Montclair State University

Montclair State University Digital Commons

Theses, Dissertations and Culminating Projects

5-2021

Fingerprint Classification Using Transfer Learning Technique

Aseel H. Aloweiwi
Montclair State University

Follow this and additional works at: https://digitalcommons.montclair.edu/etd

 Part of the Computer Sciences Commons

Recommended Citation

Aloweiwi, Aseel H., "Fingerprint Classification Using Transfer Learning Technique" (2021). *Theses, Dissertations and Culminating Projects*. 723.
https://digitalcommons.montclair.edu/etd/723

This Thesis is brought to you for free and open access by Montclair State University Digital Commons. It has been accepted for inclusion in Theses, Dissertations and Culminating Projects by an authorized administrator of Montclair State University Digital Commons. For more information, please contact digitalcommons@montclair.edu.

ABSTRACT

Fingerprints play a significant role in many sectors. Nowadays, fingerprints are used for identification purposes in criminal investigations. They are also used as an authentication method since they are considered more secure than passwords. Fingerprint sensors are already widely deployed in many devices, including mobile phones and smart locks. Criminals try to compromise biometric fingerprint systems by purposely altering their fingerprints or entering fake ones. Therefore, it is critical to design and develop a highly accurate fingerprint classification. However, some fingerprint datasets are small and not sufficient to train a neural network. Thus, transfer learning is utilized. A large Sokoto Coventry Fingerprint Dataset (SOCOFing), which contains 55,273 fingerprint images, was first used to train a convolutional neural network model to detect image alteration and level of alternations. The model was able to achieve an 81% of accuracy. Then, a few layers of SOCOFing model were used and adapted to train another smaller dataset, namely ATVS-FakeFingerprint Database (ATVS-FFp DB), which contains 3,168 fingerprint images. Twomodels were trained. The first transferring model was built to classify images into real and fake, and a remarkable classification accuracy of 99.4% was achieved. The second transferring model was used to detect if the image was fake and if the user was cooperating in the generated faked fingerprint. The model achieved a classification accuracy of 97.5%. The transfer learning technique proves to be very effective in addressing insufficient dataset issues for deep learning.

Keywords – Fingerprint Biometric System, Fingerprint, Feature Extraction, Classification, CNN, Transfer Learning, Fake, Real, Altered.

MONTCLAIR STATE UNIVERSITY

Fingerprint Classification Using Transfer Learning Technique

by

Aseel H. Aloweiwi

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

FINGERPRINT CLASSIFICATION TRANSFER LEARNING TECHNIQUE

A THESIS

Submitted in partial fulfillment of the requirements

For the degree of Master of Science

by

Aseel H. Aloweiwi

Montclair State University

Montclair, NJ

Copyright © 2021 by *Aseel H. Aloweiwi*. All rights reserved

ACKNOWLEDGMENTS

First and foremost, all thanks and praises to Allah (S.W.T) for His showers of blessings throughout my research work to complete this thesis successfully.

Special appreciation goes to my advisor Dr. Michelle Zhu, a professor and associate chair for the computer science department, for her supervision and constant support. My appreciation also goes to my co-advisor Dr. Tianyang Wang for his help and knowledge regarding this topic. Not forgotten, I would like to thank the rest of my thesis committee members: Dr. Weitian Wang and Dr. Jiayin Wang for their encouragement and insightful comments.

I would like to express my sincere gratitude to my colleague and friend Mira M. Boulos for her assistance at every stage of the research work. Her enthusiasm, motivation, and sense of humor when the times got rough are much appreciated.

I am deeply grateful to my parents for their caring and endless love. Any attempt at any level cannot be satisfactorily completed without their prayers. My parents-in-law deserve my wholehearted thanks as well for their support and love. I would like to extend my sincere thanks to my brother, Anas Aloweiwi, for his guidance and help. It was a great comfort and relief to know that you were willing to help whenever I am in need. Not forgotten, I would like to offer my heartfelt thanks to my brother, Yazan Aloweiwi, for his care and kindness.

Finally, my thanks go to my husband, Moh'd Azzam, whom without his support, this journey would not have been possible. Thank you for believing in me, helping in every single aspect, and remembering my due dates more than me. The last word goes for my little kids, Raya and Hashem, who have given me the extra strength and motivation to get things done.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

TABLE OF EQUATIONS

CHAPTER 1: INTRODUCTION

Authentication becomes an integrated part of almost every individual's life. It is one of the fundamental aspects of protecting the users' data. The user needs to prove who he/she claims to be in order to access privileged operations. Biometric technology has proved to be more secure compared to traditional methods such as passwords, answers to a prearranged set of questions, and smart cards. Traditional authentication systems are based on authenticating the individual either based on knowledge or something the user possesses like tokens. The question that always arises is: What if the password has been compromised or the token has been stolen? Google surveyed 3,000 users in 2019, and the results showed that every two users out of three use the same password for multiple accounts [1]. Therefore, if a hacker can compromise a password, he gets access to the victim's multiple accounts. To address the password problem, biometric authentication systems have been adopted by many companies to secure their users' accounts and personal data.

Biometric systems use distinct physical or behavioral characteristics to confirm an individual's identity. They are based on authenticating the individual either statically or dynamically. Static biometrics refers to physical features such as a fingerprint, iris, and face. This type of authentication is convenient and easy to use. It does not require memorizing any passwords or possess any tokens. It simply authenticates based on something the individual is. At the same time, the static property of the physical features may lead to security flaws. Once the data has been compromised, it cannot be reset. On the other hand, dynamic biometrics use behavioral characteristics such as a voice pattern and typing rhythm. It authenticates based on something the individual does. Dynamic biometrics are considered more secure than static biometrics because it examines the user behavior which is difficult to be mimicked [2].

The fingerprint is the first biometric trait that has been used for identification and

authentication purposes. Earlier in the 19th century, the United States started using fingerprints for identification purposes in criminal investigations. Nowadays, fingerprints are being used in police investigations, driver license registration, mobile phone authentication, and much more [3].

CHAPTER 2: RELATED WORKS

In “A Review of Fingerprint Image Pre-processing” [4], the authors reviewed several preprocessing techniques such as normalization and segmentation. Normalization transferred all grey-image intensity values to a desired range of values and thus improved the grayscale image. They emphasized that normalization was important to get rid of the effects caused by sensor noise or finger different pressure. Two approaches were mentioned: the first was based on the convolution of the image with the use of the Gabor filter, and the second was an adaptive fingerprint image normalization method. The first approach can only be implemented on a local mode since the mean and variance can change at different regions of the image. Then, they pointed out that two noise filters could enhance the image if they were used. The Gaussian filter performed linear smoothing while the Gradient filter performed non-linear smoothing. Lastly, they described the fingerprint segmentation process. They explained the two steps of the segmentation, namely block-wise and bit-wise. The block-wise step was responsible for extracting the foreground of the image, while the bit-wise step was responsible for removing the noise from the extracted foreground image. They indicated that the bit-wise step was time-consuming, so it was skipped. They concluded that the performance of fingerprint recognition would be improved using those preprocessing techniques.

Authors of “Fingerprint Alternations Type Detection Using Deep Convolutional Neural Network” [5] used a publicly available Coventry Fingerprint Dataset (CovFingDataset) for their research. The dataset contains 10 real fingerprints from 611 individuals. Each image has unique attributes such as gender, to which hand the finger belongs, and the finger name. The dataset also contains a total of 55,249 images that are alternated into three levels: z-cut, obliteration, and central

rotation synthetic. They proposed a convolutional neural network model that was able to detect if the image was altered and to which level of alternations it belonged. The proposed model achieved a classification accuracy of 98.55%. They also fine-tuned the ResNet model - that was trained on ImageNet – and tested it on CovFingDataset. The fine-tuned model achieved a classification accuracy of 99.86%. Although the ResNet18 model achieved better accuracy results, their proposed model achieved precision and recall score of 100% on real images while the ResNet18 model confused real fingerprints with altered ones.

In "Altered Fingerprints: Analysis and Detection" [6], the authors introduced the difference between fingerprint spoofing and fingerprint alternation. The first is used to adopt another individual's identity, while the last is used to mask an individual's identity. Due to the success of Automated Fingerprint Identification Systems (AFIS) in law enforcement and civilian applications, criminals tend to alternate their fingerprints. The authors evaluated a well-known fingerprint image quality assessment software, NFIQ, and the results showed that NFIQ has limited ability in distinguishing real fingerprints from altered ones. They proposed an algorithm in which altered fingerprints can be detected based on analyzing orientation field and minutiae distribution. The proposed algorithm and NFIQ had been tested on a large public database (NIST SD14) of altered fingerprints provided by a law enforcement agency. The proposed algorithm successfully detected 66.4% of the subjects with altered fingerprints while NFIQ detected 26.5% of such subjects.

Authors of "Anti-spoofing method for fingerprint recognition using patch-based deep learning machine" [7] introduced fingerprint spoofing and how it could be achieved using fabricated materials. They proposed a deep learning model that was based on Discriminative

Restricted Boltzmann Machines (DRBM) and Deep Boltzmann Machine (DBM) to distinguish real fingerprints from fake ones. The model was employed to extract deep features of the grayscale fingerprints. KNN classifier was used to examine spoof forgeries. The performance of the model was robust for different kinds of spoof forgeries such as wood glue, Gelatin, and playdoh. However, the model was struggled to distinguish fake fingerprints with unknown materials.

The authors of "Overview of Fingerprint Recognition System" [8] reviewed various studies regarding fingerprint recognition. They explained the main four stages of a biometric fingerprint system and created a table of the recent work for each stage. In addition, they listed the different types of databases that are used in fingerprint systems with some characteristics, such as the total number of images that can be stored and the image size.

CHAPTER 3: FINGERPRINT BIOMETRIC SYSTEM

Fingerprint biometric systems offer a reliable solution to the problems associated with traditional authentication methods. They can be used either for identification purposes (e.g., identify criminals) or for authentication purposes (e.g., accessing a mobile phone). Fingerprint sensors are already widely spread in many applications such as mobile phones. In addition, the future holds many other applications that may adopt fingerprint systems such as payments and home appliances. This chapter introduces some basic concepts about fingerprints, then sheds light on the structure of fingerprint systems.

3.1 FINGERPRINT FEATURES

Fingerprint systems are dominant among the other authentication systems for many reasons, which are:

- Fingerprint uniqueness: Every individual has his unique fingerprints. This fact also applies to identical twins; although they share the same DNA, they have different fingerprints.
- Fingerprint permanence: Fingerprints remain the same for the individual's lifetime period, unlike facial features, which may change as the person gets older. This property makes them suitable as long-term markers for an individual's identity.
- Less privacy intrusion: Scanning a fingerprint is less intrusive to the person's privacy in comparison with taking a picture or speaking into a microphone.
- Cost-efficient: Fingerprint authentication systems are lower in cost more than other biometric systems such as irises.

3.2 BASIC CONCEPTS

Fingerprints are mainly referred to as the lines that making them. The black lines are called ridges and the white area between them are called valleys. Together, they make up different patterns. Figure 1 shows the main three patterns: arch, loop, and whorl [9].

Figure 1: Fingerprint Patterns [9]

The core of a fingerprint is the most inner recurve at the center of the pattern. Delta is where three areas converge forming a triangle shape. Figure 2 shows the core and delta of a fingerprint. The pattern is arch when the lines start from one side, rise in the middle, and exit to the other side forming a hill. Usually, there are no cores or deltas. Only 5% of the population has an arch pattern. The pattern is a loop when the lines start from one side, loop at the middle and exist out to the same side. It usually has one core and one delta. The loop may face either left or right. Loop pattern is the most common pattern among the human population. The pattern is a whorl when the lines form a round shape. Usually, it has two cores and two deltas. If a fingerprint has multiple patterns or does not fall clearly under any of the categories, then it is called an accidental whorl. Occasionally, the ridges start and stop as they flow through the pattern. When the ridges' structure changes, it can form different features called minutiae such as ridge ending, bifurcation, and dot as shown in figure 2 [10]. When the line starts and stops such that its length is shorter than the average length of the other lines, then a ridge ending has been formed. Some ridges may split into two other ridges forming a bifurcation. A dot is formed when the ridge starts and

ends at a very small distance. Minutiae are the reason behind the uniqueness of every fingerprint [11], [12], [13].

Figure 2: Fingerprint Minutiae [10]

3.3 STAGES OF FINGERPRINT BIOMETRIC SYSTEM

Any biometric system consists of several modules to achieve its functionality. This section gives a detailed explanation of each stage in fingerprint biometric systems.

3.3.1 Image Acquisition Stage

Image acquisition is the process in which the image is being captured and converted into digital format. Offline and online methods are being used for this purpose [8]. The Offline method depends on the usage of ink. An inked fingertip is being pressed on a white paper sheet and then scanned to get the digital format. However, the online method depends on the live-scan technology that uses sensors to scan the fingertip and give the digital format immediately. The online method is dominant nowadays due to its ease and speed. The online method eliminates the usage of ink, gets an instant file of the image, finds the quality of the image before recording, and gives the availability to get multiple copies of it if needed [14]. Swipe sensors or touch sensors can be used to get the live-scan fingerprint. Swipe sensors tend to swipe the fingerprint row by row, while touch sensors capture the full fingerprint by one scan. Swipe sensors give more accurate images than touch sensors. As a result, less complicated matching algorithms are being used. However,

touch sensors are more convenient to the user because they are faster in scanning his fingerprint [12]. One of the widely used fingerprint sensors is the optical sensors.

3.3.2 Image Preprocessing Stage

Image preprocessing is the process in which the image is subjected to some preprocessing techniques to get the image as clear as possible, such as obtain a high-quality image. A clear fingerprint image has a high contrast between ridges and valleys [4]. One of the methods that can be applied to obtain a sharper fingerprint image is to set a threshold value. Any area that is lighter than the threshold will be discarded. On the other hand, any area that is darker than the threshold will be marked as black [12]. More techniques can be applied to the image to get an enhanced picture such as image thinning and binarization. Image thinning is the process in which the ridge width is being reduced to one pixel. Binarization is the process in which the image is transformed from 256 levels into two levels (0 and 1) refers to (white and black) respectively [8]. Figure 3 shows a fingerprint after being binarized and thinned [15].

Figure 3: Image Preprocessing: a) Original Image b) Binarized Image c) Thinned Image [16]

3.3.3 Feature Extraction Stage

Feature extraction is a process that is applied to the output image of the preprocessing step. It extracts the significant features from the fingerprint image. The fingerprint image that is captured by the sensor is an 8-bit greyscale image. Each image will require a few Mbytes of storage

depending on the sensor size. Some compression methods like JPEG can be applied to compress the image. Even after the compression, the image will still occupy some significant memory space. For that reason, feature extraction is being applied. Instead of storing the full image, the extraction feature set is stored and as a result, occupies less memory space. And most importantly, the matching algorithm that will be used will be simpler after extracting the features.

Different feature extraction methods can be applied. It depends on the matching algorithm that will be used. If a minutiae-based matching algorithm is used, then the feature extraction step is responsible for locating minutiae points. Minutiae points can be located by fetching the ridge endings and bifurcations and marking them as shown in figure 4 [16]. Once a minutiae point is identified, its location is registered as the distance between the point and the core. The angle of the minutiae point is also registered i.e., the angle of the ridge when it terminates. In addition, a minutiae point can be classified by its type and quantity which helps in searching the database quicker [12].

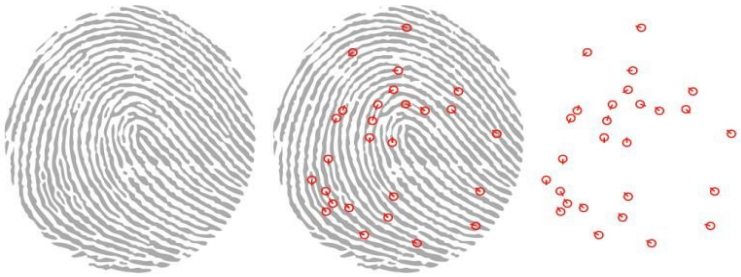

Figure 4: Extracting Features [16]

3.3.4 Matching Stage

Fresh template is compared with all the reference templates stored in the database for identification purposes, forming a 1: N matching in the matching process. For authentication purposes, the fresh template is compared with one reference template forming a 1:1 matching.

Fingerprint matching techniques can be split into two main categories: Minutiae based and non-minutiae-based.

The Minutiae-based matching algorithm performs a comparison between the minutiae points of the fingerprint – that has been extracted in the feature extraction stage - and the minutiae points of the stored templates to find a similarity. This algorithm requires a large area of skin to work with. Therefore, swipe sensors are usually used. Governments use the minutiae-based technique to identify criminals. Crime scene fingerprint is compared against those stored in the database. Successful matching demands the minutiae points to be extracted with care so that the matching characteristics can be found, and the matching algorithm can perform an efficient comparison. If 8-12 points of similarity are found, then there is a good chance there is a match. Figure 5 shows fingerprint matching based on minutiae [17].

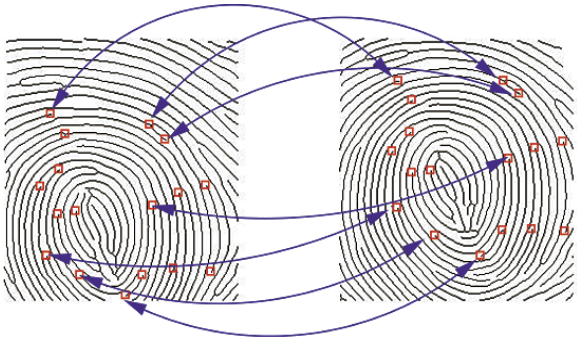

Figure 5: Fingerprints Matching Based on Minutiae [17]

Sometimes it is not easy to accurately determine the minutiae points due to the poor image quality. Therefore, the matching accuracy will be lower. The non-minutiae-based technique overcomes this problem. It extracts some other features from the ridge patterns such as local orientation and frequency, ridge shape, and texture information [18]. Those features give more discriminatory characteristics and therefore preprocessing techniques may also be skipped. Then

the extracted features of the input image are compared with the stored templates to see to what degree they match.

Hybrid methods combine both minutiae-based and non-minutiae-based techniques. Because sensor sizes become smaller in size, the use of hybrid methods becomes highly demanded. Like the sensors on mobile phones, they are small, but the minutiae part still being used.

3.4 ACCURACY

The matching accuracy of a fingerprint authentication system depends on the stability of the fingerprint over time. For a specific individual, if there is a significant change between the fingerprint provided and the fingerprint stored in the database, then the authentication will be rejected. Many factors may contribute to acquiring a different fingerprint associated with the same individual, such as improper interaction with the sensor (partial fingerprint), temporary change in the fingerprint (some cuts or scars), or some environmental factors (weather is dry).

One way to decrease the number of false rejections is to store multiple fingerprints for the same individual. For example, storing different portions of the fingerprint assume that the user will place his finger in various ways. However, storing more templates needs more storage and computational capabilities [12].

CHAPTER 4: THREATS AND COUNTERMEASURES

Although fingerprint biometric systems are considered more secure than traditional authentication methods, there are different threats that should be aware of. In this chapter, two main threats associated with fingerprint systems are introduced. In addition, hardware and software countermeasures have been presented.

4.1 THREATS

The two main threats that fingerprint systems face are spoofing and alternation.

4.1.1 Spoofing

Spoofing a biometric system means tricking the system by entering a fake fingerprint and therefore gets authenticated. Spoofing is mainly used to adopt another individual's identity. To spoof a biometric system, two operations should be done, namely capturing a fingerprint of a legitimate user, and creating an artificial fingerprint.

4.1.1.1 Capturing a Fingerprint

Capturing a legitimate user fingerprint can be done with or without genuine user collaboration. It may seem somewhat difficult without user's collaboration, but practically it is an easy process. The attacker can lift the user's fingerprint from a hard-smooth surface like glass or metal. For example, to capture fingerprints from a glass cup, it can be dusted with powder using a paintbrush. The powder will stick to the moisture of the fingerprint and thus will appear. Next, a picture of the fingerprint can be taken using a digital camera. The image will then be transmitted to the computer for further enhancement. The quality of the captured fingerprint depends on many factors, such as the nature and the smoothness of the surface that was being touched [19].

4.1.1.2 Creating an Artificial Fingerprint

With the collaboration of a genuine user, creating an artificial finger holding his fingerprint can be done using simple materials. Figure 6 shows the steps in creating such a finger [20]. Those fake fingers can fool the fingerprinting sensors with an average value of 80% [19].

Without user collaboration, creating a fake finger can be done using transparency and a photosensitive printed circuit board. First, the enhanced image is printed on a transparent medium. Next, the image is transferred from the transparent medium to a photosensitive printed circuit board. Finally, exposing the board to ultraviolet light results in a 3-D mold of the fingerprint [16].

Figure 6: Creating an Artificial Finger [20]

4.1.2 Alternations

Although fingerprint biometric systems are widely used as an identification method, fingerprints can be intentionally altered by hackers and criminals, which may fool fingerprint systems. Criminals alter their fingerprints to evade their identity. Different types of alternations can be done to change the structure of the ridge pattern, which are: obliteration, distortion, and imitation. In obliteration, the ridge pattern is being altered by burning, cutting, abrading, or applying strong chemicals. The area to be obliterated must be sufficiently large to fool fingerprint matchers. In distortion, fingerprints turned into unnatural ridge patterns by applying surgical procedures or skin grafting. However, fingerprint imitation can be done by transplanting a friction ridge skin from other parts of the body to the original finger in such a way that the altered fingerprint appears as a natural fingerprint pattern [21].

4.2 COUNTERMEASURES

One of the ways to overcome the spoofing vulnerability is to use a “liveness detection” method [19]. This method intends to detect whether the provided fingerprint is real or fake. The live human fingerprints have properties that differ from fake ones, such as thermal measurement and absorbance of light. Some biometric devices were already built such that they can sense the temperature of the finger. The ordinary epidermis temperature is between 26 to 30^{o}C at room temperature. The use of silicone faked fingers can reduce the temperature to a maximum of 2^{o}C [19]. This method helps to reduce spoofing and increases performance accuracy.

In this paper, deep learning techniques were used to detect if a fingerprint is real or altered and, if it is altered, to which alternation level it belongs. This can be achieved by first training a model with a large number of real and altered images. Then, testing the model accuracy in identifying real images from altered ones. Lastly, the model is ready to take in a new image and detect if it is real or altered. After that, transfer learning was employed on the pre-trained model to recognize real images from fake ones. Fake fingerprints were also classified as either with or without user collaboration.

CHAPTER 5: DATA SOURCE

The importance of classifying fingerprints into real, fake, or altered is rising every day. Forensic science is concerned with the body of knowledge and methods used to solve questions related to criminals and administrative law [5]. Criminals can defeat fingerprint biometric systems in some cases. They try to either spoof another individual's identity by creating an artificial fingerprint or alter their fingerprint so they will not be recognized. The main purpose of this work is to classify fingerprints to reduce spoofing and alternations by detecting whether the fingerprint is real, fake, or altered.

5.1 DATA OVERVIEW

Two datasets have been used to perform the classification task.

5.1.1 Sokoto Coventry Fingerprint Dataset (SOCOFing)

The SOCOFing dataset, which is available on Kaggle, is made up of 6,000 real images belonging to 600 African subjects of age 18 years or older. Ten fingerprints from each subject have been captured. Three different levels of alternations for obliteration, central rotation, and z-cut have been applied to get synthetically altered versions of the real fingerprints. STRANGE toolbox is a framework that is used to generate a synthetic alteration on the fingerprint images. Easy, medium and hard parameter settings in the STRANGE toolbox have been applied to the images to produce the alternations. A total number of 17,934 images have been generated with easy parameter settings, 17,067 images with medium parameter settings, and 14,272 images with hard parameter settings. Therefore, the SOCOFing dataset contains 55,273 images in total [22]. Figure 7 shows the distribution of different classes of the SOCOFing dataset.

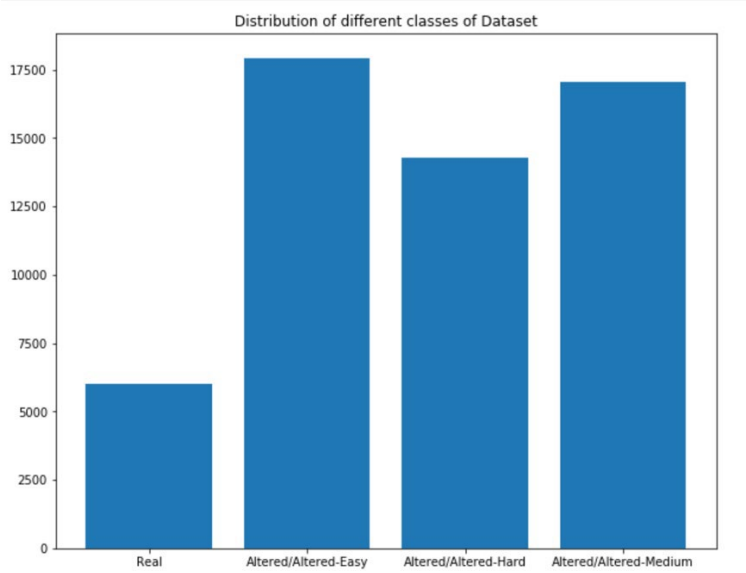

Figure 7: Distribution of Different Classes of SOCOFing Dataset

Each image has attributes such as gender, finger name, to which hand the finger belongs to and the type of alternation. All images are grayscale images. Figures 8, 9, 10, and 11 display some samples of real, easy altered, medium altered, and hard altered images respectively.

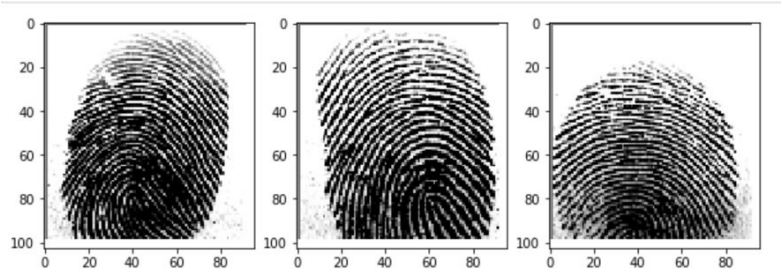

Figure 8: Real Fingerprint Samples

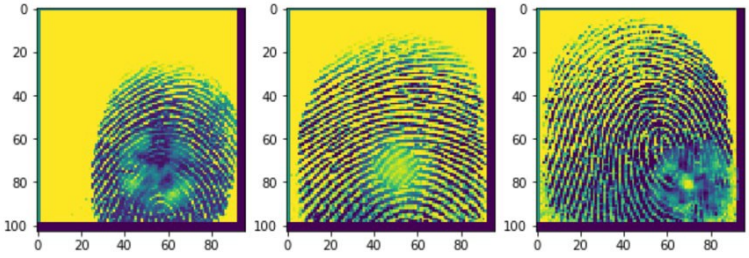

Figure 9: Altered-Easy Fingerprint Samples

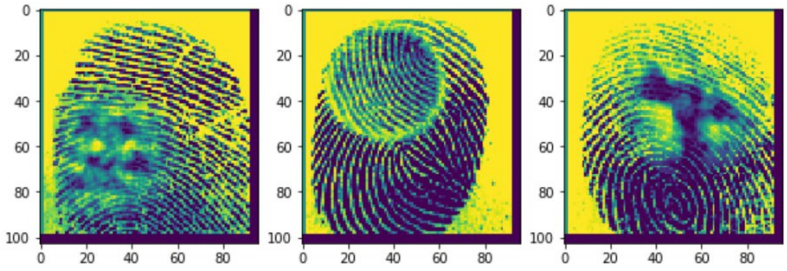

Figure 10: Altered-Medium Fingerprint Samples

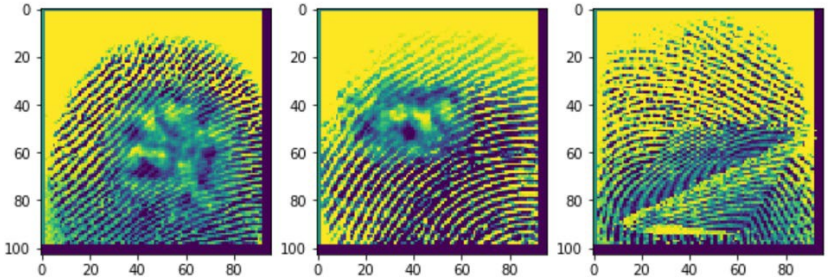

Figure 11: Altered-Hard Fingerprint Samples

5.1.2 ATVS-FakeFingerprint Database (ATVS-FFp DB) Version 1.0

ATVS-FFp database is made up of real and fake fingerprint images. It is divided into two subsets. One with the user cooperation (DATASET 1: DS_WithCooperation) and the other without the user cooperation (DATASET 2: DS_WithoutCooperation). In dataset 1, the user cooperated in generating the gummy fingers from which the fake fingerprint images were taken. It contains 816 real images and 816 fake images. The real samples have been taken from 17 users. Three different sensors have been used in capturing. In dataset 2, the user did not cooperate in generating the gummy fingers from which the fake fingerprint images were taken. It contains 768 real images and 768 fake images. The real samples have been taken from 16 users, and three different sensors have also been used in capturing. The whole dataset contains 3,168 images in total [23]. Figure 12

shows the distribution of different classes of the ATVS-FFp dataset. Figures 13, 14, and 15 display samples of real, fake with user cooperation, fake without user cooperation images respectively.

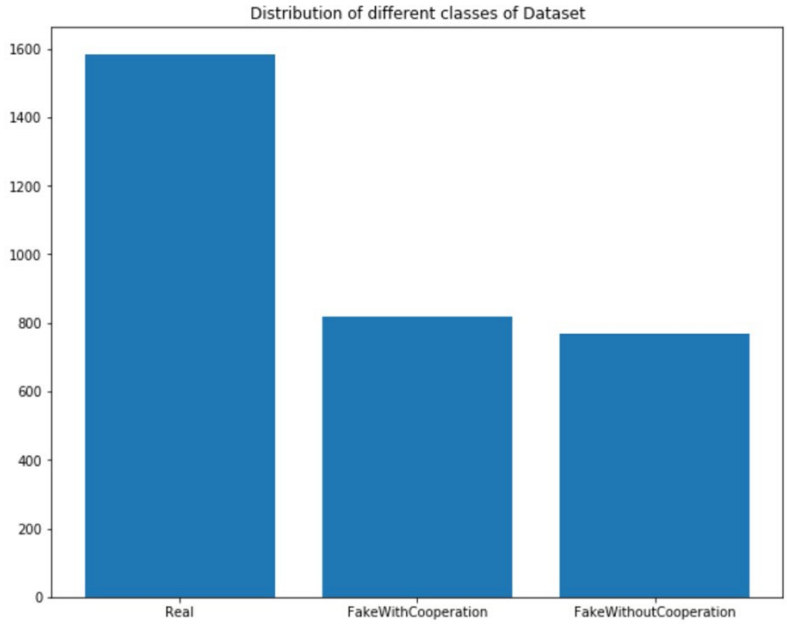

Figure 12: Distribution of Different Classes of ATVS-FFp Dataset.

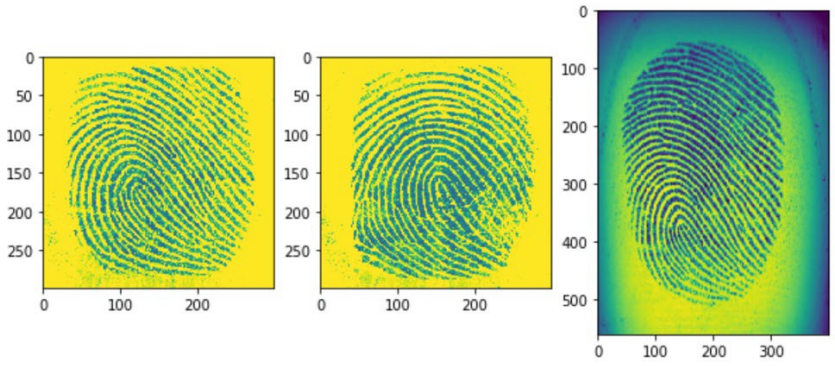

Figure 13: Real Fingerprint Samples

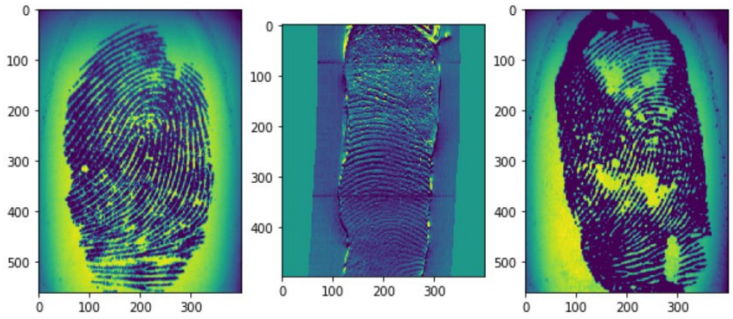

Figure 14: Samples of Fake Images with User Cooperation

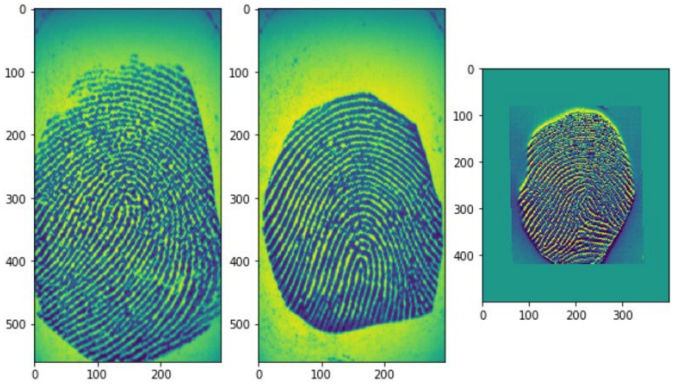

Figure 15: Samples of Fake Images without User Cooperation

5.2 DATA PREPROCESSING

The data preprocessing step is essential before feeding the data into the model. The model tends to learn better if the quality of the images is higher. The purer the data is, the better the model learns. In this project, different data preprocessing techniques were applied to produce better-quality data.

First, all the images in both datasets have been converted to grayscale images and have been resized into 100 * 100 (width * height). Smaller-sized images result in a faster training process. Then, all the data have been shuffled to avoid any chance of overfitting. Lastly, to get unique data for both training and testing sets, 10% of splitting has been applied. Figures 16 and 17 show SOCOFing and ATVS-FFp datasets after being divided into training and test sets respectively.

```
import torch

X= torch.Tensor([i[0] for i in training_data]).view(-1,100,100)
X= X/255.0
y= torch.Tensor([i[1] for i in training_data])
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('Training set:' , len(train_X))
print('Test set:' , len(test_X))
```

```
Training set: 49743
Test set: 5527
```

Figure 16: SOCOFing Dataset After Being Divided into Training and Testing Sets

```
import torch

X= torch.Tensor([i[0] for i in training_data]).view(-1,100,100)
X= X/255.0
y= torch.Tensor([i[1] for i in training_data])
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('Training set:' , len(train_X))
print('Test set:' , len(test_X))
```

```
Training set: 2852
Test set: 316
```

Figure 17: ATVS-FFp Dataset After Being Divided into Training and Testing Sets

The SOCOFing dataset was used to train the model. Then, by using the transfer learning technique, the SOCOFing trained model was fine-tuned and re-trained to deal with the ATVS-FFp dataset.

CHAPTER 6: METHODOLOGY AND EXPERIMENTAL SETUP

This chapter presents the proposed Convolutional Neural Network (CNN) that was used to perform the classification task. The CNN proved to be efficient in image processing-related tasks and therefore is suitable for classifying fingerprint images into real and fake.

6.1 SOCOFing PROPOSED MODEL

Since the SOCOFing dataset has a large number of samples, it has been used to train the model. Then the pre-trained model was used to apply transfer learning to train and test the ATVS-FFp dataset.

6.1.1 Convolutional Neural Network (CNN)

The CNN is made up of multiple consecutive layers. Each layer is formed of a set of artificial neurons. A neuron is a function that takes multiple inputs, calculates the inputs' weighted sum, and outputs an activation value [24].

6.1.1.1 How Does CNN Work?

The CNN can be divided into two main parts:

- The hidden layers

  The CNN first puts the input image into a series of hidden layers. The hidden layers are responsible for extracting the features. The layers are organized into 3 dimensions: height, width, and depth. The neurons in one layer do not connect to all the neurons of the previous layer. The network performs a series of convolution and pooling operations which result in detecting the features. Convolution is referred to as a mathematical combination of two functions to produce a third function. Convolution is usually associated with several attributes, which are:

1. Kernel size: Kernel is a filter that slides over the input. At each location, matrix multiplication is made, and the result is being summed onto the feature map. Common sizes for a kernel are 3x3 and 5x5.
2. Stride: Stride is a value that determines the step the convolution filter moves each time. If stride is equal to one, the filter moves pixel by pixel on the input. By increasing the stride, fewer overlap chances between cells may occur.
3. Padding: Since the feature map that is being generated after each layer is less in size than the input, padding can be performed by adding zeros to the input frame of the matrix. Padding helps in preventing the feature map from shrinking.

The deeper the layer is, the more detailed features are extracted. After each convolution layer, an activation function is applied to produce a non-linear output. Then, pooling is performed to reduce dimensionality and thus reduce computations in the network. Pooling leads to faster training time and controls overfitting. Max pooling is a common pooling function that takes the maximum value of each window. This helps in decreasing the feature map size but keeps the significant information. The final output of all the convolutional layers should be flattened to a single vector [25].

- The fully connected layers

The fully connected layers work as a classifier. They consist of few connected layers where the neurons in one layer are connected to all the neurons in the previous layer. It gets the extracted features, one-dimensional data, as an input. Based on the activation map of the final convolution layer, the output of the classification layer is a set of values that indicate how likely the image belongs to a class [24].

6.1.1.2 CNN in the Proposed Model

The proposed CNN model for the SOCOFing dataset has four convolutional layers. All the layers have a kernel size of 3x3 and use a stride of one. No padding has been added. The output of every convolutional layer is shaped by the Rectifier Linear Model (ReLU) function. Max pooling is applied for the four layers with a size of 2x2. The convolutional layers are followed by three fully connected layers. A SoftMax activation function is then applied. The SoftMax function is used to map the output of the network to a probability distribution. Four probability values will be generated associated with each class. The biggest probability indicates to which class the image most likely belongs. Figures 18 and 19 illustrate the feature extraction step and the classification step in the proposed model, respectively.

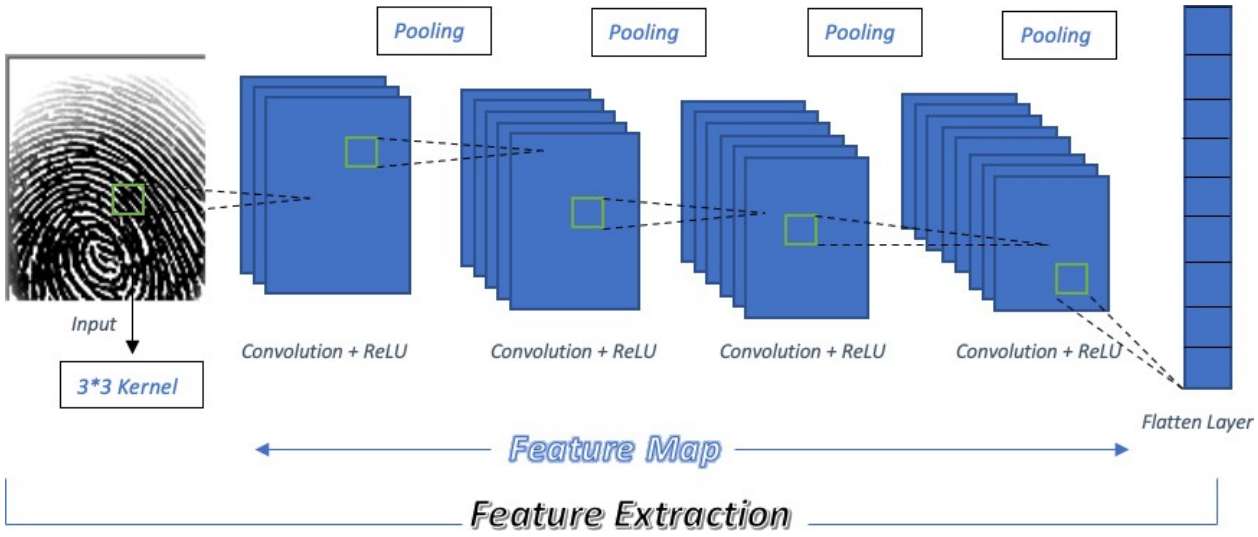

Figure 18: Feature Extraction in the Proposed Model

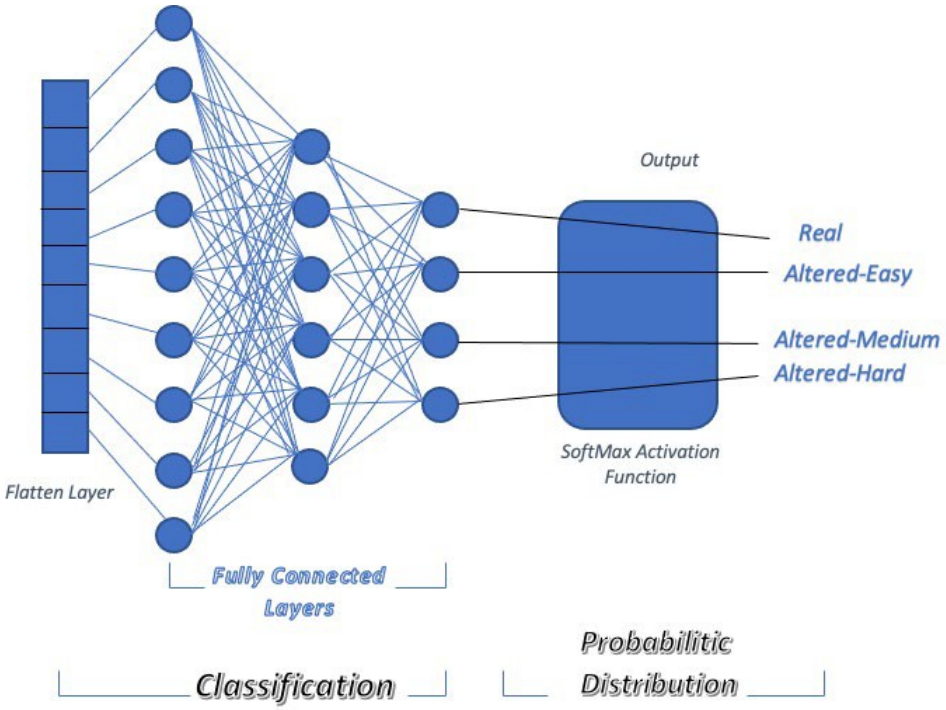

Figure 19: Classification in the Proposed Model

6.1.2 Training Process

The main purpose of training a CNN is to adjust the weights of the individual neurons so the CNN can generate better classification results.

6.1.2.1 Training Phases

A training process is composed of two main phases:

- A forward phase: In this phase, the input is being passed completely through the network. In the beginning, all the weights are randomly chosen.
- A backward phase: When the output is being generated from the previous phase, it is compared with the original label of the input. If they are mismatched, the loss is being computed using a loss function and backpropagated to adjust the weights of the neurons. Adjusting the weights will help the network to better classify the image when it is passed again through the network. An optimizer is used in this phase to optimize the tuning

process, which helps determine the weights to be adjusted instead of making random corrections [24].

Every run of the entire dataset is an epoch. The CNN needs to go through several epochs during training and adjusts the weights accordingly. After each epoch, the network becomes closer to the right classification. When the network improves, the number of adjustments is lessened. After several epochs, the network performs most efficiently even when the number of epochs increases.

6.1.2.2 Training the Proposed Model

The training set in the SOCOFing dataset, which consists of 49743 samples, has been divided into batches. A batch size refers to the number of images to run through before adjusting the weights of the neurons [26]. Splitting the dataset into batches leads to a faster training process since the weights are getting updated after each propagation. The proposed model has a batch size of 100.

The loss function used in the proposed model is MSELoss, an abbreviation for Mean Squared Error Loss. As the name indicates, MSELoss measures the mean squared error between the value returned by the model f and the actual value y as shown in Equation 1 where N is the number of data points.

$$MMMMMM = \frac{1}{NN} \sum_{ii=1}^{NN} (ffff - yyff)^2$$

Equation 1: MSELoss

Adam is the replacement optimization algorithm that has been used to update the weights of the individual neurons. It can handle sparse gradients on noisy problems. The learning rate has been set to 0.0001. The learning rate defines the step size at each iteration while moving toward a minimum of a loss function [27]. Figure 20 displays the training process results. Figure 20 shows that as the number of epochs increases, the loss value gets decreased.

```
100%|██████████| 498/498 [10:52<00:00,  1.31s/it]
  0%|          | 0/498 [00:00<?, ?it/s]
Epoch: 9. Loss: 0.11131811141967773
100%|██████████| 498/498 [10:57<00:00,  1.32s/it]
  0%|          | 0/498 [00:00<?, ?it/s]
Epoch: 10. Loss: 0.10231547802686691
100%|██████████| 498/498 [11:02<00:00,  1.33s/it]
  0%|          | 0/498 [00:00<?, ?it/s]
Epoch: 11. Loss: 0.09103181213140488
100%|██████████| 498/498 [11:00<00:00,  1.33s/it]
  0%|          | 0/498 [00:00<?, ?it/s]
Epoch: 12. Loss: 0.08432624489068985
```

Figure 20: SOCOFing Training Process Results

6.1.3 Testing

After training the model, it should be tested to evaluate its accuracy. Accuracy results indicate how the model is performed on unseen data. Testing is done by comparing the predicted class with the actual class the image belongs to. If the testing accuracy is less than the training accuracy, that is an indication that the model is overfitted. SOCOFing model has been tested on the testing dataset, which consists of 5527 samples. An accuracy result of 81% was acquired. Figure 21 displays the testing accuracy results.

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

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

```
100%|██████████| 5527/5527 [01:05<00:00, 84.22it/s]
Accuracy:  0.81
```

Figure 21: SOCOFing Testing Accuracy Results

6.2 TRANSFER LEARNING

Transfer learning is very common in deep learning and therefore, CNNs can be trained with relatively little data. This is very helpful in the data science field since not all real-world problems have a huge number of labeled data [28].

6.2.1 Overview

Transfer learning is a technique that uses a pre-trained model on a different but related task. The knowledge that a model has learned from a task with a lot of available data is applied to improve generalization in another task that does not have much data. In general, neural networks usually detect edges in the first layers, shapes in the middle layer, and specific features related to the task in the latter layers. In transfer learning, the first and middle layers are used, and the latter layers are re-trained to avoid overfitting. Transfer learning reduces the training time since sometimes it takes a long time to train a model from scratch. It also leads to better performance and does not need large datasets [28].

6.2.2 Fine-tune and Re-train SOCOFing Model

In this project, the SOCOFing model has been fine-tuned to take in the ATVS-FFp dataset, consisting of 3,168 samples. As a result, SOCOFing has 4 labeled classes while ATVS-FFp has 3 labeled classes which are: real, fake with user cooperation, and fake without user cooperation.

The SOCOFing model has been transferred to two new models. The first model was proposedto classify real images from fake ones without being specific. Real images were given a label (0) and the two fake classes were combined with a label (1). Therefore, the last fully connected layerof the SOCOFing model has been adjusted to 2 outputs instead of 4.

The second model was proposed to classify the images into 3 classes. This model was more specific and was able to tell if the fake images were done with or without the user cooperation. Real images were given a label (0), fake images with user cooperation were given a label (1), and fake images without user cooperation were given a label (2). Therefore, the last fully connected layer has been adjusted to 3 outputs instead of 4. All the other layers were kept fixed.

Both models have been trained using the MSELoss function and Adam optimizer with a 0.0001 learning rate. The ATVS-FFp training set, which consists of 2,852 images,has been passed to train the new models. The training set has been divided into batcheswith a size of 100 and the epochs were set to 30.

. Both models have been tested to verify their accuracy. The ATVS-FFp testing set, which consists of 316 samples, has been fed into the models. The first model was able to achieve a classification accuracy of 99.4% while the second model achieved a 97.5% accuracy result. Figure 22 illustrates the second model after applying the transfer learning technique. Figures 23 and 24 display the testing accuracy results for model 1 and model 2 respectively.

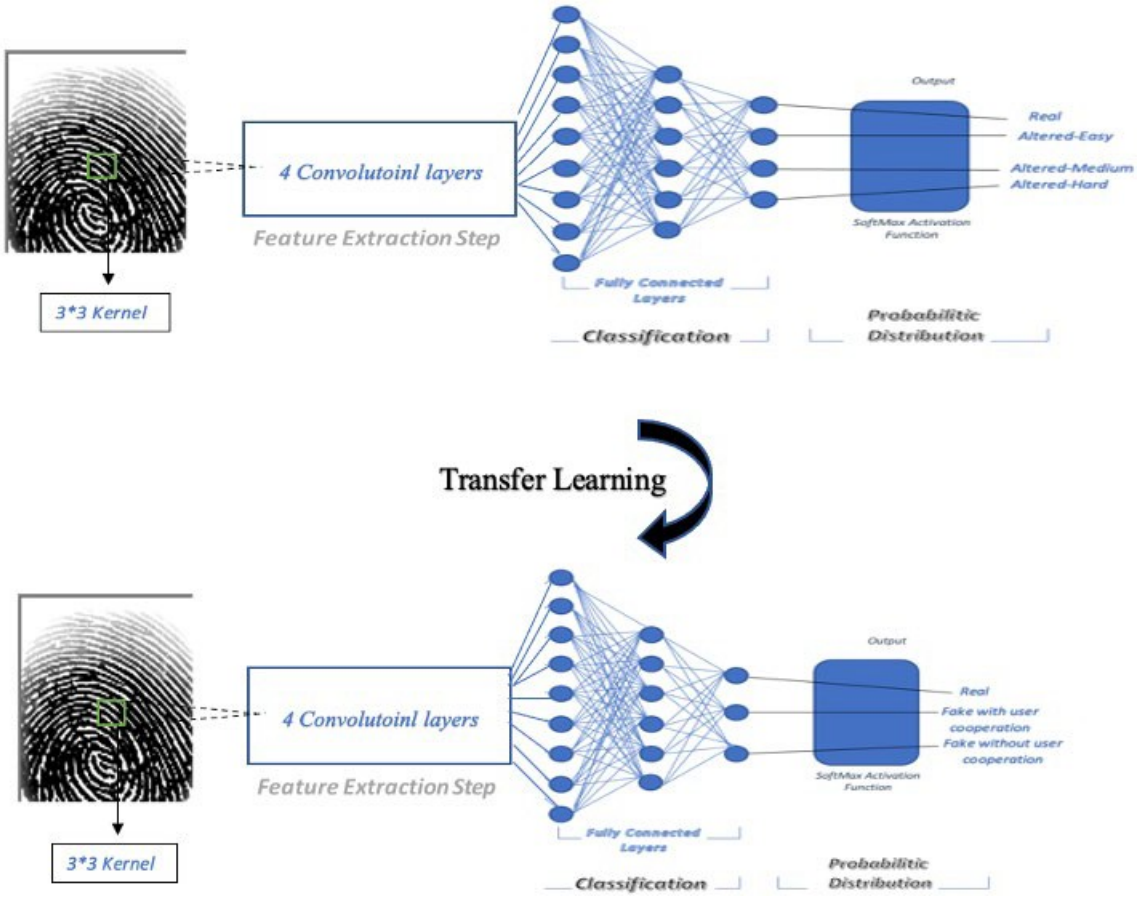

Figure 22: Model 2 After Applying Transfer Learning

```
correct = 0
total = 0
with torch.no_grad():
    for i in tqdm(range(len(test_X))):
        actual_class = torch.argmax(test_y[i])
        model_out = model1(test_X[i].view(-1, 1, 100, 100))[0]
        predicted_class = torch.argmax(model_out)

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

```
100%|██████████| 316/316 [00:03<00:00, 88.39it/s]
Accuracy:  0.994
```

Figure 23: Model 1 Accuracy Results

```
correct2 = 0
total2 = 0
with torch.no_grad():
    for i in tqdm(range(len(test_X2))):
        actual_class = torch.argmax(test_y2[i])
        model_out = model2(test_X2[i].view(-1, 1, 100, 100))[0]
        predicted_class = torch.argmax(model_out)

        if predicted_class == actual_class:
            correct2 += 1
        total2 += 1
print("Accuracy: ", round(correct2/total2, 3))
```

```
  0%|          | 0/316 [00:00<?, ?it/s]
  4%|▍         | 12/316 [00:00<00:02, 113.80it/s]
  9%|▊         | 27/316 [00:00<00:02, 120.37it/s]
 13%|█▏        | 40/316 [00:00<00:02, 122.30it/s]
 17%|█▋        | 53/316 [00:00<00:02, 123.66it/s]
 21%|██        | 66/316 [00:00<00:01, 125.14it/s]
 25%|██▌       | 79/316 [00:00<00:01, 126.55it/s]
 30%|██▉       | 94/316 [00:00<00:01, 127.79it/s]
 34%|███▎      | 106/316 [00:00<00:01, 121.50it/s]
 38%|███▊      | 121/316 [00:00<00:01, 127.77it/s]
 43%|████▎     | 136/316 [00:01<00:01, 131.62it/s]
 48%|████▊     | 151/316 [00:01<00:01, 134.92it/s]
 53%|█████▎    | 166/316 [00:01<00:01, 138.57it/s]
 57%|█████▋    | 180/316 [00:01<00:00, 138.83it/s]
 62%|██████▏   | 195/316 [00:01<00:00, 141.07it/s]
 66%|██████▋   | 210/316 [00:01<00:00, 143.10it/s]
 71%|███████   | 225/316 [00:01<00:00, 141.26it/s]
 76%|███████▌  | 240/316 [00:01<00:00, 139.11it/s]
 80%|████████  | 254/316 [00:01<00:00, 135.78it/s]
 85%|████████▍ | 268/316 [00:02<00:00, 132.61it/s]
 89%|████████▉ | 282/316 [00:02<00:00, 130.00it/s]
 94%|█████████▍| 296/316 [00:02<00:00, 129.97it/s]
100%|██████████| 316/316 [00:02<00:00, 133.67it/s]
Accuracy:  0.975
```

Figure 24: Model 2 Accuracy Results

CHAPTER 7: OBSERVATIONS AND RESULTS

The importance of classifying fingerprints increases every day. Fingerprint plays a fundamental role in identifying criminals. It is also used as an authentication method in almost every sector. This chapter discusses the results that have been gained and the observations that have been noticed throughout the study.

7.1 SOCOFing MODEL RESULTS

In this project, the SOCOFing dataset has been used to classify fingerprints' images under four categories which are: real, altered-easy, altered-medium, and altered-hard. Each category was given a label from 0-3, respectively. SOCOFing has been split into train and test datasets by a ratio of 9:1. Figure 25 summarizes all the steps that have been implemented throughout the project.

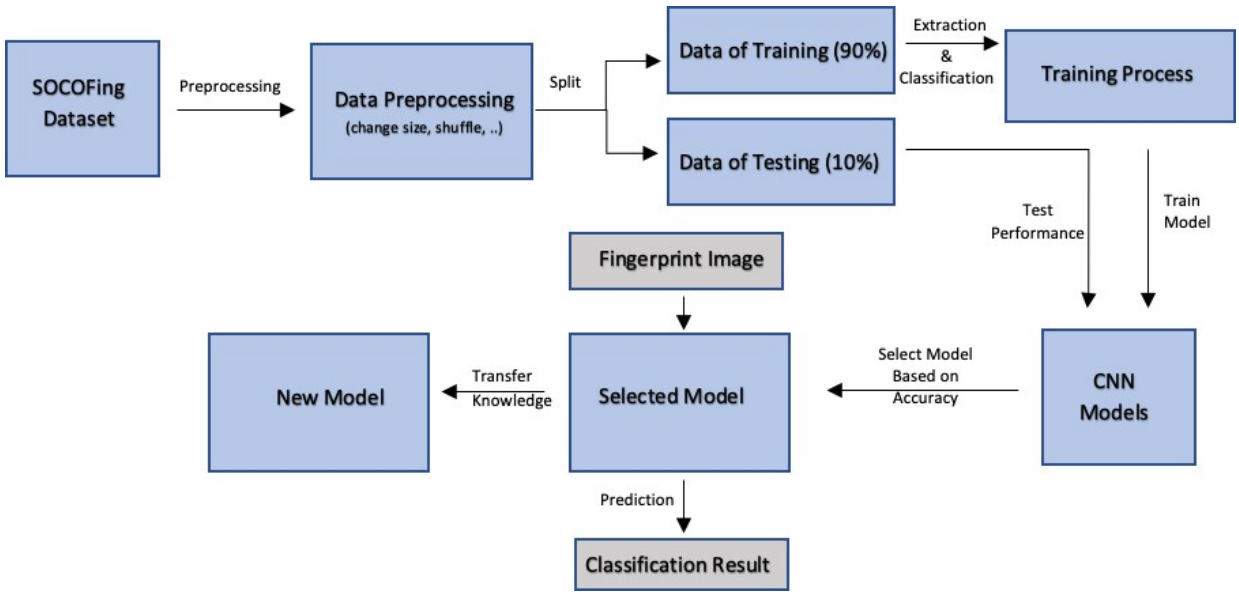

Figure 25: SOCOFing Model Structure

SOCOFing model has been run through 30 epochs. As the number of epochs was increasing, the testing accuracy was also improving. However, after epoch number 23, the network "converges", which means it becomes as good as it can be. An 81% testing accuracy result was

achieved. Figures 26 and 27 illustrate the SOCOFing model training vs testing accuracy and the SOCOFing model training vs testing loss respectively.

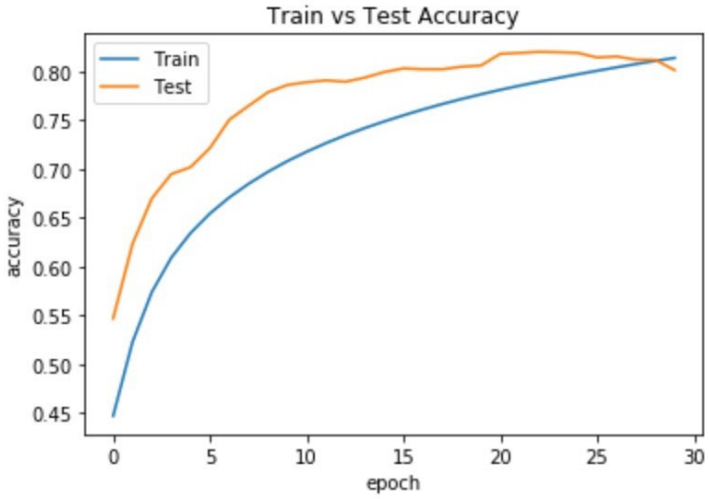

Figure 26: SOCOFing Model Training vs Testing Accuracy

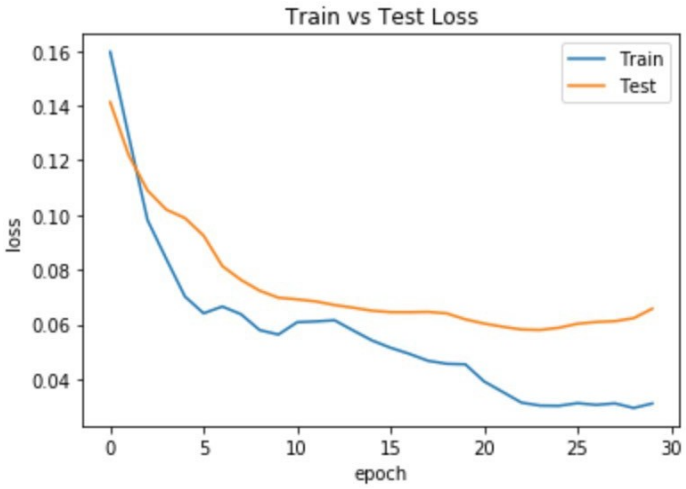

Figure 27: SOCOFing Model Training vs Testing Loss

The knowledge the SOCOFing model learned has been transferred to another two models that will classify a different but similar task. The new two models were trained using the ATVS-FFp dataset. ATVS-FFp contains 3,168 fingerprint images in total. Since it is a small dataset, it is

better to classify the images by transferring a pre-trained model to avoid overfitting and get better accuracy results.

7.2 MODEL 1 RESULTS AFTER APPLYING TRANSFER LEARNING

In the first model, the task is to classify images under two categories which are: real and fake. The real category was given a label (0) and the fake category was given a label (1). ATVS-FFp has been split into a training set and a test set by a ratio of 9:1. The model achieved a testing accuracy of 99.4%. Figures 28 and 29 illustrate model 1 training vs testing accuracy and model 1 training vs testing loss respectively. Figure 28 shows that the model was able to achieve high accuracy results after epoch number 3 and therefore, the advantage of using transfer learning appears.

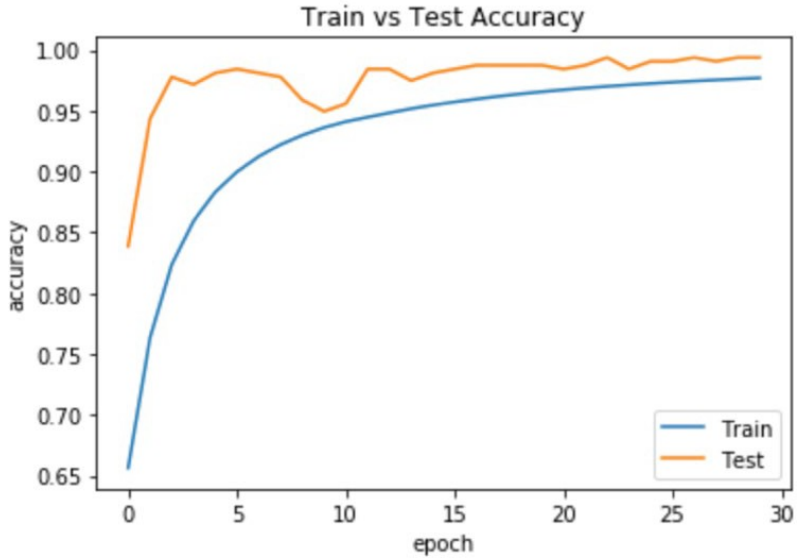

Figure 28: Model1 Training vs Testing Accuracy

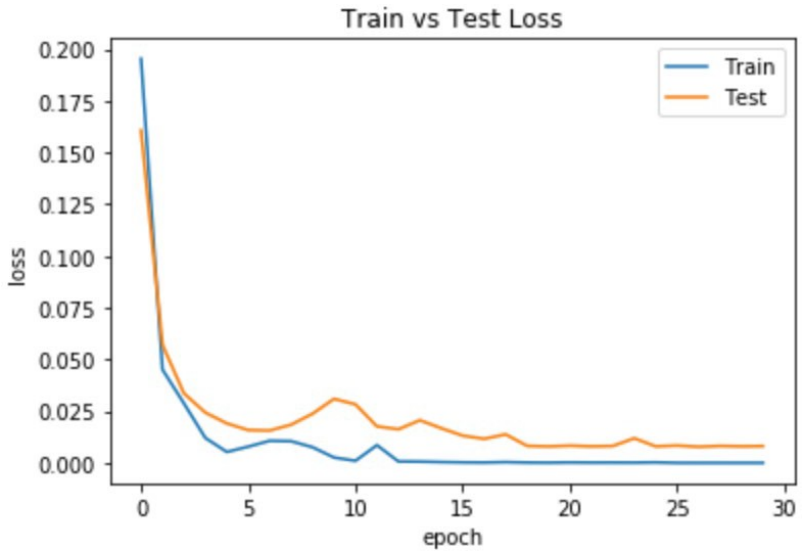

Figure 29: Model1 Training and Testing Loss

7.3 MODEL 2 RESULTS AFTER APPLYING TRANSFER LEARNING

In the second model, the task is to be more specific in classification. Fingerprint images have been classified under three categories which are: real, fake with user cooperation, and fake without user cooperation. Each category was given a label from 0-2, respectively. The model achieved a testing accuracy of 97.5%. Figures 30 and 31 illustrate model 2 training vs testing accuracy and model 2 training vs testing loss, respectively. Similarly, Figure 29 shows that the model was able to achieve high accuracy results at epoch number 3.

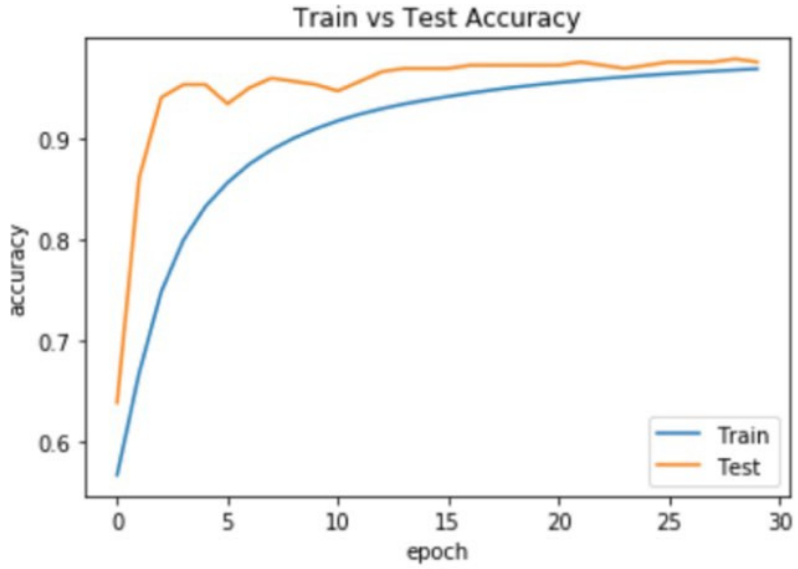

Figure 30: Model2 Training vs Testing Accuracy

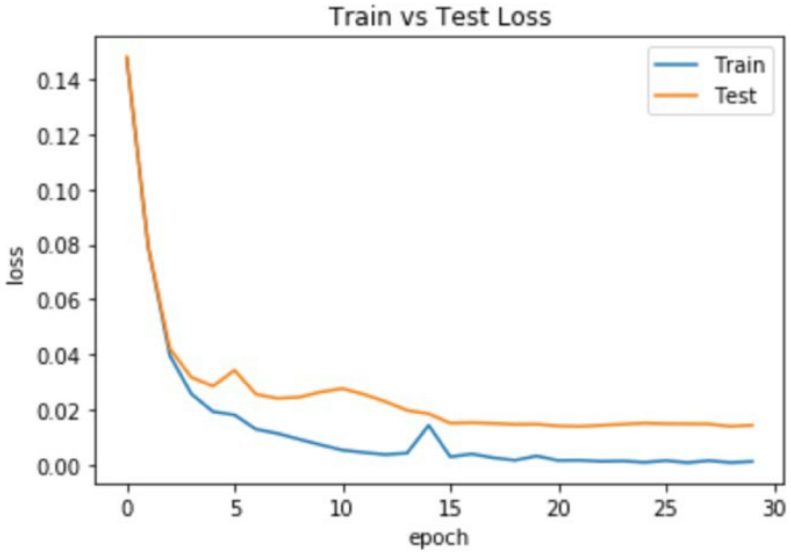

Figure 31: Model2 Training and Testing Loss

7.4 DISCUSSION

Table 1 displays the training loss for the SOCOFing model and the other two models after applying transfer learning for the first 5 epochs.

Epoch Number	SOCOFing Model	New Model 1	New Model 2
1	0.1597	0.1953	0.1476
2	0.1286	0.0451	0.0788
3	0.0982	0.0287	0.0396
4	0.0840	0.0120	0.0256
5	0.0702	0.0053	0.0192

Table 1: Training Loss Results

Table 1 shows that SOCOFing model training loss was large at the end of the first epoch. The model started with random neurons' weights in the first epoch and that justifies why the loss amount was large. After the first epoch, the loss has been backpropagated, and the weights of the neurons have been adjusted. Adjusting the weights helped the network to better classify the images when they were passed again through the network, and therefore, the training loss decreased by an amount of approximately 0.3 at the end of epoch 2. After adjusting the weights multiple times, the training loss continued to decrease by a smaller amount since the network had better classification results.

After applying transfer learning, the new models also started with a large loss amount because the last fully connected layer was still not trained. At the end of epoch two, the loss amount was significantly decreased since the first and middle layers were already trained, and the neurons'

weights of the last layer have been adjusted. At epoch five, SOCOFing training loss was still large while the other two models were getting very good classification results.

7.5 FEEDING IN REAL-WORLD IMAGES

Model 2 has been tested on real-world fingerprint images to evaluate its performance. First, an inked fingertip was pressed on a white paper sheet and then pictured to get the digital format. Figure 32 shows the real fingerprint and the predicted class by the model. The model successfully gave a label (0) to the image and classified it as a real image.

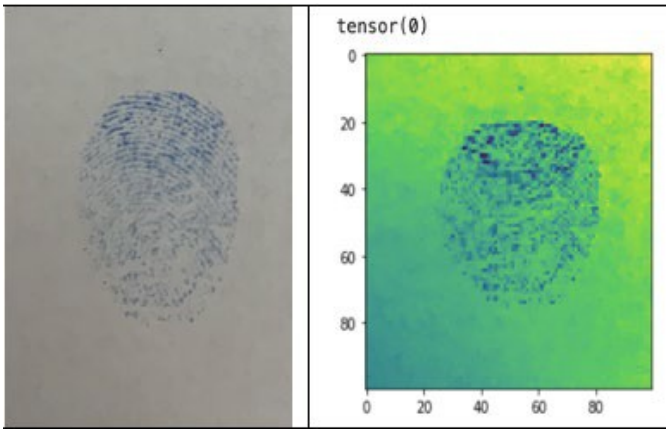

Figure 32: An Example of a Real Fingerprint

Second, a fingertip was pressed on a playdoh mold and dusted with powder to clarify the pattern of the ridges. A picture was taken to get the digital format of the fingertip. Figure 33 shows the fake fingerprint and the predicted class by the model. The model successfully gave a label (1) to the image and classified it as a fake image with the user cooperation.

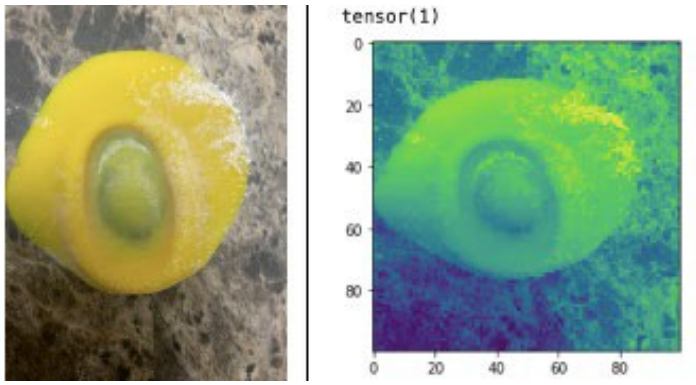

Figure 33: An Example of a Fake Fingerprint with User Cooperation

Lastly, a fake fingerprint without user cooperation has been fed into the model. Figure 34 shows the fake fingerprint and the predicted class by the model. The model successfully gave a label (2) to the image and classified it as a fake image without user cooperation.

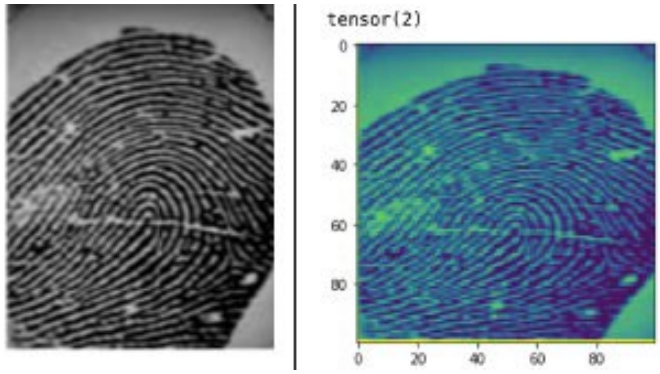

Figure 34: An Example of a Fake Fingerprint without User Cooperation

CONCLUSION AND FUTURE WORK

The importance of classifying fingerprints increases every day. Fingerprint biometric systems face several threats that may lead to incorrect identification or authentication. Criminals tend to alter their fingerprints by making scars or surgical procedures to evade their identity. Hackers also try to adopt another individual's identity by creating a fake fingerprint. This study proposed a CNN model that classifies fingerprint images under four categories. It classifies fingerprints into real or altered and determines the level of alternation if it is altered. Sokoto Coventry Fingerprint Dataset (SOCOFing) was used to train and test the model. Classification accuracy of 81% was acquired. The transfer learning technique was also applied to the proposed model. Consequently, two new models were developed and trained using ATVS-FakeFingerprint Database (ATVS-FFp DB). The first model achieved a testing accuracy of 99.4% in classifying fingerprint images into real and fake. The second model was able to classify images into real and fake and determines if the fake images were generated with or without the user cooperation. An accuracy result of 97.5% was gained.

Improving the performance accuracy of the SOCOFing model will be left for future work. Considering hierarchical classification may lead to better accuracy results.

BIBLIOGRAPHY

[1] K. Zurkus, "Google Survey Finds Two in Three Users Reuse Passwords," Infosecurity Magazine, 5 Feb 2019. [Online]. Available: https://www.infosecurity-magazine.com/news/google-survey-finds-two-users/.

[2] D. Vergara, "Static vs Behavioral : What's the Future of Biometric Authentication?," ITProPortal, 3 Jan 2019. [Online]. Available: www.itproportal.com/features/static-vs-behavioural-whats-the-future-of-biometric-authentication/..

[3] P. Pete, "Fingerprint Readers: History's Oldest Biometric," Penguin Pete's Greatest Hits, 20 July 2020. [Online]. Available: www.penguinpetes.com/wordpress/2020/07/20/fingerprint-readers-historys-oldest-biometric/.

[4] A. A. Abood, G. Sulong and S. U.Peters, "A Review of Fingerprint Image Pre-processing," ResearchGate, June 2014. [Online]. Available: www.penguinpetes.com/wordpress/2020/07/20/fingerprint-readers-historys-oldest-biometric/.

[5] Y. Shehu, A. Ruiz-Garcia, A. E. James and V. Parade, "Detection of Fingerprint Alternations Using Deep Convolutional Neural Networks," ResearchGate, September 2018. [Online]. Available:

https://www.researchgate.net/publication/327895763_Detection_of_Fingerprint_Alterations_Using_Deep_Convolutional_Neural_Networks.

[6] S. Yoon, J. Feng and A. K. Jain, "Altered fingerprints: analysis and detection," IEEE Trans PAttern Anal Mach Intell, March 2012. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/21808092/.

[7] D. M. Uliyan, S. Sadeghi and H. A. Jalab, "Anti-spoofing method for fingerprint recognition using patch-based deep learning machine," ScienceDirect, April 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215098619300527.

[8] M. M. Ali, V. H. Mahale, P. Yannawar and A. Gaikwad, "Overview of Fingerprint Recognition System," research gate,, March 2016. [Online]. Available: https://www.researchgate.net/publication/310953762_Overview_of_Fingerprint_Recognition_System.

[9] "Meaning of Fingerprints," We Must Know, Mar 27. [Online]. Available: www.wemustknow.wordpress.com/2010/03/27/meaning-of-fingerprints/.

[10] C. C. Ho and C. Eswaran, "Consodilation of fingerprint databases: A Malaysian case study," ResearchGate, December 2011. [Online]. Available: https://www.researchgate.net/publication/220981197_Consodilation_of_fingerprint_databases_A_Malaysian_case_study.

[11] F. Yahya, K. Kadir, H. Nasir, and S. I. Safie, "Fingerprint Biometric Systems," ResearchGate, 2016 September. [Online]. Available: https://www.researchgate.net/publication/309961589_Fingerprint_Biometric_Systems.

[12] "BIOMETRIC TECHNOLOGIES," 2017. [Online]. Available: https://www.fingerprints.com/asset/assets/downloads/fingerprints-biometric-technologies-whitepaper-2017-revb.pdf.

[13] R. Subban and D. Mankame, "A Study of Biometric Approach Using Fingerprint Recognition," ResearchGate, January 2013. [Online]. Available: https://www.researchgate.net/publication/271293116_A_Study_of_Biometric_Approach_Using_Fingerprint_Recognition.

[14] P. Sureshbabu and M. Sakthivadivu, "A Review on Biometrics Authentication System Using Fingerprint," The Research Publication, February 2019. [Online]. Available: www.trp.org.in/issues/a-review-on-biometrics-authentication-system-using-fingerprint..

[15] F. Romulo, L. Carneiro, J. Bessa and J. L. De Moraes, "Techniques of Binarization, Thinning and Feature Extraction Applied to a Fingerprint System," ResearchGate, 2014 October. [Online]. Available: https://www.researchgate.net/publication/267325536_Techniques_of_Binarization_Thinning_and_Feature_Extraction_Applied_to_a_Fingerprint_System.

[16] "Morpho Places First in NIST 2014 MINEX Fingerprint Benchmark," yahoo!finance, 17 November 2014. [Online]. Available: https://finance.yahoo.com/news/morpho-places-first-nist-2014-144000749.html.

[17] H. Le Hong, H. N. Nguyen and T.-T. Nguyen, "A Complete Fingerprint Matching Algorithm on GPU for a Large Scale Identification System," ResearchGate, January 2016. [Online]. Available: https://www.researchgate.net/publication/301253150_A_Complete_Fingerprint_Matching_Algorithm_on_GPU_for_a_Large_Scale_Identification_System.

[18] J. Yang, "Non-minutiae Based Fingerprint Descriptor," IntechOpen, 20 June 2011. [Online]. Available: https://www.intechopen.com/books/biometrics/non-minutiae-based-fingerprint-descriptor.

[19] I. Iqbal, "Biometrics: Security Issues and Countermeasures," Semantic scholar, 2015. [Online]. Available: https://www.semanticscholar.org/paper/Biometrics%3A-Security-Issues-and-Countermeasures-Iqbal/b5ff21d17ffedaf871615d675aeea2174ae2c7be?p2df.

[20] S. Chang, K. Larin, Y. Mao, W. Almuhtadi and C. Flueraru, "Fingerprint Spoof Detection By NIR Optical Analysis," IntechOpen, 27 July 2011. [Online]. Available: https://www.intechopen.com/books/state-of-the-art-in-biometrics/fingerprint-spoof-detection-by-nir-optical-analysis.

[21] J. Feng, A. K. Jain and A. Ross, "Detecting Altered Fingerprints," ResearchGate, August 2010. [Online]. Available: https://www.researchgate.net/publication/220929010_Detecting_Altered_Fingerprints.

[22] Y. Shehu, A. E. James, A. Ruiz-Garcia and V. Palade, "Sokoto Coventry Fingerprint Dataset," ResearchGate, July 2018. [Online]. Available: https://www.researchgate.net/publication/326681401_Sokoto_Coventry_Fingerprint_Dataset.

[23] "ATVS-FakeFingerprint Database (ATVS-FFp DB) Version 1.0," Ideal Biometrics Test, [Online]. Available: http://biometrics.idealtest.org/dbDetailForUser.do?id=11.

[24] B. Dickson, "What are convolutional neural networks (CNN)?," TechTalks, 6 January 2020. [Online]. Available: https://bdtechtalks.com/2020/01/06/convolutional-neural-networks-cnn-convnets/.

[25] D. Cornelisse, "An intuitive guide to Convolutional Neural Networks," freecodecamp, 24 April 2018. [Online]. Available: https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/.

[26] J. Brownlee, "Difference Between a Batch and an Epoch in a Neural Network," Machine Learning Mastery, 20 July 2018. [Online]. Available: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/.

[27] J. Brownlee, "How to Configure the Learning Rate When Training Deep Learning Neural Networks," Machine Learning Mastery, 23 January 2019. [Online]. Available: https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/.

[28] N. Donges, "WHAT IS TRANSFER LEARNING? EXPLORING THE POPULAR DEEP LEARNING APPROACH," built in, 16 June 2019. [Online]. Available: https://builtin.com/data-science/transfer-learning.