learning techniques were used to achieve the best performance results. Supervised machine learning algorithms learn by example. The term supervised learning comes from the concept of training the dataset. The training dataset always consists of input images, which are also always combined with their proper outputs. During the training process, any patterns in the data which relate to the selected outputs will be examined by using the supervised learning algorithm. After training, a supervised learning algorithm will take in new unidentified inputs and will be able to identify the new inputs with their correct labels based on the preceding training data. The main purpose of a supervised learning model is to presume the correct label for newly presented input data. A supervised learning algorithm has a simple equation that can be seen in Equation 1. In this equation, Y is the predicted output and x is the input value. This function is used mainly to connect input features to their predicted output which is created by the model during the training process.

$$Y = f(x)$$

Equation 1: Supervised Learning Algorithm Equation

6.2 Convolutional Neural Network

Learning face features by using the Convolutional Neural Network (CNN) was the main applied technique in this study. To be able to learn face features from images, deep neural networks have to be applied. Therefore, a Convolutional Neural Network is a Deep Learning Algorithm, which concentrates on processing data, including images. Since images are represented as

graphical data, then each image includes a chain of pixels where each pixel has a value to help in determining the color and brightness level of each pixel. CNN is also combined with multiple layers of artificial neurons, which are mathematical functions that can compute the weighted sum of various inputs and outputs. In simple words, CNN is used to extract image features and convert them to a smaller size without losing their characteristics. CNN architectures assume that the inputs are images, which allow encoding definite properties into the model. These then make the forward functions more efficient to implement and reduce the number of parameters in the network [32]. A convolutional neural network is a series of steps that need to be taken to properly get the model to work. Every step of the CNN is demonstrated in Figure 24. CNN requires less preprocessing compared to other classification algorithms.

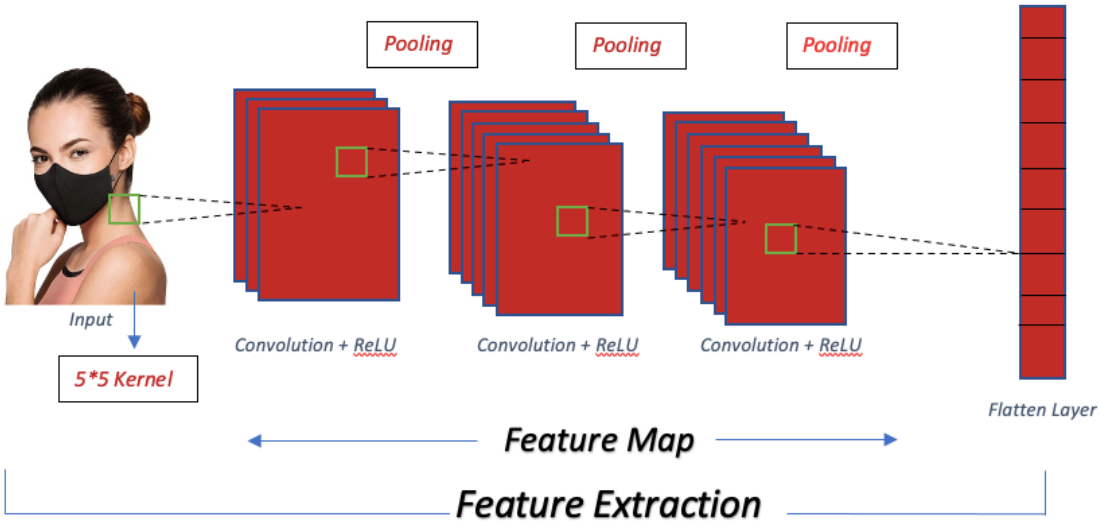


Figure 24: Convolutional Neural Network Architecture

6.2.1 Convolutional Layer

The convolutional layer performs as the main building block of the CNN process, which does most of the computing operations. In this layer, the CNN usually detects basic features, such as the shape of edges from the input image. In this paper, three convolutional layers were applied

to achieve the best performance of the model. Conservatively, the first layer is responsible for capturing the low-level features, such as color, gradient coordination, edges, and angles. When the architecture goes through another convolutional layer, the output of the first layer becomes the input of the second layer. With more layers being added, the architecture adapts to high-level features, such as a combination of curves and straight edges.

Each convolutional layer had two other parameters, which are kernel size and stride. In this experiment, the kernel size was set to 5x5, which helps in detecting differently sized features in the input image which will lead to different sized feature maps. Likewise, the amount of movement between applications of the filter to the input image is referred to as the stride and is almost always balanced in height and width sizes. Thus, in this experiment, the stride dimensions were set to 1.

After every convolutional step, ReLU has been used. ReLU stands for Rectified Linear Unit, which is a non-linear operation. ReLU is an element-by-element operation that has to be applied to each pixel. In the feature map, ReLU restores all negative pixel values by zero. The purpose of ReLU is to present non-linearity in the CNN model since most of the real-world data would be non-linear. As the model is building, applying all these operations, and adding more convolutional layers, the architecture goes through activation maps that produce more convoluted features to fully understand the features of human faces.

6.2.2 Max Pooling

In this thesis, to build the convolutional neural network, 2x2 max-pooling was used to decrease the three-dimensional size of the convoluted feature. 2x2 max-pooling is beneficial for decreasing the computing power necessitated to process the data through dimensionality reduction. Furthermore, it is useful for extracting major features, therefore maintaining the effective process of training the model. Max pooling also returns the highest value from the image's part that was

covered by the kernel. Using max pooling can also eliminate noisy activations and reduces dimensionality. The convolutional layer and the pooling layer together help the model to understand image features. Figure 25 shows an example of how 2x2 max-pooling is performed.

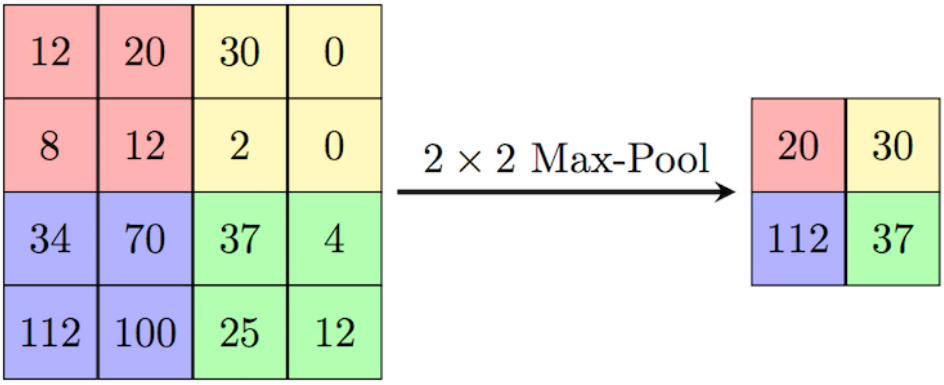


Figure 25: 2x2 Max Pooling [32]

6.2.3 Fully Connected Layer

In this methodology, another two fully connected layers were added, which helped in learning a non-linear mixture of the high-level face features which were obtainable during the process of the kernel. The SoftMax activation function was used after the fully connected layers. SoftMax function works by transferring a vector of numbers into a vector of probabilities. SoftMax results in normalizing the outcomes as well as converting those outcomes into probabilities that must add up to 1.0. Since input images were converted into a multi-layer neural network, flattening an image should then occur. The flattened output is then fed to a feed-forward neural network and backpropagation is applied to every step of the training process. The main purpose of using a fully connected layer is to use high-level features for classifying the input image into either people wearing a mask or people not wearing a mask. Figure 26 presents the whole architecture of adding fully connected layers.

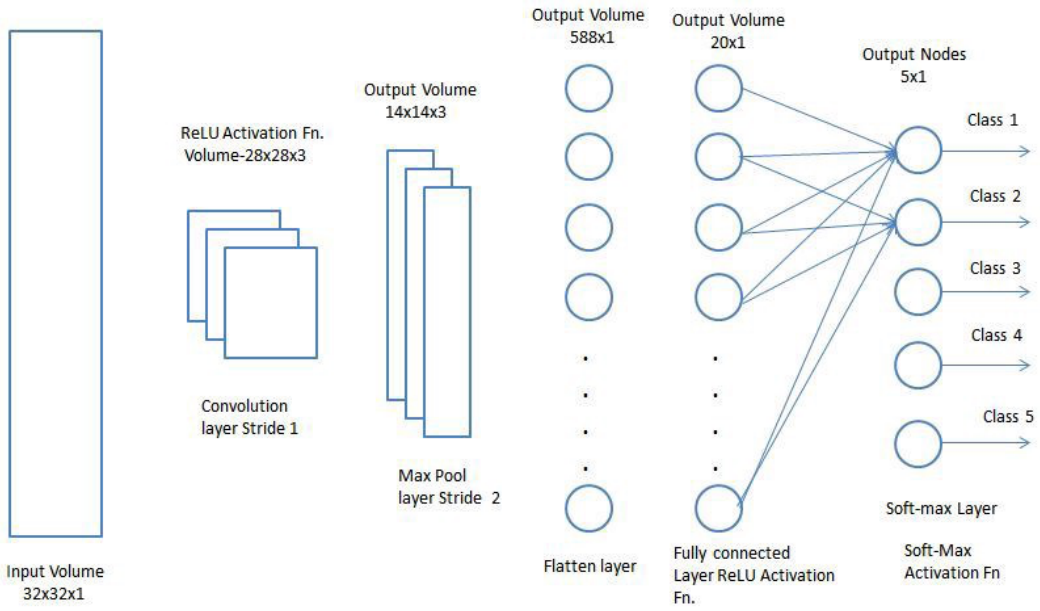


Figure 26: Fully Connected Layers Architecture [33]

6.3 Training

The training process is also known as backpropagation. The training step is very important for the model to find the weights that best accurately represent the input data to match its correct output class. Thus, these weights are constantly updated and moved towards their optimal output class. In this study, a Face Mask Detection Dataset was used to train the CNN model. During the training process, training data was split into smaller sizes of batches of 100. Batch size includes splitting the whole dataset into a chain of the reduced amounts of data fed into the model one at a time. Splitting the training dataset into batches helps in training the model faster and controlling the gradient error accuracy. Likewise, the learning rate of 0.001 has been applied to set the size of a step in the direction of the minimizing the loss function. The optimization algorithm, forward pass, loss function, backward pass, and weights updates are followed to train the model from the labelled data.

6.3.1 Adam Optimizer

During the training process of this paper, Adam optimizer has been applied to the CNN model. Adam is an adaptive learning rate optimization algorithm that has been proposed especially for training deep neural networks. Adam optimizer works by calculating each learning rate according to different parameters within the model [35]. Thus, using Adam optimization algorithms was very beneficial for this methodology because it uses valuations of the first and second gradient moments to acclimate the learning rate for each of the neural network weights.

6.3.2 MSELoss

In this approach to achieve the best accuracy results, the MSE loss function was used. MSE stands for Mean Square Error, which is commonly used and is the sum of squared distances between target variable and predicted values [36]. Equation 2 shows how the MSE loss function is being calculated, where n is the number of data points, y_{true} is the actual value for data point i and $y_{predicted}$ is the value returned by the model.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Ytrue - Ypredicted)^2$$

Equation 2: MSE Loss Function Equation

6.3.3 Running Epochs

An epoch refers to each cycle that a model takes through the full training dataset. For example, feeding the neural network with training data for more than one epoch, the result should become better in terms of predicting the given unseen data, which is the test data. In this approach, 30 epochs were applied to achieve the best accuracy results of predicting the test data. Table 1 shows the impact of epochs on the loss percentage.

Epoch Number	Percentage of Loss
0	1.23%
5	0.65%
10	0.40%
15	0.04%
20	0.03%
25	0.02%
30	0.01%

Table 1: Epoch Number and Loss Percentage Results

6.4 Testing

In this very last step of testing, sample testing images were passed through the convolutional neural network, and the predicted and actual true classes were compared. The model achieved 91.5% accuracy results after applying 30 epochs. Figure 27 shows the code segment with the best accuracy result.

```
correct = 0
total = 0
with torch.no_grad():
    for i in tqdm(range(len(test_X))):
        real_class = torch.argmax(test_y[i])
        model_out = net(test_X[i].view(-1, 1, 100, 100))
[0]  # returns a list,
        predicted_class = torch.argmax(model_out)

        if predicted_class == real_class:
            correct += 1
        total += 1
print("Accuracy: ", round(correct/total, 3))
```

```
100%|██████████| 755/755 [00:06<00:00, 111.20it/s]

Accuracy:  0.915
```

Figure 27: Accuracy Results of Testing the Pre-Trained Model

6.5 Transfer Learning

Transfer learning is one of the machine learning techniques which works by reusing a pre-trained model that was originally built for another dataset. Transfer learning then reuses that model as a starting point for a new usually smaller dataset. Transfer learning can be applied for speeding up the model's training time and solve the problem of insufficient data. Usually, while building the CNN models, a lot of time is spent on building and connecting convolutional layers. Transfer learning works by using the previous neural network model to identify edges in the earlier layers, structures in the middle layer, and high-level features in the later layers [37]. The early and middle layers are usually being used, but the last layers are only retrained. Loading the new dataset and applying data resizing, shuffling and grayscale conversion, fine-tuning was applied to improve the accuracy.

6.5.1 Training and Fine-tuning

During the transfer learning process, fine-tuning was applied on the new dataset with a very low learning rate. During this step, the previously constructed CNN model has been used except the output layer. Thus, we need to adjust the output layer to the number of target dataset classes for the new model. For example, in this approach, the initial pre-trained CNN model had two classes, but the new output layer in the CNN model, had three classes. Also, during transfer learning, the learning rate was lowered to 0.0001. Figure 28 presents the fine-tuning procedures.

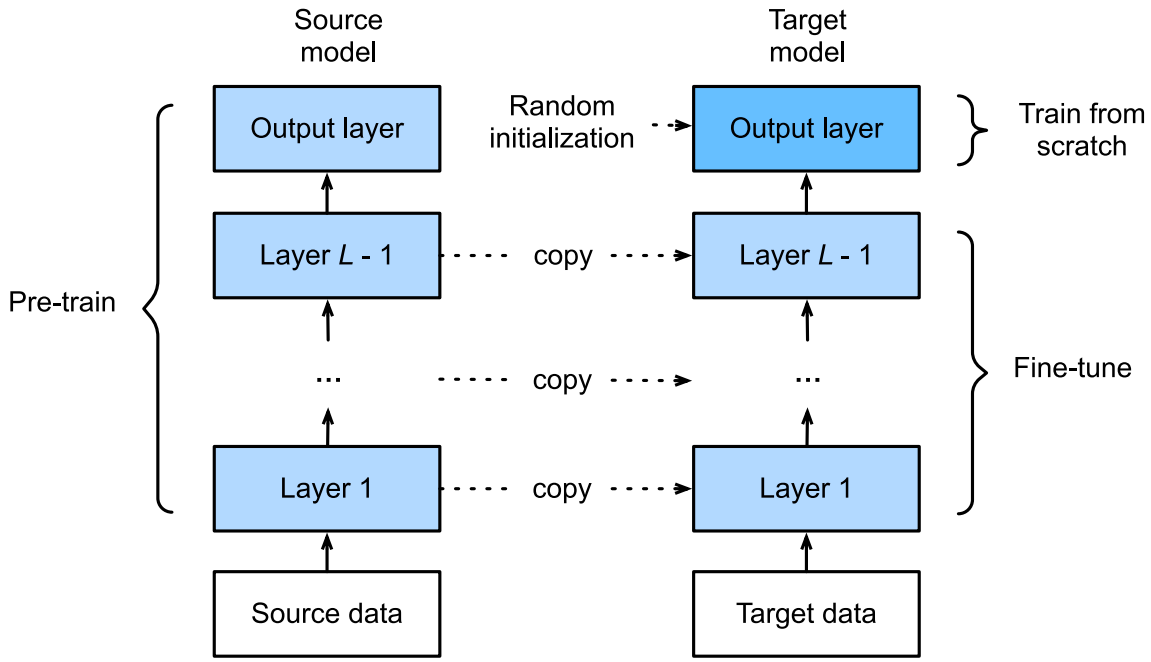


Figure 28: Fine-tuning Steps [37]

After completing the fine-tuning step, the training step took place using the Face Mask Detector Dataset. During the training step, a batch size of 100 was applied to help train the model faster and improve the accuracy performance. Likewise, the MSE Loss function, as well as Adam optimizer, were also used in this new dataset. Besides splitting the data, an optimization algorithm, forward pass, loss function, backward pass, and updating weights were followed.

6.6 Testing

In this very last step, the testing process was applied in order to test whether the new model works. Therefore, by passing sample images through the convolutional neural network, the predicted and the actual class labels were compared. Accuracy was calculated by dividing the number of accurate results by the entire test dataset number. In this approach, the model achieved an accuracy result of 97.1% after applying just 9 epochs. The result was considered a remarkable achievement since classifying people with half of their faces covered is not easy. Figure 29 shows

the code segment with the best accuracy result that was achieved by the transfer learning CNN model.

```
correct = 0
total = 0
with torch.no_grad():
    for i in tqdm(range(len(test_X))):
        real_class = torch.argmax(test_y[i])
        model_out = loaded_net(test_X[i].view(-1, 1, 100, 100))[0]
        predicted_class = torch.argmax(model_out)

        if predicted_class == real_class:
            correct += 1
        total += 1
print("Accuracy: ", round(correct/total, 3))
```

```
100%|██████████| 207/207 [00:01<00:00, 105.33it/s]

Accuracy:  0.971
```

Figure 29: Transfer Learning Accuracy Results

CHAPTER 7

RESULTS AND DISCUSSION

7.1 Matching Predicted Classes with Real Classes

In this thesis, a performance accuracy of 97.1% was achieved from using the Deep Supervised Learning technique, including Convolutional Neural Network (CNN) and Transfer Learning. The approach did not just stop at finding out the accuracy percentage but also printing out the arguments of the maxima with a given data, and results were tested and found accurate as well. In this approach, the predicted class matched with the real class proving that the model worked. People who wear a face mask were set to class 0. People who do not wear a face mask were set to class 1. People who wear a face mask but incorrectly were set to class 2. Figure 30 presents an image of someone who does not wear a face mask and the model was able to predict that the input image matches class 1. Figure 31 illustrates an image of someone who does not wear a mask and the model was able to predict that the input image matches class 0. Figure 32 shows an image of someone who wears a mask but incorrectly, and the model was able to predict that the input image matches class 2.

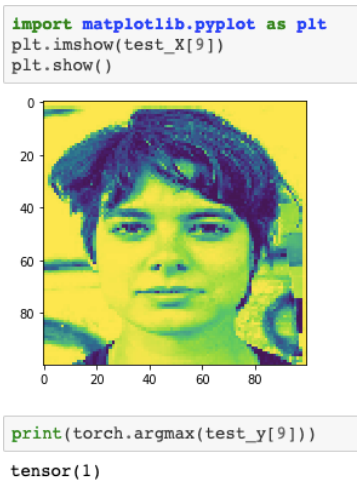
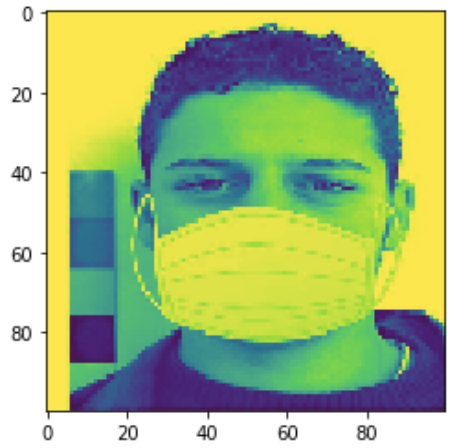


Figure 30: Example of Class 1 Prediction

```
import matplotlib.pyplot as plt
plt.imshow(test_X[25])
plt.show()
```

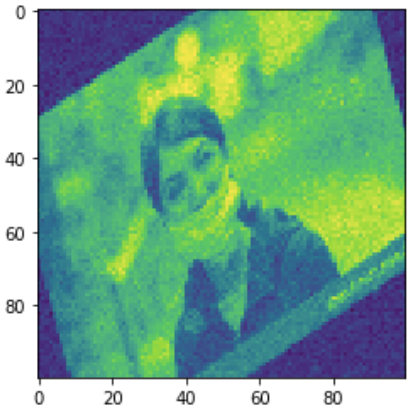



```
print(torch.argmax(test_y[25]))
```

```
tensor(0)
```

Figure 31: Example of Class 0 Prediction

```
import matplotlib.pyplot as plt
plt.imshow(test_X[32])
plt.show()
```




```
print(torch.argmax(test_y[32]))
```

```
tensor(2)
```

Figure 32: Example of Class 2 Prediction

7.2 Pre-trained Model Training vs. Testing Results

In this thesis, to achieve the best accuracy results of the pre-trained model, 30 epochs were applied. Figure 33 illustrates a graph of comparison in loss between training and testing of the pre-trained model over 30 epochs. Also, figure 34 presents a graph of comparison in accuracy training and testing of the pre-trained model over 30 epochs. Each graph illustrates that the loss is getting lower while the accuracy is getting higher when applying more epochs. Figure 35 also shows that with every applied epoch, the training accuracy percentage is getting higher.

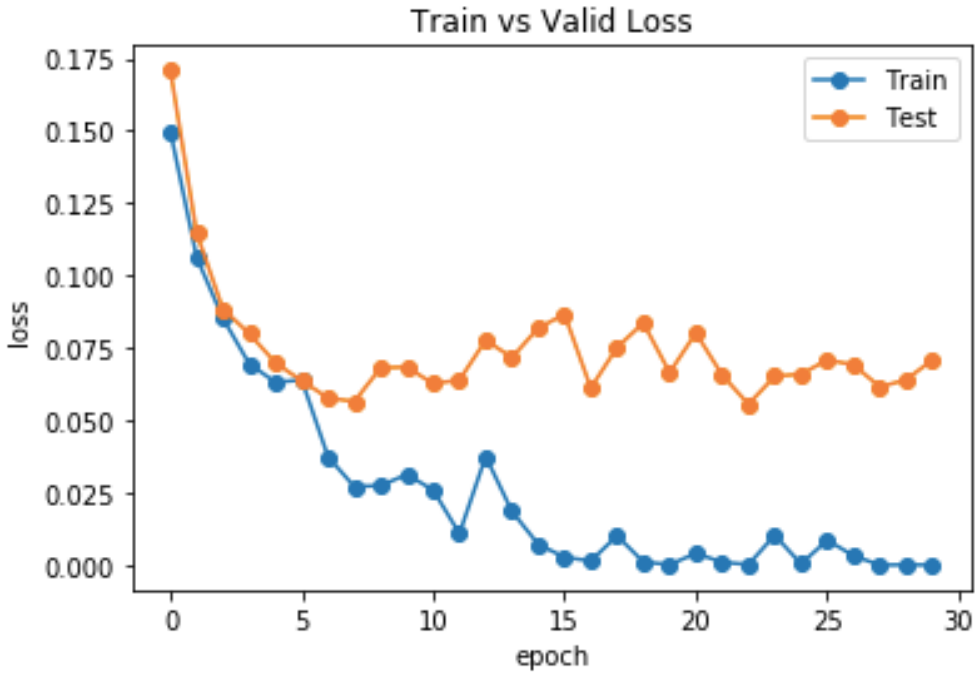


Figure 33: Pre-trained Model Training vs. Testing Loss

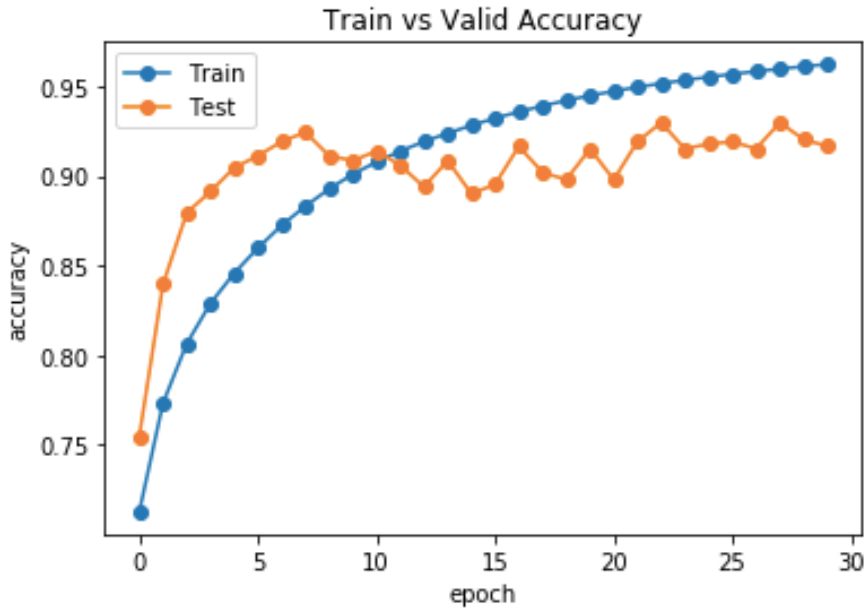


Figure 34: Pre-trained Model Training vs. Testing Accuracy

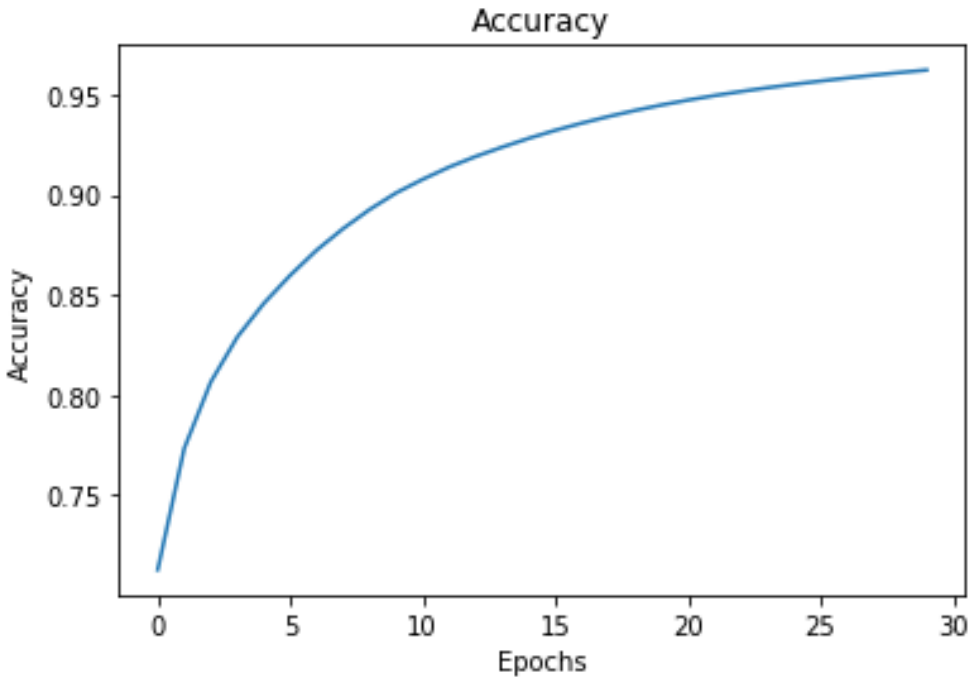


Figure 35: Pre-Trained Model Training Accuracy Results

7.3 Pre-trained Model vs. Transfer Learning Model

In this approach, the Convolutional Neural Network (CNN) model was initially built to train and identify people who wear a face mask. Later, the same CNN model was reused to apply transfer learning to classify a new dataset which also contains people who wear a face mask incorrectly. Table 2 presents a comparison in loss between the pre-trained model and transfer learning model. In this approach, the pre-trained model achieved 91.5% accuracy results after applying 30 epochs, while the transfer learning model achieved 97.1% accuracy results after applying 9 epochs only. As the graph shows, the loss started much lower in transfer learning than the pre-trained model, which indicates the application of transfer learning and fine-tuning techniques improved the model's accuracy results. Likewise, figures 36 and 37 present graphs indicating a comparison of loss and accuracy results between training and testing of the transfer learning model.

Epoch Number	Pre-trained Model Loss	Transfer Learning Loss
1	0.12381	0.00080
2	0.09998	0.00075
3	0.07724	0.00072
4	0.06642	0.00069
5	0.06532	0.00066
6	0.05882	0.00063
7	0.04837	0.00059
8	0.05490	0.00056
9	0.05534	0.00053

Table 2: Pre-trained vs. Transfer Learning Loss

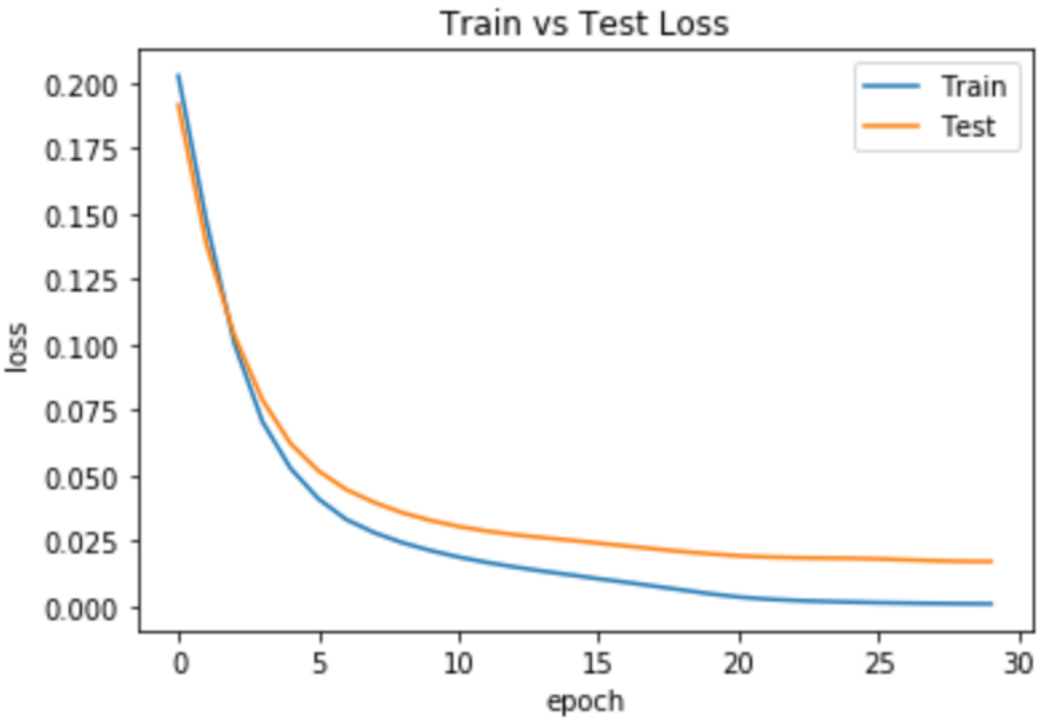


Figure 36: Transfer Learning Model Training vs Testing Loss

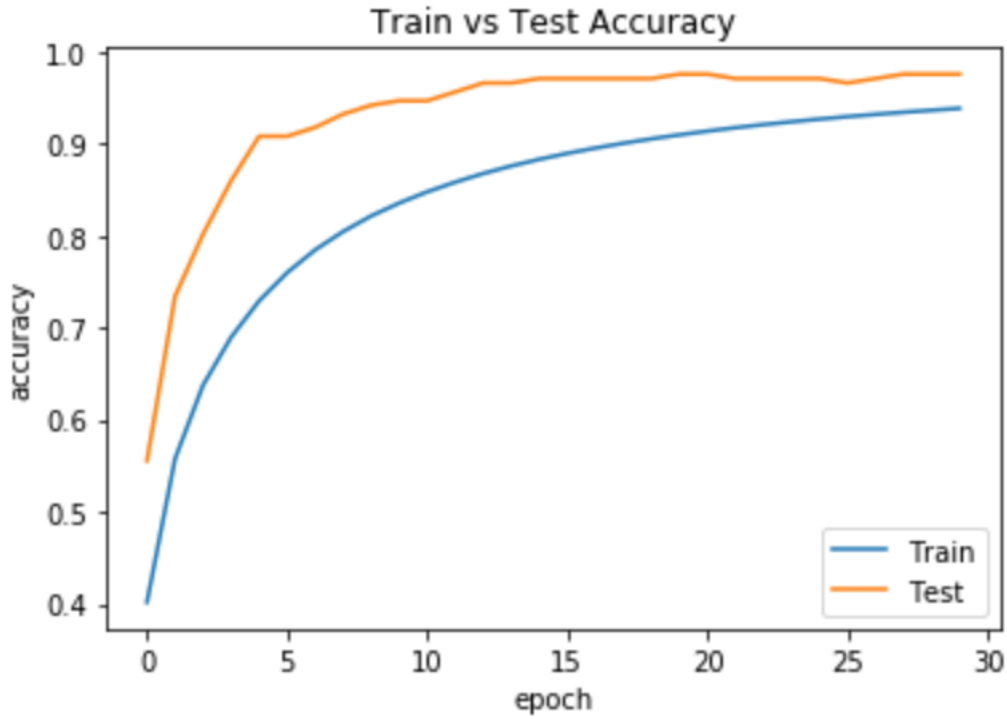


Figure 37: Transfer Learning Model Training vs Testing Accuracy

7.4 Predicting Real-World Images

In this experiment, to ensure the effectiveness and accuracy of the proposed model, testing and predicting real-world images were proposed. In this approach, the CNN model was deployed on images from mobile devices. This approach was presented with new challenges due to the illumination, angle, and resolution of images. All the mentioned challenges were not described in training and testing previous datasets. Feeding real-world images into the proposed model is important to test the effectiveness of the model. Correct predictions indicate that the model is reliably integrated with designing a real-world application for classifying facial recognition and face mask detection. As previously mentioned, people who wear a face mask were labeled with class 0, people who do not wear a face mask were labeled with class 1, and people who wear a face mask but incorrectly were labeled with class 2. Figures 38, 39, and 40 present examples of real-world mobile images that were used to conclude that the proposed CNN model was able to match each image with its correct class. This indicates that the CNN model of this experiment is highly effective and provides accurate results.

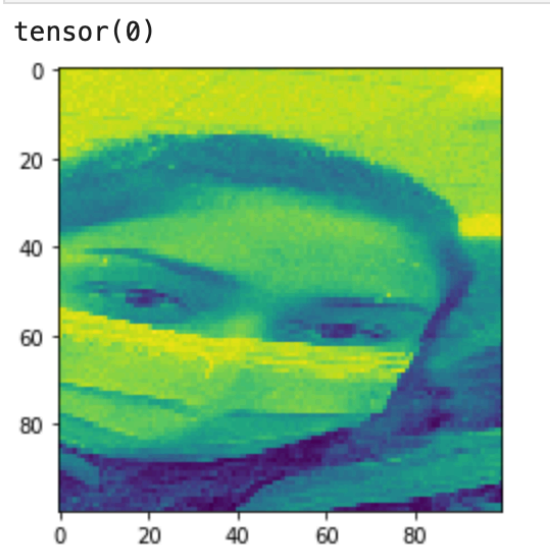


Figure 38: Real-world Image of Class 0 Prediction

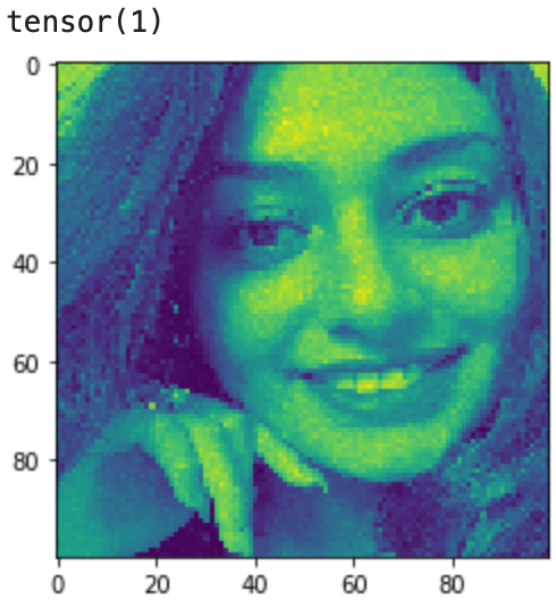


Figure 39: Real-world Image of Class 1 Prediction

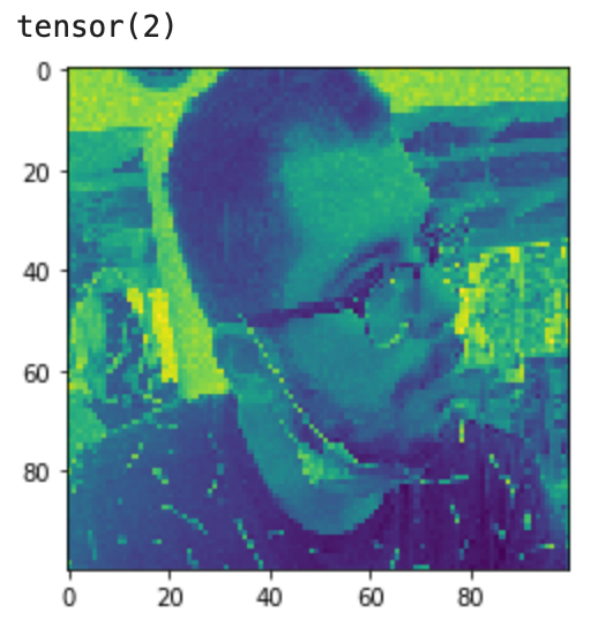


Figure 40: Real-world Image of Class 2 Prediction

7.5 Experiment 2

In this study, the results of the previous approach (experiment 1) presented the transfer learning model, which contained three different classes, learning from the pre-trained model,

which included only two classes. Additionally, experiment 1 developed significant performance accuracy results of 97.1%.

In experiment 2, Face Mask Detection Dataset was used for the pre-training of the CNN model, but another class was added. The extra class was collected from the Face Mask Detector dataset. The class included images of people who wear a face mask but incorrectly. With that said, the pre-trained model in this experiment included three classes, which are people who wear a face mask, people who do not wear a face mask, and people who wear a face mask but incorrectly.

7.5.1 Experiment 2 Pre-trained Model Training and Testing

In this experiment, the CNN model was build using three convolutional layers by setting kernel size to 5x5, and the stride was set to 1. Also, after each convolutional layer, ReLU was applied. Likewise, 2x2 max pooling was also used as well as two fully connected layers were added. Additionally, batch size was set to 100, and the learning rate was set to 0.001. Moreover, Adam optimizer and MSE loss function were used during this experiment. The model was set to run for 30 epochs. After 30 epochs, the model achieved 87.5% accuracy results, which are lower than Experiment 1 pre-trained accuracy results. Figure 41 presents the best testing accuracy result was achieved during the pre-trained model.

```
correct = 0
total = 0
with torch.no_grad():
    for i in tqdm(range(len(test_X))):
        real_class = torch.argmax(test_y[i])
        model_out = net(test_X[i].view(-1, 1, 100, 100))[0]  # retu
rns a list,
        predicted_class = torch.argmax(model_out)

        if predicted_class == real_class:
            correct += 1
        total += 1
print("Accuracy: ", round(correct/total, 3))
```

```
100%|██████████| 848/848 [00:06<00:00, 125.69it/s]
Accuracy:  0.875
```

Figure 41: Experiment 2 Pre-Trained Model Accuracy Result

7.5.2 Experiment 2 Transfer Learning Training and Testing Results

In experiment 2, the Face Mask Detector dataset was applied without the third class, namely images of people who incorrectly wear a face mask. The transfer learning model was built using only two classes, which include people who wear a face mask, and people who do not wear a face mask. During this step, the saved CNN model had been used except the output layer. Thus, adjusting the output layer to the number of target dataset classes was an essential step. For example, the initial pre-trained CNN model had three classes, but the new output layer in the CNN model, had two classes. Also, during the transfer learning, the learning rate was lowered to 0.0001, a batch size of 100, MSE Loss function, and Adam optimizer were used. In this experiment, the model achieved a testing accuracy result of a 98.5% after applying just 9 epochs. Figure 42 presents the testing accuracy results after applying transfer learning techniques.

```
correct = 0
total = 0
with torch.no_grad():
    for i in tqdm(range(len(test_X))):
        real_class = torch.argmax(test_y[i])
        model_out = loaded_model_test(test_X[i].view(-1, 1, 100, 10
0))[0]  # returns a list,
        predicted_class = torch.argmax(model_out)

        if predicted_class == real_class:
            correct += 1
        total += 1
print("Accuracy: ", round(correct/total, 3))
```

```
100%|██████████| 137/137 [00:01<00:00, 127.65it/s]

Accuracy:  0.985
```

Figure 42: Experiment 2 Transfer Learning Accuracy Results

7.6 Experiment 1 and Experiment 2 Comparison

In this thesis, two different experiments were proposed. The second experiment showed better accuracy results using transfer learning technique. However, the first experiment was mainly focused on because transferring the knowledge of the pre-trained CNN model from two classes to

three classes was more challengeable. Yet, the model was able to still achieve remarkable accuracy results. Table 3 presents a comparison of loss results between experiment 1 and experiment 2 during the transfer learning process.

Epoch Number	Experiment 1 Transfer Learning Loss	Experiment 2 Transfer Learning Loss
1	0.00080	0.08965
2	0.00075	0.06122
3	0.00072	0.04906
4	0.00069	0.04346
5	0.00066	0.04052
6	0.00063	0.03538
7	0.00059	0.03033
8	0.00056	0.02853
9	0.00053	0.02758

Table 3: Experiment 1 vs Experiment 2 Comparison

CHAPTER 8

CONCLUSION AND FUTURE WORK

This thesis presented a study on facial recognition and face mask detection through deep learning techniques by building a CNN model using transfer learning and fine-tuning techniques. This process gave accurate and quick results for facial recognition security systems despite the fact that half of the faces are covered with face masks. The test results show a high accuracy rate in identifying individuals wearing a face mask, not wearing a face mask, and wearing a face mask but in an incorrect manner. The model was able to achieve a 97.1% of performance accuracy, which is a significant achievement. Moreover, the study presented a useful tool in fighting the spread of the COVID-19 disease by allowing all individuals to wear a face mask while performing biometric authentication.

Facial recognition with face mask is becoming more and more importance over the past year due to the spread of the COVID-19 virus. Our future works include alarm if somebody is not wearing a face mask properly, and detection of the social distancing.

.

Bibliography

[1] L. Mayron, Y. Hausawi and G. Bahr, "Secure, Usable Biometric Authentication Systems," July 2013. [Online]. Available: https://www.researchgate.net/publication/262346173_Secure_Usable_Biometric_Authentication_Systems.

[2] R. Chaudhari, A. Pawar and R. Deore, "The Historical Development of Biometric Authentication Techniques," 10 October 2013. [Online]. Available: https://www.ijert.org/research/the-historical-development-of-biometric-authentication-techniques-a-recent-overview-IJERTV2IS101132.pdf.

[3] T. Group, "Biometrics: definition, use cases and latest news," Thales Group, 06 April 2021. [Online]. Available: https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/inspired/biometrics.

[4] A. Hunter, "Why Biometric Authentication is Better than Passwords," 4 November 2019. [Online]. Available: https://hakin9.org/why-biometric-authentication-is-better-than-passwords/.

[5] K. Boeckl, "Back to basics: Multi-factor authentication (MFA)," NIST, 19 April 2021. [Online]. Available: https://www.nist.gov/itl/applied-cybersecurity/tig/back-basics-multi-factor-authentication.

[6] E. McKeown, "What Is Multi-factor Authentication (MFA)?," Ping Identity, 03 September 2020. [Online]. Available: https://www.pingidentity.com/en/company/blog/posts/2017/what-is-multi-factor-authentication-mfa.html.

[7] Kaspersky, "Kaspersky," 26 April 2021. [Online]. Available: https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition.

[8] J. Schuppe, "How Facial recognition became a routine policing tool in America," NBC News, 11 May 2019. [Online]. Available: https://www.nbcnews.com/news/us-news/how-facial-recognition-became-routine-policing-tool-america-n1004251.

[9] T. Waitt, "The Truth about Law Enforcement's Use of Facial Recognition Technology," American Security Today, 19 April 2018. [Online]. Available: https://americansecuritytoday.com/truth-law-enforcements-use-facial-recognition-technology/.

[10] R. Gambler, "Facial Recognition Technology," United States Government Accountability Office, September 2020. [Online]. Available: https://www.gao.gov/assets/gao-20-568.pdf.

[11] Coolfire, "How U.S. Airports Use Facial Recognition," Coolfire Core Solutions, 29 Novemeber 2018. [Online]. Available: https://www.coolfiresolutions.com/blog/airport-facial-recognition-technology/.

[12] Z. L. L. Z. R. M. X. D. J. H. Meng Shen, "IriTrack: Liveness Detection Using Irises Tracking for Preventing Face Spoofing Attacks," October 2018. [Online]. Available: https://arxiv.org/pdf/1810.03323.pdf.

[13] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," Sustainable cities and society, March 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7775036/#sec0085.

[14] M. Loey, G. Mangogaran, T. M.H.N. and K. N.E.M., "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," National Library of Medicine, 1 January 2021. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32834324/.

[15] J. L. Q. Y. a. Z. L. S. Ge, "Detecting Masked Faces in the Wild with LLE-CNNs," IEEE Conference on Computer Vision and Pattern Recognition, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8099536.

[16] M. R. I. M. S. a. A. S. M. S. Ejaz, "Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition," 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8934543.

[17] S. V. M. a. N. V. Dionisio, "Real-Time Facemask Recognition with Alarm System using Deep Learning," IEEE Control and System Graduate Research Colloquium (ICSGRC), 2020. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9232610.

[18] A. Tarhini, "Face Recognition: An Introduction," Ali Tarhini, 5 December 2010. [Online]. Available: https://alitarhini.wordpress.com/2010/12/05/face-recognition-an-introduction/.

[19] S. R. B. a. J. S. Kumar, "Geometric shaped facial feature extraction for face recognition," 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7887965.

[20] V. Varshney, "CNNs: The Key to Computer Vision," Data Driven Investor, 14 February 2020. [Online]. Available: https://medium.datadriveninvestor.com/cnns-the-key-to-computer-vision-29c6fe1c6fdc.

[21] T. P. o. A. PAI, "Understanding Facial Recognition Systems," 19 February 2020. [Online]. Available: https://www.partnershiponai.org/wp-content/uploads/2020/02/Understanding-Facial-Recognition-Paper_final.pdf.

[22] V. Gupta and S. Mallick, "Face Recognition: An Introduction," Learn OpenCV, 16 April 2019. [Online]. Available: https://learnopencv.com/face-recognition-an-introduction-for-beginners/.

[23] A. Othman, "What Is A Biometric System, and How To Secure It," Veridium, 19 July 2018. [Online]. Available: https://www.veridiumid.com/biometric-system-secure/.

[24] M. U. A. A. D. Y. G. M. V. P. a. M. K. M. Sivaram, "Biometric Security and Performance Metrics: FAR, FER, CER, FRR," International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2019. [Online]. Available: https://ieeexplore.ieee.org/document/9004275?.

[25] L. Li, Z. Xia, X. Jiang, Y. Ma, F. Roli and X. Feng, "3D Face Mask Presentation Attack Detection Based on Intrinsic Image Analysis," 28 March 2019. [Online]. Available: https://arxiv.org/pdf/1903.11303.pdf.

[26] D. YEUNG, R. BALEBAKO, C. I. GUTIERREZ and M. CHAYKOWSKY, "Face Recognition Technologies," 2020. [Online]. Available: https://www.rand.org/content/dam/rand/pubs/research_reports/RR4200/RR4226/RAND_RR4226.pdf.

[27] R. Materese, "NIST Launches Studies into Masks' Effect on Face Recognition Software," NIST, 4 August 2020. [Online]. Available: https://www.nist.gov/news-events/news/2020/07/nist-launches-studies-masks-effect-face-recognition-software.

[28] T. S. Administration, "TSA to implement Executive Order regarding face masks at airport security checkpoints and throughout the transportation network," 31 January 2021. [Online]. Available: https://www.tsa.gov/news/press/releases/2021/01/31/tsa-implement-executive-order-regarding-face-masks-airport-security.

[29] D. o. H. Security, "News Release: Airport Screening While Wearing Masks Test," 04 January 2021. [Online]. Available: https://www.dhs.gov/science-and-technology/news/2021/01/04/news-release-airport-screening-while-wearing-masks-test.

[30] O. Gurav, "Face Mask Detection Dataset," 2020. [Online]. Available: https://www.kaggle.com/omkargurav/face-mask-dataset.

[31] S. Patnaik, "Face mask detector," 2021. [Online]. Available: https://www.kaggle.com/spandanpatnaik09/face-mask-detectormask-not-mask-incorrect-mask.

[32] T. A. M. a. S. A.-Z. S. Albawi, "Understanding of a convolutional neural network," 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8308186.

[33] H. &. G. P. &. S. M. K. &. M.-M. R. &. B. P. &. S. V. Panwar, "A Deep Learning and Grad-CAM based Color Visualization Approach for Fast Detection of COVID-19 Cases using Chest X-ray and CT-Scan Images.," Researchgate, 2020. [Online]. Available: https://www.researchgate.net/publication/343508866_A_Deep_Learning_and_Grad-CAM_based_Color_Visualization_Approach_for_Fast_Detection_of_COVID-19_Cases_using_Chest_X-ray_and_CT-Scan_Images/.

[34] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way," Towards Data Science, 15 December 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[35] Prakharry, "Intuition of Adam Optimizer," 24 October 2020. [Online]. Available: https://www.geeksforgeeks.org/intuition-of-adam-optimizer/.

[36] M. Binieli, "Machine learning: an introduction to mean squared error and regression lines," 16 October 2018. [Online]. Available: https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/.

[37] M. Burugupalli, "IMAGE CLASSIFICATION USING TRANSFER LEARNING AND CONVOLUTION NEURAL NETWORKS," July 2020. [Online]. Available: https://library.ndsu.edu/ir/bitstream/handle/10365/31517/Image%20Classification%20Using%20Transfer%20Learning%20and%20Convolution%20Neural%20Networks.pdf?sequence=1.

[38] Z. Elhamraoui, "Fine-tuning in Deep Learning," Artificial Intellgence, 23 June 2020. [Online]. Available: https://ai.plainenglish.io/fine-tuning-in-deep-learning-909666d4c151.