ROBUSTNESS OF FINGERPRINT VERIFICATION ALGORITHMS AGAINST SYNTHETIC DEFORMATIONS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Sinem CANTÜRK

October 2019
İZMİR

ACKNOWLEDGMENTS

I would first like to thank my supervisors Asst. Prof. Nesli ERDOĞMUŞ of the faculty of Computer Engineering at İzmir Institute of Technology and Prof. Dr. Holger FRÖNING at Heidelberg University for the invaluable advice and positive encouragement they have provided me throughout the course of this project.

I am highly indebted to all my fellows for making these three years of education priceless. Without their ambitious participation and guidance, this accomplishment would not have been possible.

Finally, I would like to express my very profound gratitude to my parents for providing me with continuous love and support and for their kind co-operation and encouragement in completion of this assignment.

ABSTRACT

ROBUSTNESS OF FINGERPRINT VERIFICATION ALGORITHMS AGAINST SYNTHETIC DEFORMATIONS

Fingerprint recognition is one of the biometric techniques used for the identification of humans. The developments and research about fingerprint recognition to date are of great importance in advancing fingerprint recognition and verification scenarios. The fact that fingerprint recognition systems are used almost everywhere and are easily accessible is directly proportionate to a large amount of research in these areas.

During the acquisition of the fingerprint, there are many environmental factors that may affect the quality of the print and eventually, its ability to be recognized. For a fingerprint recognition algorithms, it is important to handle the difficulties that arise due to those variations.

The aim of the thesis is to obtain and compare the results of not only an existing feature-based fingerprint recognition techniques but a fingerprint recognition technique that uses deep learning. The main focus is on how fingerprint verification algorithms behave under the circumstances of synthetically distorted fingerprint images. After developing two different verification systems, the goal is to compare system results with and without distorted images. The results of the two methods with and without externally added deformations effect on the fingerprint image is compared. The first system has a feature-based approach comparing the images via local features on the fingerprint. In order to do this two different descriptors that are called ORB and SIFT are used. In the feature-based approach, there is also a matching part and this part is tried with two different matching algorithms that are called Brute Force Matcher and Approximate Nearest Neighbor (ANN) matcher.

The second algorithm makes the decision of match or non-match by feeding the raw fingerprint images as an input to a deep neural network and comparing the feature vectors calculated by the network. This study has revealed that deep neural network approach has given more robust and faster results on both the original dataset and distorted versions of the dataset.

ÖZET

PARMAK İZİ EŞLEŞTİRME ALGORİTMALARININ SENTETİK BOZULMALARA KARŞI DAYANIKLILIĞI

Parmak izi tanıma, insanların tanımlanmasında kullanılan biyometrik tekniklerden biridir. Parmak izi tanıma konusundaki gelişmeler ve araştırmalar, parmak izi tanıma ve doğrulama senaryolarının geliştirilmesinde büyük öneme sahiptir. Parmak izi tanıma sistemlerinin hemen hemen her yerde kullanılması ve kolayca erişilebilir olması, bu alanlarda yapılan araştırmalarla doğrudan orantılıdır.

Bir yüzey üzerinde parmak izi alırken, büyük olasılıkla bazı sorunlar olacaktır ve bu durum parmak izi tanıma performansını etkiler. Güvenli bir tanıma sistemi için bu tür sorunların sisteme nasıl tepki vereceği önemlidir.

Tezin amacı sadece mevcut özellik tabanlı parmak izi tanıma tekniklerinin değil aynı zamanda derin öğrenme, görüntü işleme temeli alan parmak izi tanıma tekniklerinin sonuçlarını elde etmek ve karşılaştırmaktır. Önemli olan, parmak izi doğrulama algoritmalarının sentetik olarak bozulmuş parmak izi görüntülerinin koşulları altında nasıl sonuçlar verdiğidir. İki farklı doğrulama sistemi geliştirdikten sonra, sistem sonuçlarını çarpık görüntülerle ve düzgün görüntülerle karşılaştırmaktır. Parmak izi görüntüsü üzeri harici olarak deformasyon etkisi olan ve olmayan iki yöntemin sonuçları karşılaştırılır. İlk sistem, görüntüleri parmak izi üzerindeki özelliklerle karşılaştıran özellik tabanlı bir tanıma sürümüdür. Bunu yapmak için ORB ve SIFT teknikleriyle iki farklı tanımlayıcı kullanılır. Bu konvansiyonel yaklaşımda, eşleşen bir parça da vardır ve bu kısım Brute Force Matcher ve Yaklaşık En Yakın Komşu (ANN) eşleştiricisi olarak adlandırılan iki farklı eşleştirme algoritmasıyla denenir. İkinci algoritma, ham parmak izi görüntülerini modele giriş olarak alarak parmak izi görüntülerini eşleştirmek için tamamen sinir ağı modelinin sonucunu alarak eşleştirme sonuçlarını belirler. Bu çalışma, derin sinir ağı yaklaşımının hem orijinal veri setinde hem de veri setinin çarpık versiyonlarında daha sağlam ve daha hızlı sonuçlar verdiğini ortaya koydu.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN . . . . . Aproximate Nearest Neighbor
CNN . . . . . Convolutional Neural Network
FAR . . . . . False Acceptance Rate
FRR . . . . . False Rejection Rate
HTER . . . . . Half Total Error Rate
EER . . . . . Equal Error Rate
FMR . . . . . False Match Rate
FNMR . . . . . False Non Match Rate
FMR . . . . . False Match Rate
FNMR . . . . . False Non-Match Rate
MC . . . . . Minutiae Code
ANN . . . . . Artificial Neural Network
RELU . . . . . Rectified Linear Unit
R-CNN . . . . . Region based Convolutional Neural Network

CHAPTER 1

INTRODUCTION

Each fingerprint is unique in its structure and is never the same as that of another person. When the finger is pressed on a flat surface, the pattern structure at the finger leaves a trace of the same pattern on the surface which is called a fingerprint (Adán et al., 2008). Fingerprint has various characteristics, such as line patterns, line frequencies, locations of reference points (i.e. core and delta points), locations of minutiae, and so on. Minutiae refers to specific points in a fingerprint, and these points are the small details in a fingerprint such as ridge ending, bifurcation, and dot (Solutions, 2016). All the distinctive features supply the individuality and uniqueness of the fingerprint.

In the mid-1800s, scientific studies were carried out to determine the characteristics of two fingerprints. These characteristics were that two fingerprints from different fingers did not have the same line pattern, and fingerprint calligraphy patterns did not change throughout life. These works led to the use of fingerprints for identification in judicial cases, first in Argentina in 1896, second in Scotland Yard in 1901 and then in other countries in the early 1900s. The computerization of the fingerprinting started with the use of computer technologies in the early 1960s. Increasing of the computer hardware capabilities improved the processing of these images. Since then, automatic fingerprint identification systems have been widely used in judicial units worldwide. In the 1980s, innovations in technology, such as personal computers and optical scanners, enabled fingerprint detection to be practical in non-punitive applications. In the late 1990s, the introduction of cheap fingerprint reader devices and the development of fast, reliable mapping algorithms paved the way for tailoring fingerprints to personal use (Saleh, 2014). The importance of fingerprint is that being always unique feature of human and easily obtainable.

In this thesis, two fingerprint verification methods, which are feature and deep learning-based, are evaluated and tested with and without synthetically distorted images. Data augmentation technique is used both for increasing the number of training data and for testing models with synthetically distorted data. Based on these methods presented, the motivation of the thesis is to compare the results of these two methods according to experimental setup that is explained in Chapter 4.

The thesis consists of five chapters and its outline is given as follows:

- Chapter 1 serves as an introduction to explain the definition, the history and the importance of fingerprint. In addition, a brief summary of the thesis is provided.
- Chapter 2 concentrates on the background of biometric recognition, fingerprint recognition and problems of fingerprint degradation. This chapter also gives the related works about the three main topics.
- Chapter 3 gives general information about the two techniques that are used for fingerprint verification. Additionally, another important contribution of this thesis, namely, synthetic distortion generation is included in this chapter.
- Chapter 4 present the experiments and results of the two different fingerprint verification algorithms that are feature-based method and a fully convolutional neural network (CNN) model. In this chapter, database that is used for the experiments performance evaluations and comparisons of these algorithms are explained for both non-distorted images and distorted images.
- Chapter 5 summarizes the two different ways of fingerprint recognition and gives the final remarks.

CHAPTER 2

BACKGROUND

2.1. Biometric Recognition

Biometric data should have measurable and unique features. For instance, physical characteristics of people such as signature, sound, keystrokes are used for biometric systems. Biometric systems can be defined as automated systems that process this biometric data, identify and match identities. There are many biometric systems that can be utilized to recognize and identify people's characteristics. The most common among those systems are fingerprint recognition, face recognition, iris recognition, finger vein pattern recognition, palm vein pattern recognition (Recogtech, 2019).

With the increase of computation power of computers and electronic technologies, the popularity of biometric systems, which are powerful in terms of security, have become popular (Pato and Millett, 2010).

Biometric recognition gives reliable results because human specific individual biometric properties are used. Methods such as password, pin number and so on are much weaker than biometric identification methods since they can be easily stolen. Users do not need to have a card or password in biometric systems. Each person is the key of his or her own security system. It is safe against risks such as stealing and copying. Therefore, biometric systems are secure and convenient to use (Jain et al., 2004). Based on the application context, a biometric system may operate either in verification mode or identification mode Jain et al. (2004). Figure 2.1 demonstrates the procedure of verification and identification processes.

- Verification Mode: It validates a person's identity by comparing the acquired biometric data with the biometric patterns stored in the gallery. That is, it verifies whether this data belongs to that person or not. The system makes a one-to-one comparison to determine whether the claim is true or not Wayman (2001).
- Identification Mode: This mode identifies who biometric data belongs to. Identification needs a one-to-many comparison to find an induvidual's indentity Wayman (2001).

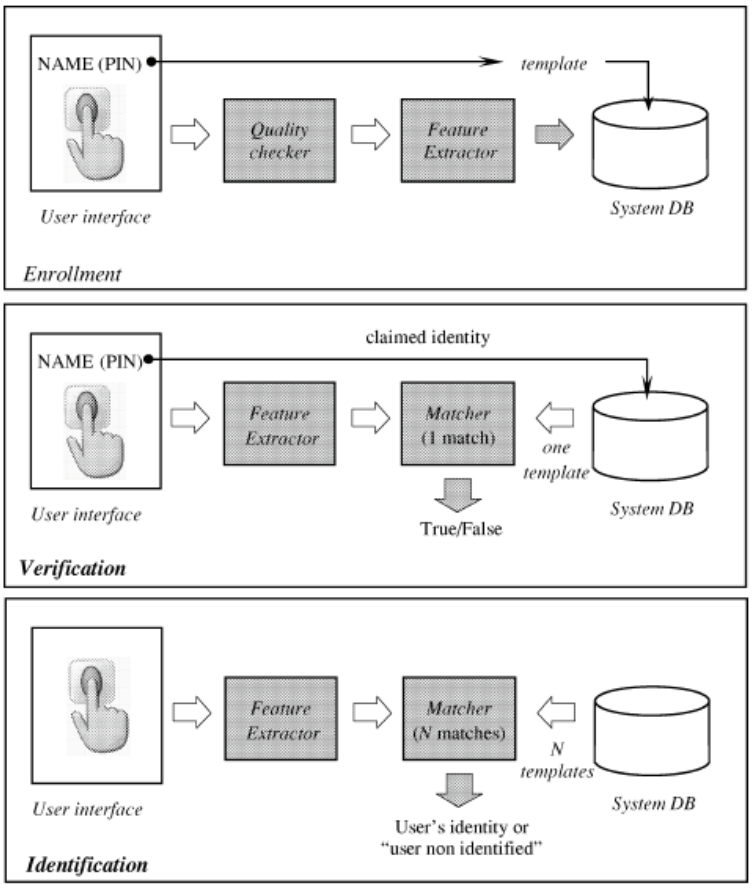

Figure 2.1. Block diagrams of enrollment, verification, and identification tasks are shown using the four main modules of a biometric system, i.e., sensor, feature extraction, matcher, and system database Jain et al. (2004).

Performance Metrics for Verification: Comparison of two different fingerprints gives match or non-match result. The metrics for successful and erroneous results are listed below:

- False Acceptance Rate (FAR): The ratio of the ”match” result for two different fingers. It is also known as False Match Rate (FMR).
- False Rejection Rate (FRR): The ratio of the ”not match” result for two same fingers. It is also known as False Non Match Rate (FNMR).
- Half Total Error Rate (HTER): (FAR + FRR) / 2
- Equal Error Rate (EER): This is used for determine the threshold value for the decision of FAR and FRR. The lower EER value means the higher accuracy of the biometric system. At this point FAR and FRR values are equal.

The Figure 2.2 explains the relationship between False Match Rate (FMR) and False Non-Match Rate (FNMR) over genuine and imposter score distribution. Two probability distributions of genuine and impostor pairs are shown in a joint way. Matching score close to 1 indicates that the images are from the same finger. The areas under the graphs give the FNMR and FMR values.

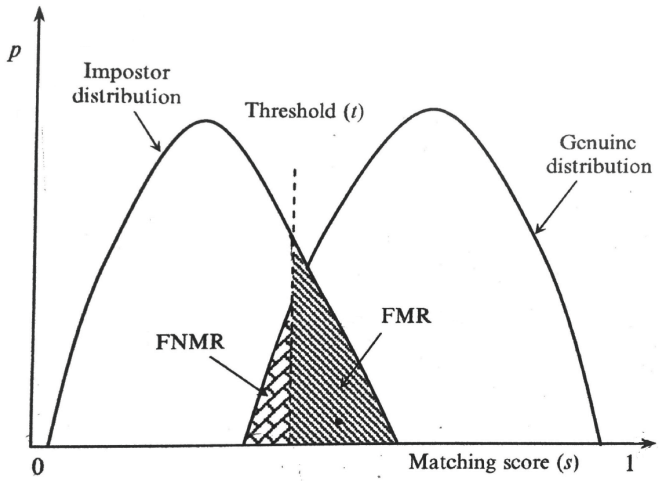

Figure 2.2. FMR and FNMR for a given threshold t are displayed over the genuine and impostor score distributions (Maltoni et al., 2009).

There is a real trade-off between FMR and FNMR. Figure 2.3 roughly demonstrates this tradeoff. The system can give different results depending on the threshold value of t. Changing the value of threshold increases or decreases the values of FMR and FNMR. It can be seen that EER point is where FMR and FNMR values are the same. Criteria for threshold selection will be based on this EER point in the fingerprint recognition algorithms.

In this thesis FAR, FRR, HTER and EER metrics will be evaluated.

2.2. Fingerprint Recognition

Fingerprint recognition consists of algorithms for extracting salient features from fingerprints and matching them. A crucial problem is that fingerprint images may be in poor quality, because of dirty/damaged sensor surfaces, wet/dry fingertips, uncooperative users, etc. Even though fingerprint matching problem has been studied for almost four decades, large intra-class variability due this problem is still a challenge.

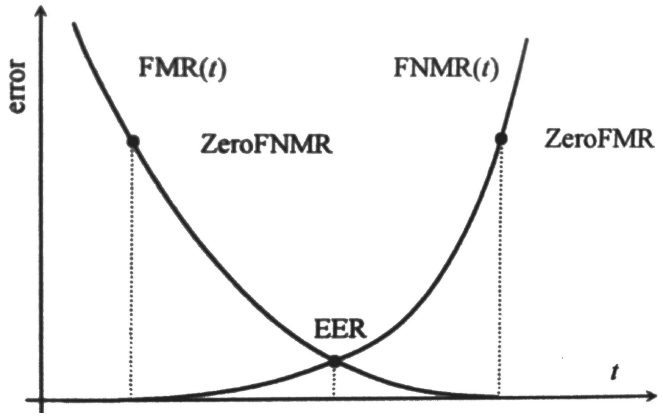

Figure 2.3. A demonstration of FMR(t) and FNMR(t) curves, the points corresponding to EER, ZeroFMR, ZeroFNMR (Maltoni et al., 2009).

2.2.1. Methods

This section will be discussed in three main topics. The first is the human approach that explains how forensic experts recognize fingerprints. The second topic explains how feature-based method works and the last one focuses on deep learning method.

2.2.1.1. Human Approach

A fingerprint is the result of the regeneration of the fingertip epidermis, which forms lines and valleys (M. Bolle et al., 2002). In fingerprint images, lines and valleys are represented by dark and bright areas respectively. Lines increase friction between the surfaces of the fingers and other objects that they are useful for gripping and maximizing the ability to recognize different textures.

In the human approach, acquisition can be conducted both offline or live scan. Then, a good representation of fingerprint is obtained by means of extracting the salient features properly.

The fingerprints are categorized at the global level according to the general line or valley pattern that provides the separator configuration. This classification leads to three general classes; they are a loop, arch, and whorl. These classes can be further divided into the right loop, left the loop, arch, tented arch, whorl. These can be observed in Figure 2.4.

Figure 2.4. Images of basic fingerprint classes, right loop, left loop, arch, tended arch, whorl

Next, the core and minutiae points of the compared fingerprints are manually labelled and compared by fingerprint experts. The decision is usually is made based on the total number of matched minutiae points.

2.2.1.2. Feature-Based Approach

Fingerprint Image Enhancement : The accuracy rate of the recognition and verification systems used for fingerprints highly depends on the quality of fingerprint images and the ability to extract details. The quality of embossed structures is an important characteristic and contain the characteristic detail information required for detail extraction. A line structure must first be obtained in order to extract details from the fingerprint image obtained by digitizing an inked fingerprint or scanning directly from the sensor. The line structures in the fingerprint may not always be well defined. Therefore, preprocessing the image, is necessary to obtain more reliable detail extraction.

If the captured fingerprint image contains various noises, matching results may deteriorate. Pre-processing is highly important for detecting minutiae and extracting fatures. Commonly used enhancement technics are histogram equalization, binarization, thinning and Gabor filtering. In Figure 2.5, an original fingerprint is shown with its binarized and thinned versions.

Fingerprint Feature Extraction : Many studies have been done in the area of feature extraction. Using image processing techniques, such as convolving a specific type of filter on images, fingerprint features can be extracted in an enhanced image. Among the variety

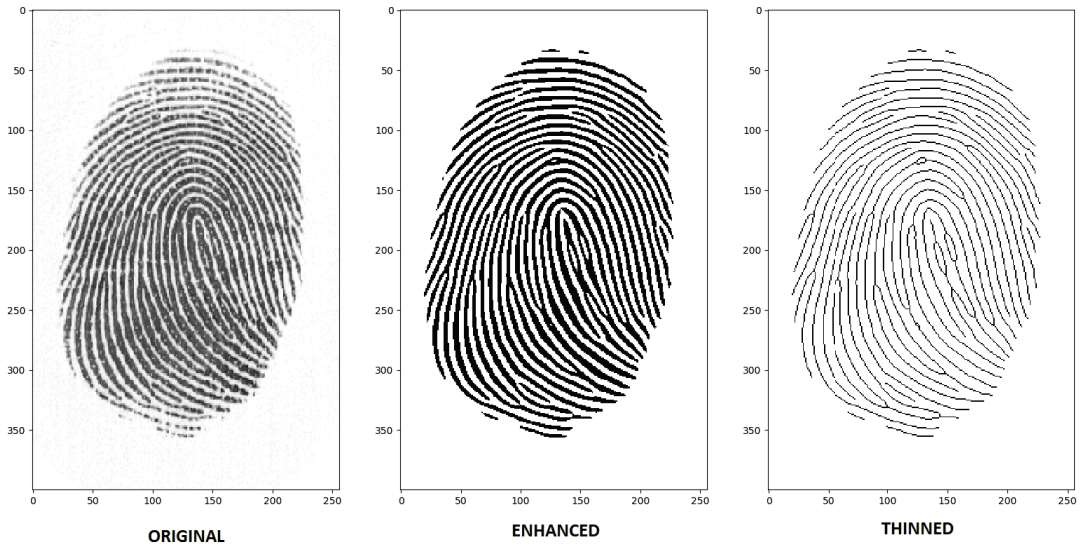

Figure 2.5. Demonstrations of some preprocessing steps

of minutiae types, the most significant ones are bifurcations, ridge endings, islands, dots, bridges and ridge closings.

In Figure 2.6, features of a fingerprint are shown.

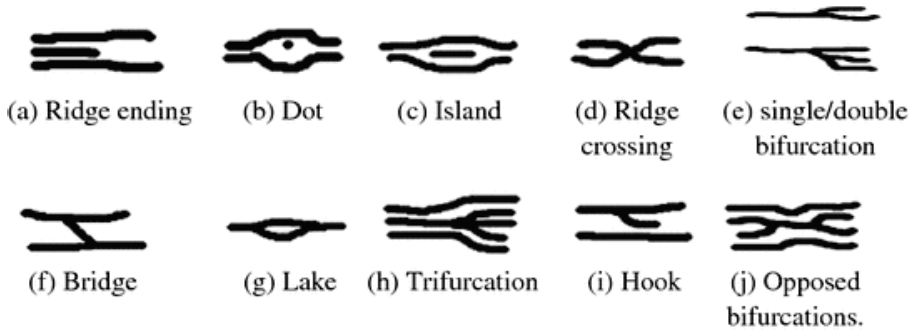

Figure 2.6. Mostly used features of fingerprint (Source: Elmouhtadi et al. (2018))

Convolving 3x3 filters on the fingerprint image can help us to extract some features. In Figure 2.7, 3x3 filters to detect ridge endings and bifuractions are given.

Minutiae Matching : A fingerprint is made of a series of ridges and valleys on the surface of the finger. The uniqueness of a fingerprint can be determined by the pattern of ridges and valleys as well as the minutiae points.

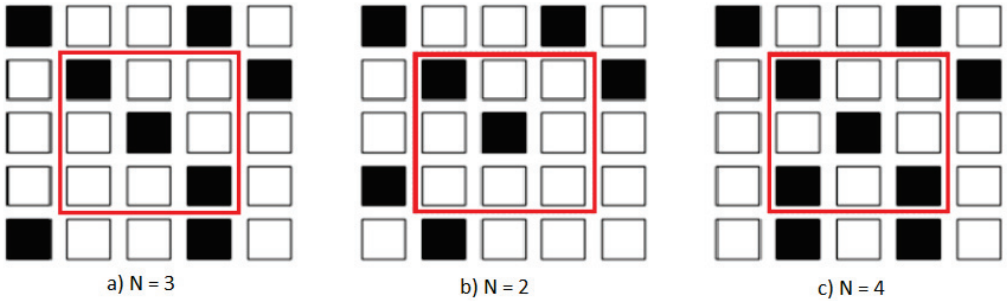

Figure 2.7. 3x3 windowing for feature extraction to detect ending and bifurication points (Maltoni et al., 2009).

Minutiae points are local ridge characteristics that occur at either a ridge bifurcation or a ridge ending. The minutiae-based matching algorithm determines whether the two minutiae sets are from the same finger or not. After image processing and feature extracting stages, following three stages are applied to image for fingerprint verification:

- Alignment stage: Given two fingerprint images to be matched, minutiae are transformed into a new coordinate system according to the similarity of ridges.
- Matching stage: After obtaining two sets of minutia points, the Brute-Force matching algorithm is used to count the matched minutiae pairs.
- Decision stage: The calculated matching similarity between two fingerprint images is applied to a thresholding stage. Based on this threshold value, images are being said if they match or not.

Related Works : In the work of (Aguilar et al., 2007), a good matching performance for fingerprints with high variation is obtained by using minutiae codes (MC). Firstly, minutiae points are extracted from a fingerprint image. The proposed MC is invariant to rotation of the fingerprint image. Adjustment factor is introduced to handle the problem due to differences. Different images of the same fingerprint or different inking or pressure can be addressed by this approach. The adjustment factor is calculated from the minutiae code of the two fingerprints being matched. There are two stage fingerprint matching processes. In the first stage, only a few minutiae codes are checked to decide if the second stage of the matching process is required. This matching technique creates a faster process.

The authors propose a very sophisticated and powerful method to recognize fingerprint images in the paper (Jain et al., 2000). Some missing parts on the traditional

ways of fingerprint recognition are mentioned by the authors. One of them is that minutiae based structure cannot cover the whole information about the fingerprint. The method developed against these problems is a filter-based approach. It mainly uses Gabor filters to take both local and global structures. That information is stored as finger code. Euclidean distance is used in matching phase due to its speed. In this approach, the false acceptance rate is very row.

2.2.1.3. Deep Learning Approach

For fingerprint recognition, developing neural network-based solutions is another option instead of using conventional image processing methods. The following sections give a brief explanation of techniques and environments to set up a model for fingerprint verification.

Deep learning has become the hot topic of the tech environment during recent years. Deep learning is an advanced machine learning method that takes input and gives the predicted output. Deep learning models are created and developed based on Artificial Neural Network (ANN) techniques that were inspired by neural connections in hte brain. It includes lots of artificially created neurons connected to each other. These neurons are connected to each other with synapses which are called 'weights' in ANN literature. This kind of models can perform tasks such as classification, pattern recognition, feature extraction, and clustering. Given an input to the model, the process of learning a given task can happen with a forward and backpropagation to update the weights. This is also known as the training of a model.

During the learning period, weights are updated according to the input data and backpropagation errors. Weight changes refer to learning. If there are no weight changes in the ANN, the learning process stops. Initially, these weight values are randomly assigned. ANNs change these weight values as input samples are fed into the network. The aim is to find the weight values that will produce the correct outputs for the samples given to the network. If the network reaches the correct weight values, it means that the model have the ability to make generalizations of the samples for test cases.

Convolutional Neural Network (CNN) : Convolutional Neural Network is an Artificial Neural Network that is most popularly used for analyzing images. CNN's can also be used for other data analysis or classification problems as well. Most generally, we can think of CNN as an artificial neural network that has some type of specialization for

being able to detect patterns. This pattern detection is what makes CNN useful for image analysis.

CNN has hidden layers called convolutional layers and these layers are precisely what makes it special for complex problems. Hidden layers make network to capture the complex relationships between features in the input. Layers usually have non-convolutional layers as well but the basis of a CNN is the convolutional layers. Just like the other layers, the convolutional layer receives input, propagates the input in some way and then outputs the transformation to the next layer. With a convolutional layer, this transformation is a convolution operation.

While simple CNN model is able to detect and classify simple shapes, edges or objects in an input image, in a deeper network, these filters can detect specific objects like eyes, ears, hairs or further features scales and even deeper layers the filters are able to detect even more sophisticated features like biometric objects in fingerprint images MathWorks (2019).

Here is listed basic explanation of CNN layers:

- Convolutional Layer: Core of the CNN models. Convolutional layers apply filtering process to extract features in an image and outputs feature maps as a result of this filtering.
- Non-Linearity Layer: This layer consists of activation function taking feature maps and outputs activation map.
- Rectification Layer: Performs element-wise operations on the volume.
- Pooling Layer: Used to downsample the number of weights for reducing the size of maps.
- Dropout Layer: This layer has a function for reducing overfitting problem by randomly ignoring selected neurons during training process.
- Fully-Connected Layer: This layer aims to map a class probability distribution for classification tasks.

An example CNN model and its layers can be shown in Figure 2.8.

DenseNet : For an advanced image classification and processing tasks, CNN models are getting deeper and deeper through the years. When the neural networks get deeper the problem of vanishing gradient starts to play more and more important role in developing

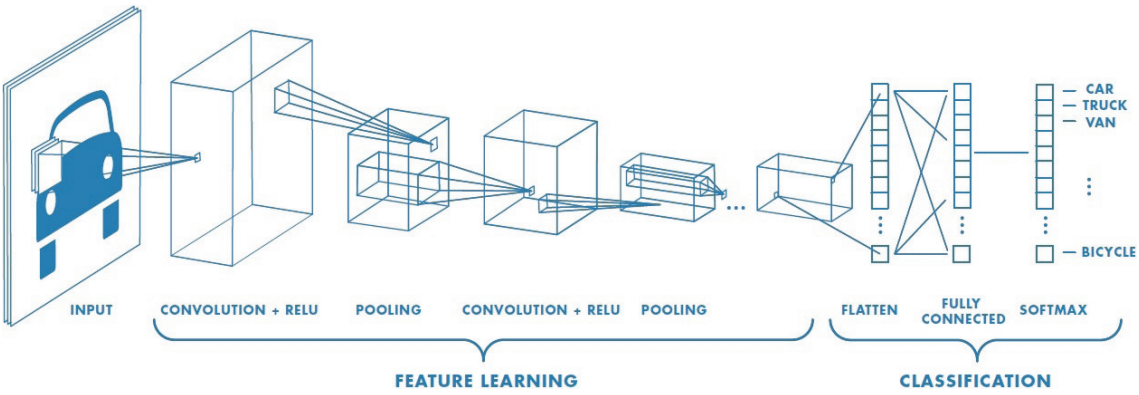

Figure 2.8. Layers of a CNN model (Source: MathWorks (2019))

CNN models. To overcome this problem, DenseNet model simplifies the connectivity pattern through layers by connecting every layer to the subsequent layer. This also means that every layer has the information of the feature maps of the preceding layers (Huang et al., 2017).

Figure 2.9 shows the layers and feedforward connections between each layer for the DenseNet model.

DenseNet model has a dense block which includes convolutional layers. These convolutional layers have batch normalization, rectified linear unit (RELU) and convolution operations, respectively.

DenseNet offer a varied number of models differs on a number of dense layers which also means the parameter size. In this thesis, DenseNet121 is trained and applied in the developed algorithm. The total number of trainable parameters in this model is 7M. Figure 2.10 shows the layer structure of DenseNet121.

Keras : Keras is one of the most important packages about deep learning (Chollet et al., 2015). In fact, Keras itself is not a deep learning library. Keras offers a high-level API that you can use in Google Tensorflow, Microsoft CNTK, and Theano deep learning libraries. In this way, you can train your deep learning architecture using different packages.

Keras is also coming one step more with being user-friendly via implementing easily, modularity especially in neural layers, cost functions, optimizers, initialization schemes, activation functions, and regularization schemes. These are all standalone modules that you can combine to create new models. It provides easy access to extensive different models and the other advantage is that it works with Python that doesn't need separate models configuration files in a declarative format. Models are described in Python

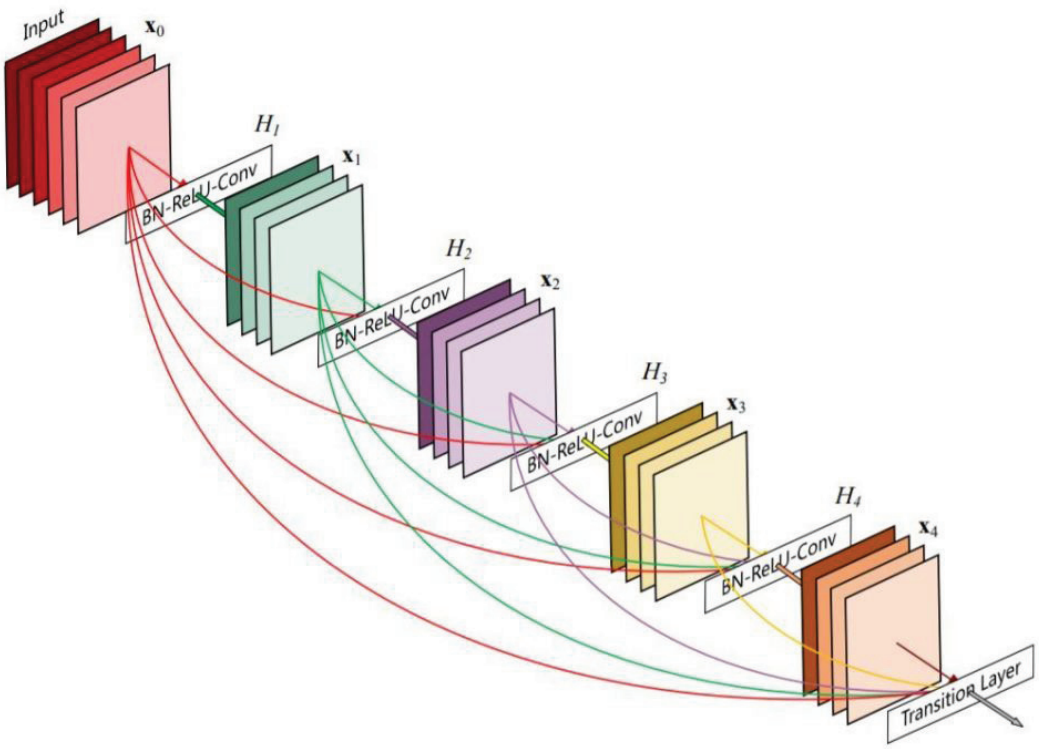

Figure 2.9. Feed Forward Connections in DenseNet (Source: (Huang et al., 2017))

code, which is compact, easier to debug, and allows for ease of extensibility.

The algorithm developed within this thesis is fully CNN model and this model is created using Keras.

OpenCV : OpenCV is a library of computer vision techniques. In order to do that it has several algorithms for image processing and machine learning. With numerous functions of OpenCV, many applications such as face recognition, object classification, detecting human movements, plate recognition, processing in three-dimensional images, image comparison using this libraryis possible. OpenCV also supports Deep Neural Network libraries such as Tensorflow, Torch/Pytorch, Caffe, Keras (team, 2000).

Detecting intrusions from the surveillance video, monitoring of mining equipment, detection of drowning in swimming pools, events such as the interpretation of images from many sources, from professional devices to mobile phones carried by everyone, can be given as examples of OpenCV's daily applications.

Related Works : A new fingerprint matching method based on Convolutional Neural Networks (ConvNets) has been proposed in (Zhu et al., 2018). The method is basically

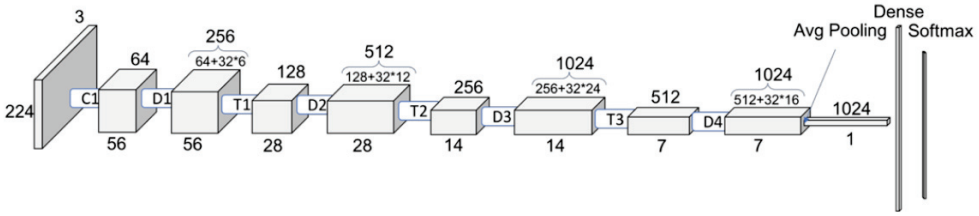

Figure 2.10. Layer Structure Demonstration in DenseNet121

designed as a classification problem that decides as match of non-match for each fingerprint comparison. The most important feature of this method is to learn the relational features directly and to decide the similarities at the pixel level. Due to the robustness of the algorithm during feature extraction on fingerprint images, the proposed method can be applied to partial fingerprint matching.

In (Lu Jiang et al., 2016), authors focus on minutiae for fingerprint recognition. Minutiaes are usually extracted after some image processing methods like thinning binarization and some other enhancement techniques. Nevertheless these operations are not very reliable. This work mainly focuses on using deep learning for minutiae extraction. The algorithm can be applied directly to the raw fingerprint images. Moreover, the accuracy is higher than the other methods for extracting minutiae and some techniques are used for both to avoid overfitting and to improve the robustness. Thanks to this approach system can achieve good performance as it not only makes all use of information in fingerprint images but also learns the minutiae patterns from large amounts of data.

Another paper (Wang Yani et al., 2016) proposed a robust damaged fingerprint recognition algorithm, which is based on the CNN. It not only has a high resistance to abnormal degeneration, but the recognition process is also simpler than the feature points matching algorithm. The recognition rate based on deep learning is compared with the fingerprint identification algorithm based on Kernel Principal Component Analysis. Experiments results show that fingerprint recognition based on deep learning has higher robustness.

Both (Peralta et al., 2017) and (Ruxin Wang et al., 2016) mainly focus on using deep learning techniques which is CNN for fingerprint classification to identify the fingerprints in large datasets. These methods avoid the necessity of an explicit feature extraction process by incorporating the image processing within the training of the classifier. Moreover, these approaches are also good for low-quality images and gives better results for

extracting features and higher success rate on classification.

In (Liu et al., 2018) authors mainly focus on extracting singular points via Region based CNN (R-CNN) method with an orientation constraint. The specialty of this work is that it can extract singular points from the image directly and there is no need for image preprocessing before recognition. The proposed algorithm achieves 96.03% detection rate for core points and 98.33% for delta points which outperforms other algorithms.

2.2.2. Fingerprint Degradation

Lack of robustness against image-quality degradation is one of the biggest problems in fingerprint verification. Poor quality images give bad results most of times. The quality of fingerprint images have great impact on the results. Biometric sample quality is categorized in three points: 1)Physical characteristic features of the subject 2)Fidelity, which is the degree of similarity between a biometric sample and its source, attributable to each step through which the sample is processed 3)Utility, which refers to the impact of the individual biometric sample on the overall performance of a biometric system, where the concept of sample quality is a scalar quantity that is related monotonically to the performance of the system.(Grother and Tabassi, 2007)

Besides the characteristic of fingerprint, the environmental factors such as humidity, temperature has bad effects the acquired fingerprint samples.

Related Works Poor-quality images result in spurious and missing features, thus degrading the performance of the overall system. Therefore, it is important for a fingerprint recognition system to estimate the quality and validity of the captured fingerprint images. Existing approaches for fingerprint image quality estimation are (Alonso-Fernandez et al., 2007).

CHAPTER 3

METHODOLOGY

This chapter mainly focuses on the evaluation of two different fingerprint verification techiques give the technical background of the developed algorithms. Many studies have been done so far, and many new technologies have been used and developed. Recognition algorithms applied with advanced technology have given speed and capacity to fingerprint recognition systems. Some information about the fingerprint systems and technical features that have been developed from the past will be explained.

In this thesis, two different fingerprint verification algorithms are analyzed considering both feature-based method and deep neural network method. Performance of the algorithms is evaluated on the MCYT-100-Fingerprint dataset.

3.1. Feature-Based System

A fingerprint identification system is developed that is based on the method of finding features of fingerprint. Then, these features are used to find the formal descriptors of the region around them that identifies the fingerprint itself. The system is tested on a subset of the MCYT-100 Fingerprint dataset. General flowchart of the system can be seen in Figure 3.1.

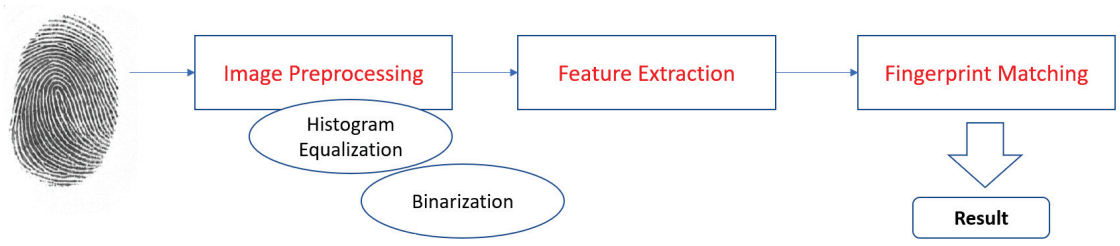

Figure 3.1. Fingerprint Recognition General Flowchart

3.1.1. Image Preprocessing

Some preprocessing techniques are required to be applied to the fingerprint image before the verification or identification steps. In this section, used image processing techniques are explained.

3.1.1.1. Histogram Equalization

The first process to be applied on a fingerprint image is local histogram equalization to the image. By equalizing the histogram of an image, pixel intensities are streched out in histogram range. As a result of histogram equalization, image contrast is enhanced. By applying local histogram equalization, image becomes enhanced using small windows instead of global histogram equalization.

3.1.1.2. Normalization

The second process to be applied on a fingerprint image is normalization of the image. Normalization is a pixel-wise operation and that does not change the clarity of the ridge and valley structures. The purpose of the process is to reduce the variations in gray-level values along ridges and valleys, that makes the subsequent processing steps easier (Lin Hong et al., 1998).

3.1.1.3. Binarization

Binarization of an image allows us to clear the image from unnecessary noise like ”wrinkled” surface. The Otsu threshold is used to automatically select the best generic threshold for the image to get a good contrast between foreground and background information. After this process, the image is converted to binary form and have only two different values.

3.1.2. Detecting and Comparing Feature Points

The features of fingerprints are detected with two different algorithms called Oriented FAST and Rotated BRIEF (ORB) and Scale-Invariant Feature Transform (SIFT). The ORB descriptor is mainly based on BRIEF, that is not only rotation invariant but also resistant to noise. SIFT is the mostly used well known feature detection-description algorithm. Sift detector is based on Difference of Gaussians (DoG). Feature points are detected by DoG by using local maxima of various image scales. Once the features are detected in an image, vectors of descriptors are obtained and matched. The simple and efficient ways to compare these descriptors are with the so-called Brute-Force matching and Approximate Nearest Neighbour (ANN). Brute-Force matching simply calculates the hamming distance between the descriptors. The technique of matching tries all possibilities. With these ways, we will get an estimate indicating how similar are two fingerprints. This type of matching is the most used and well-known matching technique. Figure 3.2 shows the matches of the two fingerprints of a identical finger. ANN tries to find approximate nearest neighbor features. By doing this, the number of mathing decrease but this is not mean that it will find the best matching. Parameters of ANN can change according to demand of application.

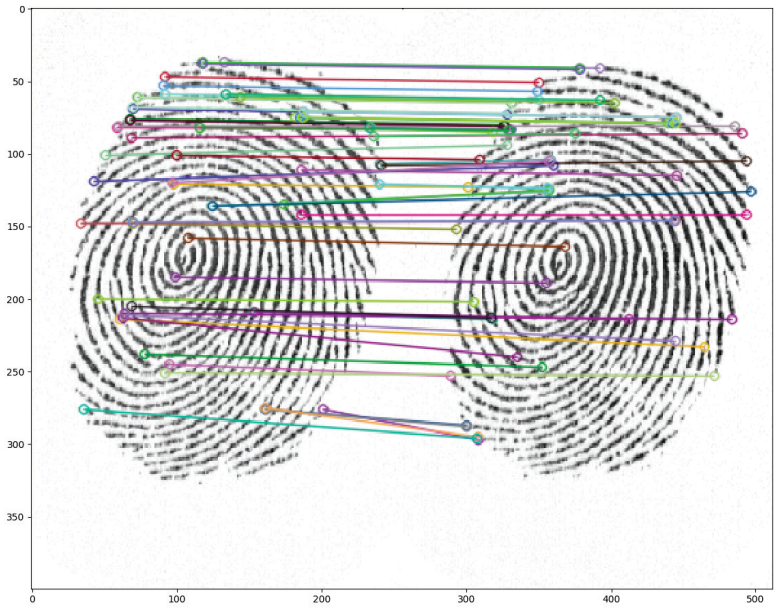

Figure 3.2. Brute Force Matching of two images belongs to same finger

3.2. Deep Neural Network System

In the CNN method, which is based on Densenet121 model, the neural network is trained to verify two fingerprint input images if they match or not. CNN model takes raw fingerprint input images as inputs.

3.2.1. Data Augmentation

In order to train a CNN model properly, a large number of training data is required for input classes. For this purpose, each fingerprint sample is generated 5 times with different zoom and rotation values. Training dataset also includes the distorted version of the original fingerprints.

Keras library has a function to generate images with random zoom, shear and rotation values. In Figure 3.3, the original and two images generated with the help of Keras is shown.

Figure 3.3. MCYT-100 database with two different transformation

3.2.2. Training of neural network model and Feature extraction

During the training process of the CNN model, the first half of the dataset images are selected for training and the rest of the images in dataset are used for testing of the model. In this case of MCYT-100 dataset, the total number of training class is 100. After the augmentation of original and distorted training data, each class has 240 number of samples.

The network is trained using Adagrad optimizer. With this optimizer, parameters have specific learning rates depending on how frequently they get updated during training (Duchi et al., 2011).The simulation parameters of the algorithm are listed in Table 3.2.2.

Densenet121 model is trained to classify half of the classes in dataset. After the training process, this pre-trained model is used for feature extraction. Figure 2.10 shows the overall architecture of Densenet121 model. The feature vector is extracted after the global average pooling layer which is before the last layer of the model, (fully connected layer).

Table 3.1. Simulation and Network Model Parameters

Parameter	**Value**
Number of Epoch	10
Batch size	4
Number of training class	100
Number of training data	24000
Number of testing class	50
Validation Split	0.20
Graphics Card	NVIDIA GeForce 1050Ti

Extracted feature vectors have a shape of 1x1024. They are used for verification of test images. Figure 3.4 shows two different fingerprint images given as input to the network.

As a result of each image, model gives a feature vector and Euclidian distance between these two vectors, δ, are calculated as:

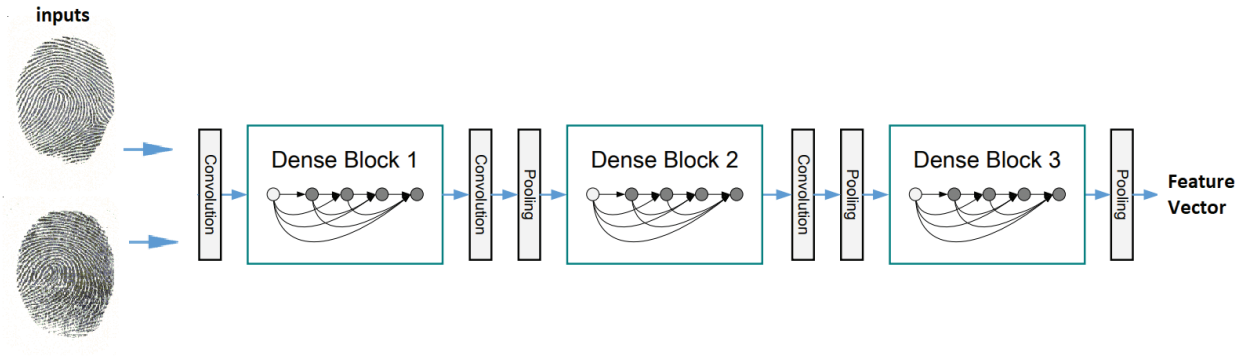

Figure 3.4. Two input images to the model

$$\delta = \sqrt{\sum_{i=1}^{1024}(x_i - y_i)^2} \tag{3.1}$$

where x_i and y_i are the feature vector values.

According to a predetermined threshold value thr that are calculated for each experiment as explained in Section 4.2, the fingerprints are decided to match or not.

$$X = \begin{cases} NotMatch, & \text{if } thr > \delta \\ Match, & \text{otherwise} \end{cases} \tag{3.2}$$

3.3. Applied Distortions

Noise in an image is any degradation caused naturally or by external disturbance. The goal of generating synthetic distortions is evaluating the robustness of fingerprint recognition systems for two developed algorithms. Adding distortions can make images hard to recognize. As a result, it will show the robustness of the two algorithms against the external effects.

Three types of distortions are applied synthetically. These are gaussian, concave effect and salt and pepper noise on the fingerprint images. In Figure 3.5, the original image, and its noisy variants can be seen.

Gaussian noise has the statistical properties of a normal distribution which is also known as Gaussian distribution. The probability density function of a Gaussian random variable is given by:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{3.3}$$

where σ is a standard deviation and μ is the mean value of the Gaussian distribution. While generating Gaussian distributed noisy images, σ is chosen as 1.5 and the mean value μ is chosen as 0.

The other type of noise is called salt and pepper noise and also known as impulse noise and caused by sudden disturbances in the image. Noise has an effect of randomly distributed black or white pixels on the image.

While generating salt and pepper noise, a fixed ratio of noise is added over the image. This ratio is selected as 5% of the total number of image pixels.

The last type of distortion is called concave effect and it can affect both in x and y-axis of the image. It has an effect like a concave mirror by offsetting the image pixels sinusoidally.

Figure 3.5. Original image and synthetically distorted versions

CHAPTER 4

EXPERIMENTS AND RESULTS

4.1. Database

Dataset used in the development of biometric recognition systems is of great importance for any supervised learning. Since biometric systems play a critical role in vital cases, its databases should be large and various for testing of recognition systems in actual working conditions.

In the thesis, the dataset that is used for performance evaluation is the MCYT bimodal database. This dataset was acquired by four Spanish academic institutions in 2001 (Ortega-Garcia et al., 2003). It consists of both fingerprint and online signatures. Collecting the fingerprint images was handled with two types of devices.

The name of the first device is CMOS-based capacitive capture device, model 100SC from Precise Biometrics, with a resolution of 500 dpi and the second device is an optical capture device, model UareU from Digital Persona also with a resolution of 500 dpi. While the capacitive device is 89 kB and the resolution of images is 300x300, the case of the optical device's size is 102 kB and its resolution is 256x400 pixels.

In order to estimate recognition performances under different acquisition conditions, the MCYT Fingerprint has 12 different samples for each finger of a person and there is the same number of data for two different sensors. Briefly, each individual provides a total number of 240 (10x12x2) fingerprint images to the database. The dataset has 100 individual people in total.

Figure 4.1 shows the example fingerprint images captured with CMOS-based capacitive and UareU devices in MCYT database.

Figure 4.1. MCYT-100: Images of the same finger collected with two devices

4.1.1. Existing Results of MCYT Dataset

MCYT dataset is created for testing of levels of image quality. In the view of that, some experiments are done in (Ortega-Garcia et al., 2003). 4 different quality of groups. The quality of images increase from group 1 to 4. And as a result while the least quality group has 2.1% of EER, the highest quality group has 5.5% EER.

4.2. Experimental Setup

MCYT-100 dataset has twelve images of all 10 fingerprints of 100 people. But in this thesis, only two randomly selected fingerprints of each person is utilized. In other words, 200 fingerprints with 12 acquisitions are included in this study. For each experiment, this subset is further divided into training (50%), validation (25%) and test (25%) partitions randomly. This random division is applied 5 times, creating 5 folds for the experiments. In this way, the algorithms could be tested for their generalization capabilities. From training and validation folders, same fingerprint pairs (intra-class pairs) and different fingerprint pairs (inter-class pairs) are generated to determining the most accurate equal error rate (EER) value. To find the EER point all possibilities are calculated and ROC curve is plotted. The crossing value of FAR and FRR is taken as a threshold value

that is called EER.

Training and validation datasets for each trial contains both non-distorted and distorted image types. By doing this, ROC values will be chosen considering both image types and algorithms are expected to be more robust against the external noises.

In order to prepare a fair experimental set-up, the same sets of fingerprint pairs that are generated as intra-class and inter-class pairs, used for testing for all algorithms.

- For Feature-Based Model: There is no training procedure because of that reason the training folder is not used for the feature-based model. In each set, there is 25% randomly initialized validation folder. From the folder, same and different finger pairs are generated for determining the most accurate equal error rate value. To find the EER point, all possibilities are calculated and ROC curve is plotted. Only the curves of the feature-based algorithm with ORB and BF Matcher is shown for group 1 are shown and the plots and the decision criteria are the same for the rest algorithms.

- For CNN Model: Training folder for every other group of dataset covers randomly chosen %50 of MCYT-Dataset. In each group, there is %25 randomly initialized validation folder. From the folder, intra-class and inter-class pairs are generated for determining the most accurate equal error rate value. To find the EER point all possibilities are calculated and ROC curve is plotted. The system is tested for each randomly initialized test data for every group. Test folders cover %25 of MCYT-Dataset.

4.3. Results with Feature-Based Approach

Performance results for feature-based approaches are obtained for both original and distorted images. The chosen threshold values are determined as discussed in the previous part. According to the threshold values, the test data is used for with and without distortion. There are total 4 different feature-based algorithms and results.

This section will cover the results of individual algorithms for all groups of data. The detailed and generalized comments on results will be explained in the Section 4.5.

The results for the algorithm with ORB descriptor and Brute Force matcher are written in the Table 4.1.

This algorithm gives close results for each of the groups. While the best result is obtained for the non-distorted dataset, worst result is obtained for Gaussian noise dataset.

Table 4.1. ORB Descriptor with Brute Force Matcher

ORB With BFMatcher Results			
Datasets		Mean	Variance
Non-Distorted Dataset	FAR	0.28	0.02
	FRR	0.30	0.03
	HTER	0.29	0.01
Gaussian Noise	FAR	0.18	0.04
	FRR	0.53	0.09
	HTER	0.35	0.03
Concave Noise	FAR	0.27	0.07
	FRR	0.25	0.04
	HTER	0.26	0.05
Salt and Pepper Noise	FAR	0.23	0.09
	FRR	0.44	0.10
	HTER	0.34	0.09

While FAR and FFR results are close to each other for non-distorted dataset, the difference is higher for noisy datasets. The threshold values differ from set to set. ROC curves for nondistorted and distorted data of ORB with Brute Force Matcher algorithm is demonstrated in Figure 4.2.

The results for the algorithm with ORB descriptor and ANN matcher are written in the Table 4.2.

Table 4.2. ORB Descriptor with Approximate Nearest Neighbor Approaches

ORB With ANN Results			
Datasets		Mean	Variance
Non-Distorted Dataset	FAR	0.30	0.05
	FRR	0.32	0.06
	HTER	0.31	0.03
Gaussian Noise	FAR	0.22	0.03
	FRR	0.37	0.40
	HTER	0.30	0.21
Concave Noise	FAR	0.36	0.05
	FRR	0.28	0.03
	HTER	0.32	0.03
Salt and Pepper Noise	FAR	0.25	0.04
	FRR	0.47	0.14
	HTER	0.36	0.10

This algorithm, which contains ORB descriptor with ANN matching technique,

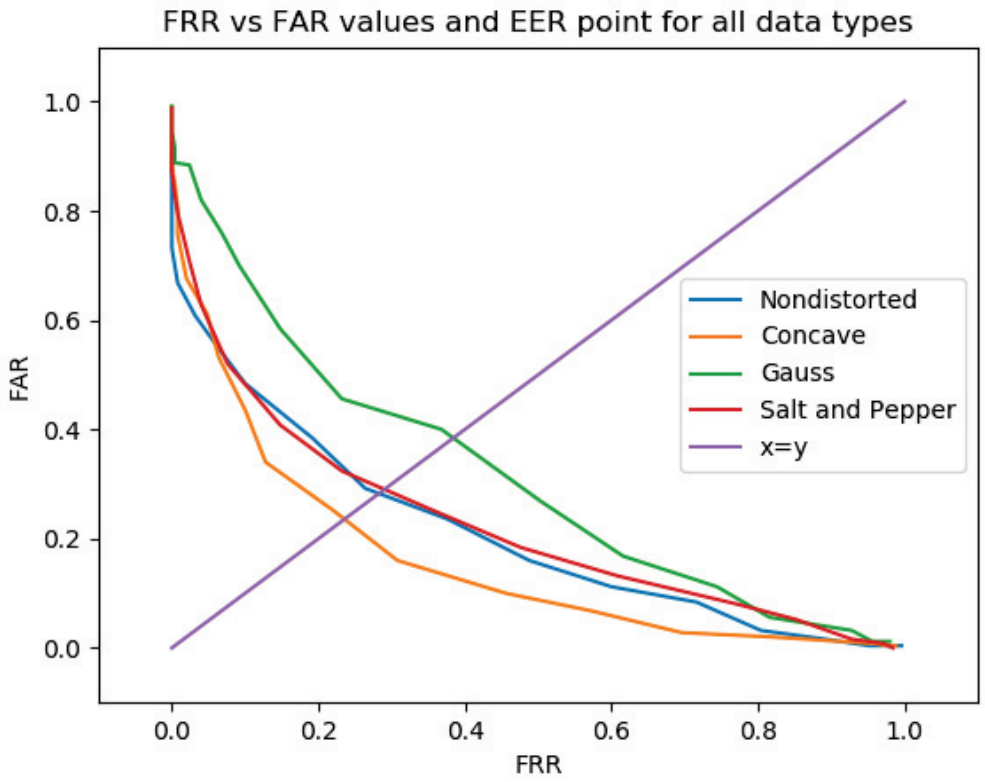

Figure 4.2. ROC Curves of all datasets for ORB with BFMatcher

gives similar results with the algorithm with ORB descriptor and Brute Force matcher for all distorted and non-distorted data types. ROC curves for nondistorted and distorted sets of data for ORB with ANN algorithm is demonstrated in Figure 4.3.

The results for the algorithm with SIFT descriptor and Brute Force matcher are written in the Table 4.3.

This algorithm gives far better results for both distorted and non-distorted datasets with the previous algorithms which use ORB descriptor in common. ROC curves for nondistorted and distorted sets of data for SIFT with Brute-Force matcher algorithm is demonstrated in Figure 4.4.

The results for the algorithm with SIFT descriptor and ANN matcher are written in the Table 4.4.

This algorithm, which contains SIFT as a feature extractor and ANN as a matcher, gives similar results with the previous algorithm. These last two tables show the significance of SIFT descriptor on FAR and FFR results. The detailed and generalized results will be explained in the following sections. ROC curves for nondistorted and distorted

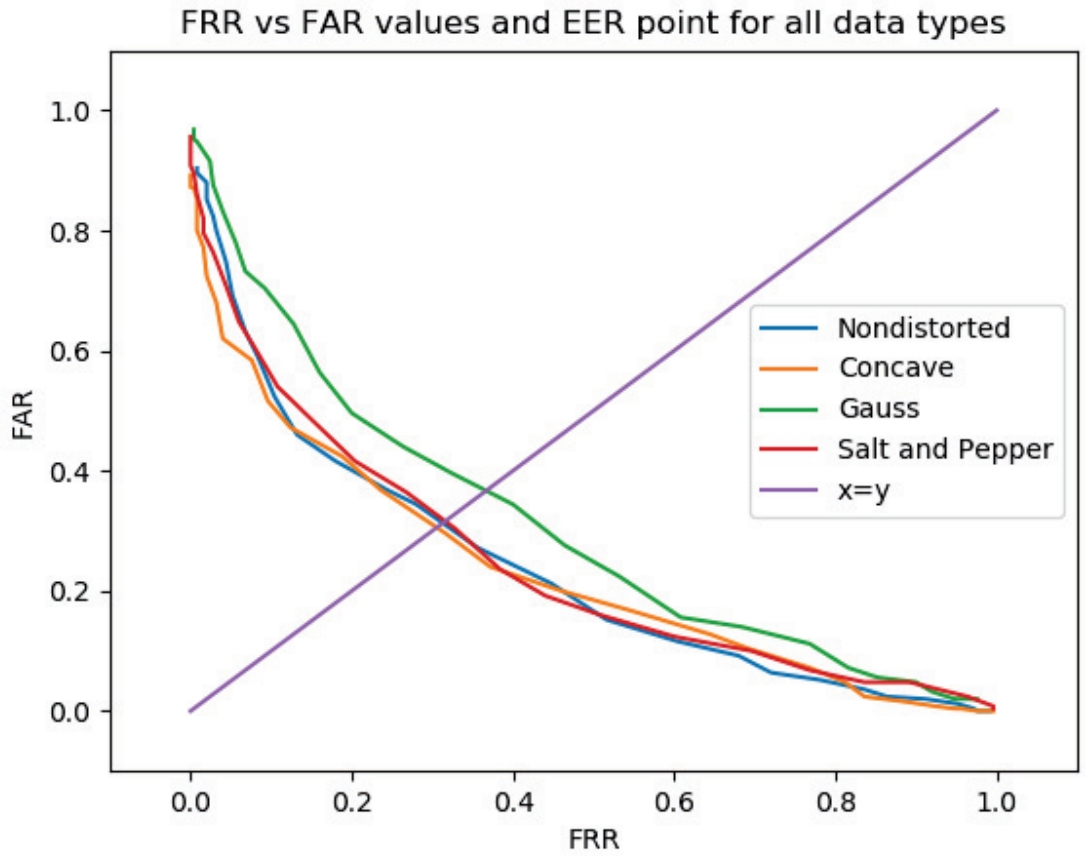

Figure 4.3. ROC Curves of all datasets for ORB with ANN

sets of SIFT with ANN algorithm is demonstrated in Figure 4.5.

4.4. Results with Deep Learning Approach

Performance results for Deep Learning Approach are obtained from both original and distorted images. According to the threshold values for every groups, the test results for with and without distortion datasets are written in the Table 4.5.

The end-to-end CNN algorithm has similar results with the algorithms using SIFT as a descriptor. ROC curves for nondistorted and distorted sets of CNN algorithm is demonstrated in Figure 4.6.

Table 4.3. Sift Descriptor with Brute Force Matcher

Sift with BF-Matcher Results			
Datasets		Mean	Variance
Non-Distorted Dataset	FAR	0.01	0.01
	FRR	0.05	0.03
	HTER	0.03	0.01
Gaussian Noise	FAR	0	0
	FRR	0.46	0.05
	HTER	0.23	0.03
Concave Noise	FAR	0	0
	FRR	0.09	0.04
	HTER	0.04	0.02
Salt and Pepper Noise	FAR	0	0
	FRR	0.35	0.34
	HTER	0.17	0.17

4.5. Comparing Test Results

This section covers the results of all algorithms by looking at the mean and variance values of all group of datasets. The mean and variance of acquired FRR, FAR and HTER results for non-distorted dataset are given in the Table 4.6.

Figures 4.7 shows ROC curves of all algorithms for nondistorted dataset.

The results show that, feature-based algorithm SIFT with BF matcher has slightly better HTER values for non-distorted dataset. The best accuracy is obtained with the algorithm containing SIFT and BF methods with 1% FAR, 4% FRR and 3% in HTER. If we look at the variance values, again, feature-based SIFT with BF matcher give more stable results than others.

For all groups with gaussian noise, mean and variance of acquired FRR, FAR and HTER results are given in the Table 4.7.

Figures 4.8 shows ROC curves of all algorithms for Gaussian noisy dataset.

The results show that as a best result, CNN approach gives 13% in HTER value. While the FAR value is 8%, the FRR value is 18%. The second best algorithm is the feature-based SIFT with BF matcher method with a 19% error rate. By looking at the

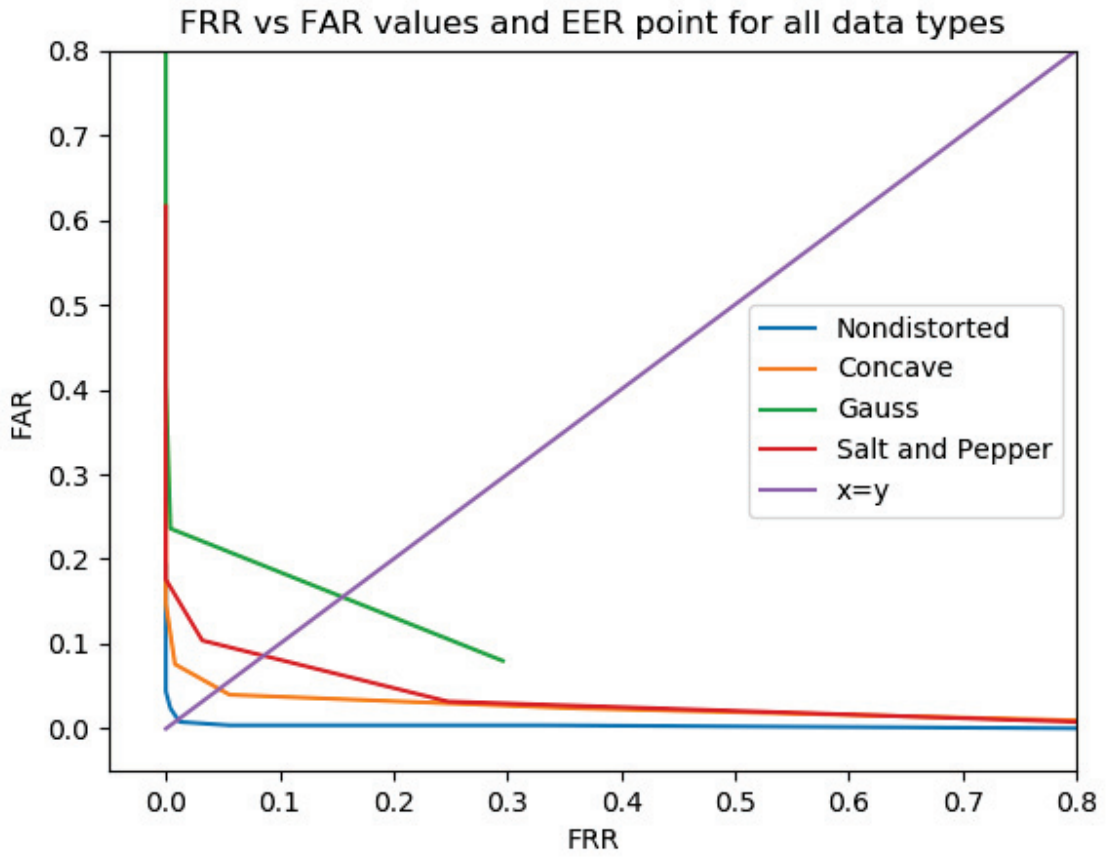

Figure 4.4. ROC Curves of all datasets for SIFT with BF-Matcher

variance values of algorithms in Gaussian noisy distorted dataset, the CNN and feature-based SIFT with BF matcher has similar values. These two algorithms are more reliable for Gaussian noisy datasets rather than the other algorithms. The less variance values of these algorithms shows the robustness and reliabilities of these algorithms against Gaussian noise type.

For all groups with concave distortion, mean and variance of acquired FRR, FAR and HTER results are given in the Table 4.8.

Figures 4.9 shows ROC curves of all algorithms for concaved dataset.

The results show that, feature-based SIFT with BF matcher approach gives slightly better value in HTER by 6%. CNN method has also close results in HTER values by 8%. By looking at the variance values of algorithms in concave distorted dataset, the feature-based SIFT with BF matcher has best value. These mean and variance values make this algorithm best for concaved distorted datasets.

Table 4.4. Sift Descriptor with Approximate Nearest Neighbor Approaches

Sift With ANN results			
Datasets		Mean	Variance
Non-Distorted Dataset	FAR	0.06	0
	FRR	0.09	0.03
	HTER	0.07	0.02
Gaussian Noise	FAR	0.02	0.03
	FRR	0.59	0.65
	HTER	0.30	0.31
Concave Noise	FAR	0.05	0.03
	FRR	0.14	0.03
	HTER	0.09	0
Salt and Pepper Noise	FAR	0.15	0.23
	FRR	0.25	0.29
	HTER	0.12	0.1

For the last type of distortion, salt and pepper noise, mean and variance of acquired FRR, FAR and HTER results are given in the Table 4.9.

Figures 4.10 shows ROC curves of all algorithms for Salt and Pepper noisy dataset.

The results show that, feature-based SIFT with BF matcher approach gives lowest error rate in HTER by 13%. CNN method and feature-based SIFT with ANN have also close results in HTER values by 18%. By looking at the variance values of algorithms, the CNN method has best value.

Table 4.10 list the required times for a single test comparison test for all algorithms.

The feature-based verification algorithm and end-to-end CNN model output results are almost the same for both models. However, CNN model turns the processing work of graphical card to the advantage, CNN model has advantages over the time to take a verification test. While a test for CNN model takes only 0.14 seconds, the feature-based methods, which gives approximate results, require more time in general.

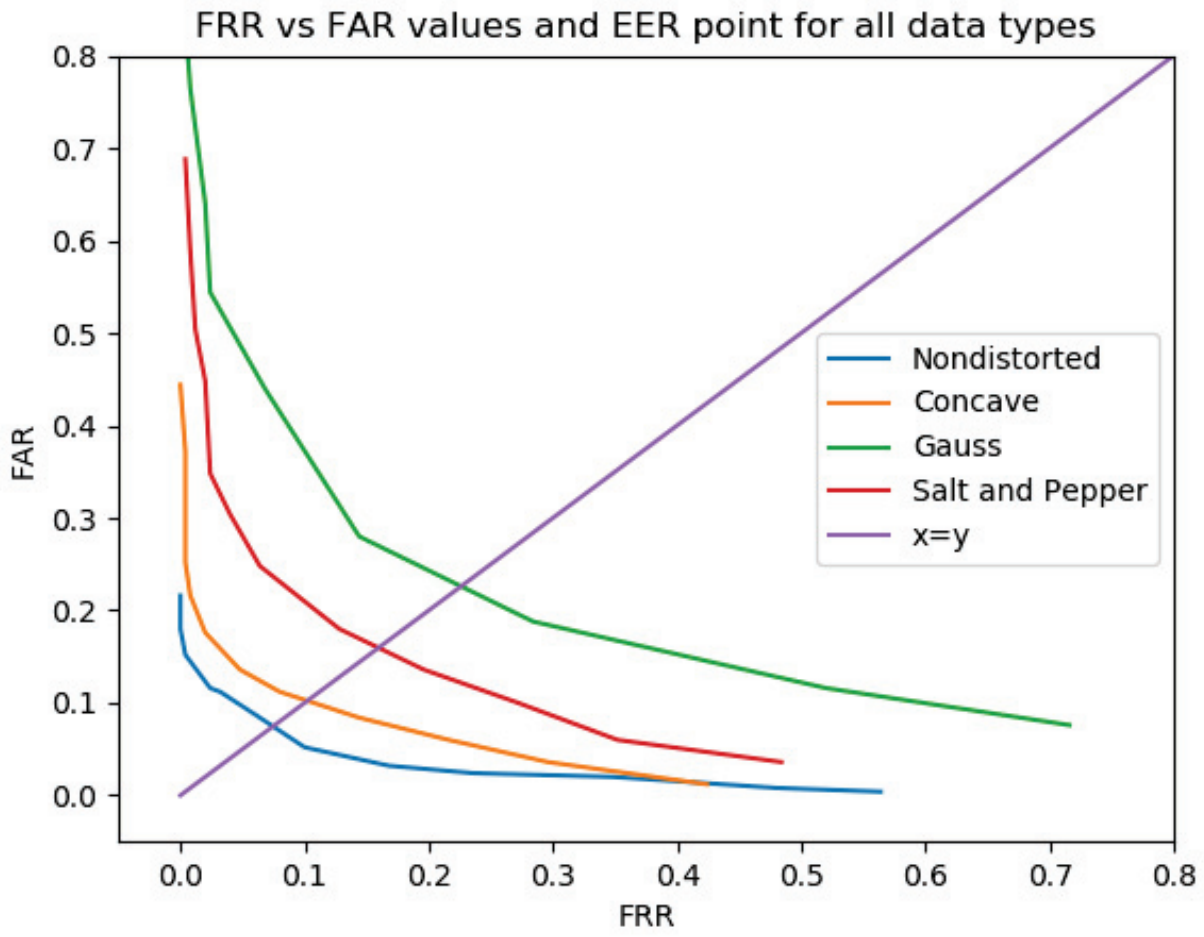

Figure 4.5. ROC Curves of all datasets for SIFT with ANN

If we examine the feature extracting and matching algorithms, we can say that ORB works faster than SIFT algorithm. And Brute Force matcher works faster than ANN algorithm.

To sum up all results, while looking at the FRR, FAR and HTER results, it is seen that for all noise types, accuracy is worse than the results of non-distorted images. The feature-based algorithm with SIFT and BF matcher has the best mean and variance results for non-distorted dataset. We can say that the CNN model is more robust and reliable to external effects and still preferable due to its robustness and better results on accuracies.

Table 4.5. Results for CNN approach

CNN results			
Datasets		Mean	Variance
Non-Distorted Dataset	FAR	0.03	0.05
	FRR	0.07	0.06
	HTER	0.04	0.01
Gaussian Noise	FAR	0.04	0.10
	FRR	0.26	0.31
	HTER	0.15	0.12
Concave Noise	FAR	0.01	0.0123
	FRR	0.10	0.08
	HTER	0.04	0.02
Salt and Pepper Noise	FAR	0.13	0.17
	FRR	0.13	0.08
	HTER	0.16	0.09

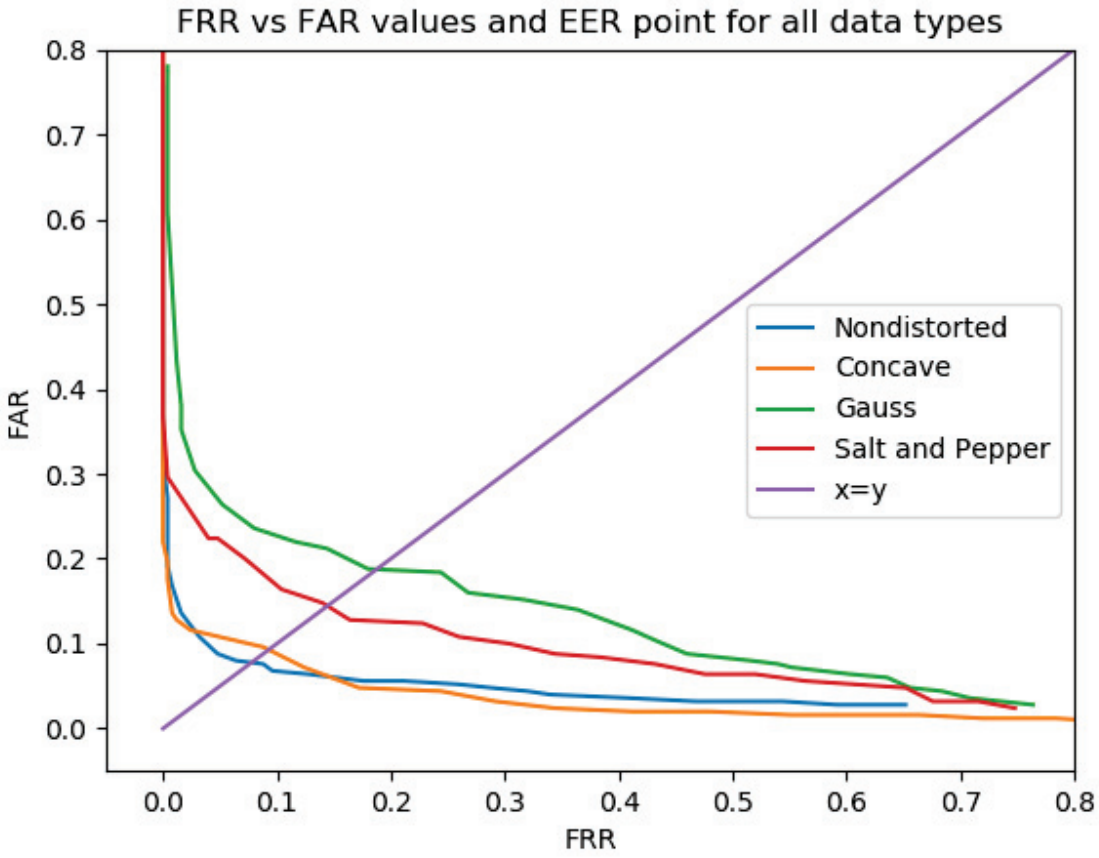

Figure 4.6. ROC Curves of all datasets for CNN Algorithm

Table 4.6. Test Results for both Feature-Based Method and CNN Method with Non-Distorted images

Mean and Variance Results for Non-Distorted Dataset			
		Mean	Variance
Feature-Based ORB with BF Matcher Method	FAR	0.28	0.05
	FRR	0.29	0.04
	HTER	0.28	0.2
Feature-Based ORB with ANN Method	FAR	0.29	0.06
	FRR	0.38	0.47
	HTER	0.31	0.23
Feature-Based SIFT with BF Matcher Method	FAR	0.01	0.01
	FRR	0.04	0.04
	HTER	0.03	0.01
Feature-Based SIFT with ANN Method	FAR	0.06	0.02
	FRR	0.09	0.05
	HTER	0.07	0.04
CNN Method	FAR	0.06	0.11
	FRR	0.08	0.07
	HTER	0.07	0.1

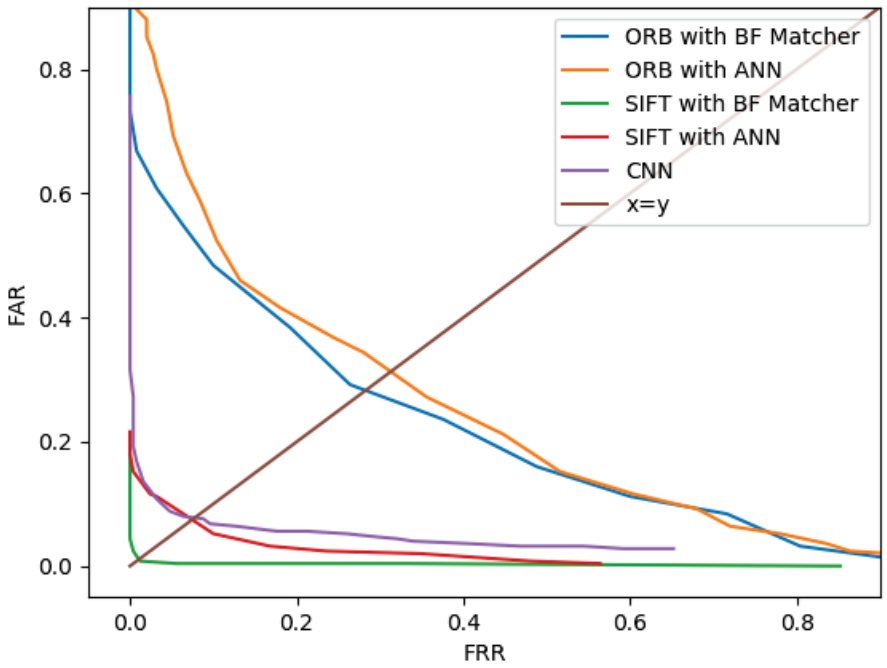

Figure 4.7. ROC Curves of all algorithms for nondistorted dataset

Table 4.7. Test Results for both Feature-Based Method and CNN Method with Distorted (Gaussian noisy) images

Mean and Variance Results for Gaussian noisy Dataset			
		Mean	Variance
Feature-Based ORB with BF Matcher Method	FAR	0.14	0.16
	FRR	0.49	0.14
	HTER	0.31	0.12
Feature-Based ORB with ANN Method	FAR	0.24	0.07
	FRR	0.38	0.47
	HTER	0.31	0.23
Feature-Based SIFT with BF Matcher Method	FAR	0.02	0.08
	FRR	0.44	0.07
	HTER	0.19	0.14
Feature-Based SIFT with ANN Method	FAR	0.01	0.04
	FRR	0.72	0.73
	HTER	0.35	0.34
CNN Method	FAR	0.08	0.19
	FRR	0.18	0.37
	HTER	0.13	0.17

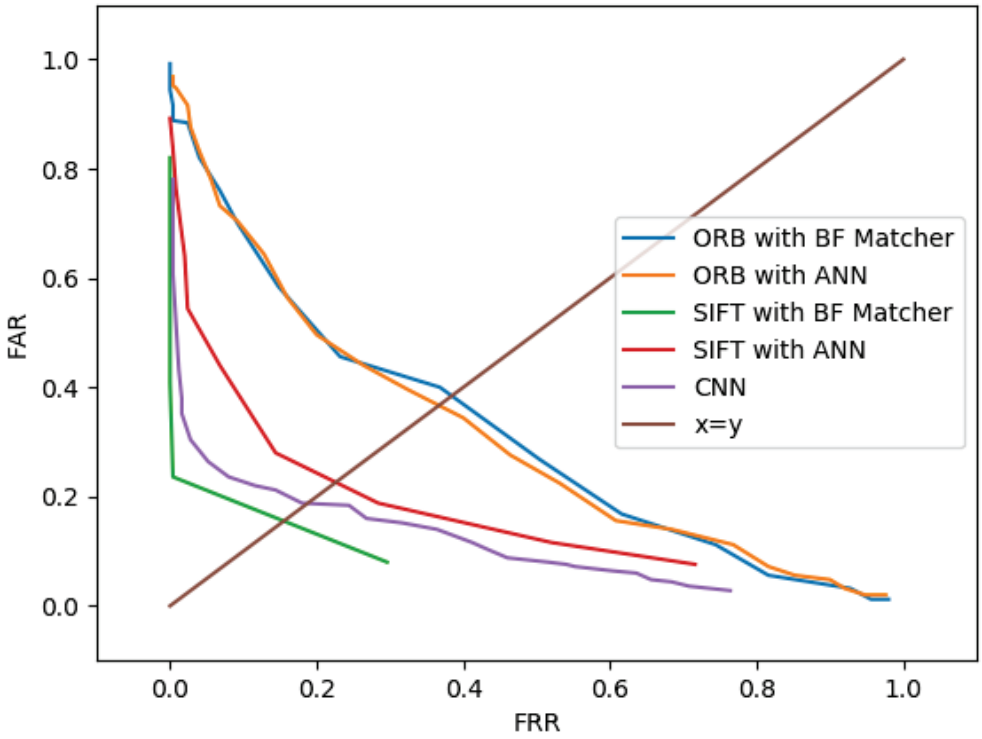

Figure 4.8. ROC Curves of all algorithms for Gaussian noisy dataset

Table 4.8. Test Results for both Feature-Based Method and CNN Method with Distorted (Concaved) images

Mean and Variance Results for Concaved Dataset			
		Mean	Variance
Feature-Based ORB with BF Matcher Method	FAR	0.29	0.15
	FRR	0.25	0.04
	HTER	0.27	0.07
Feature-Based ORB with ANN Method	FAR	0.32	0.10
	FRR	0.27	0.07
	HTER	0.29	0.07
Feature-Based SIFT with BF Matcher Method	FAR	0.03	0.12
	FRR	0.08	0.06
	HTER	0.06	0.03
Feature-Based SIFT with ANN Method	FAR	0.12	0.31
	FRR	0.16	0.06
	HTER	0.14	0.08
CNN Method	FAR	0.05	0.14
	FRR	0.1	0.09
	HTER	0.08	0.11

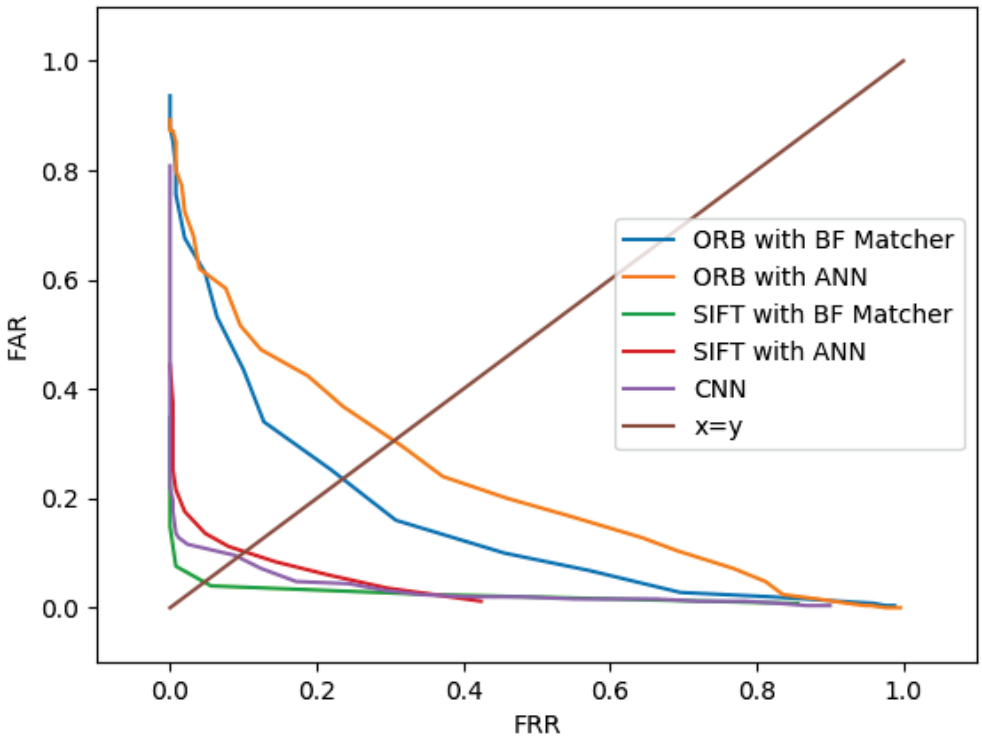

Figure 4.9. ROC Curves of all algorithms for concaved dataset

Table 4.9. Test Results for both Feature-Based Method and CNN Method with Distorted (Salt and Pepper noisy) images

Mean and Variance Results for Salt and Pepper noisy Dataset			
		Mean	Variance
Feature-Based ORB with BF Matcher Method	FAR	0.2	0.15
	FRR	0.36	0.37
	HTER	0.28	0.26
Feature-Based ORB with ANN Method	FAR	0.21	0.14
	FRR	0.01	0.02
	HTER	0.31	0.53
Feature-Based SIFT with BF Matcher Method	FAR	0.01	0.01
	FRR	0.26	0.41
	HTER	0.13	0.20
Feature-Based SIFT with ANN Method	FAR	0.13	0.24
	FRR	0.21	0.34
	HTER	0.18	0.24
CNN Method	FAR	0.16	0.18
	FRR	0.1	0.12
	HTER	0.18	0.1

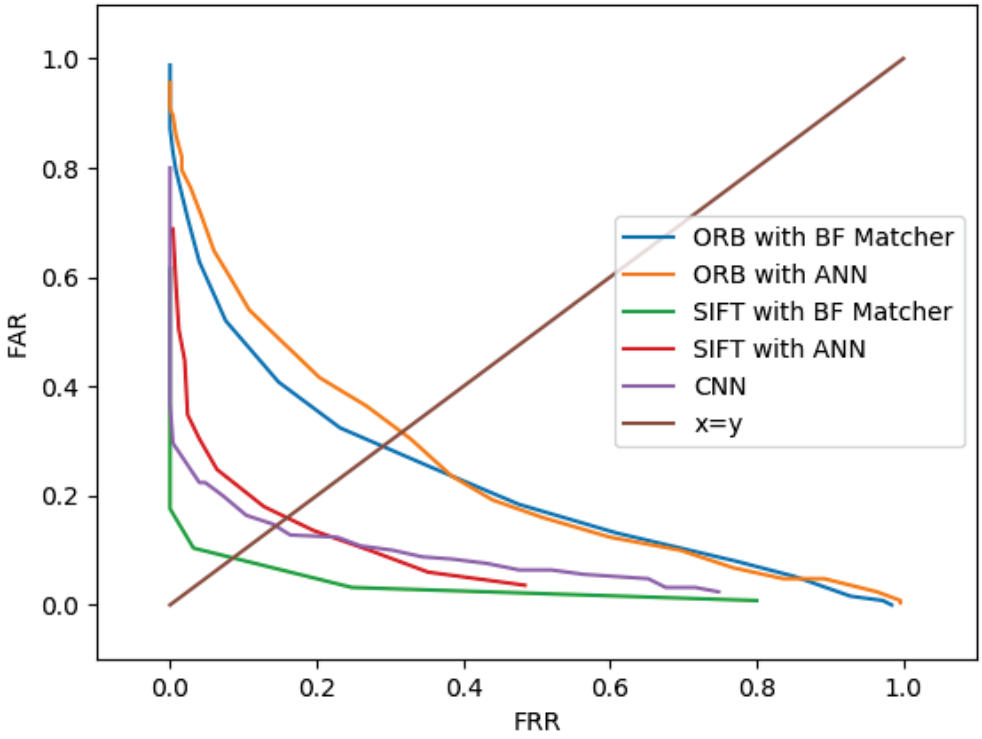

Figure 4.10. ROC Curves of all algorithms for Salt and Pepper noisy dataset

Table 4.10. Required time for testing a image for both models

Model Name	**Feature Extracting**	**Feature Matching**	**Total time**
Feature-Based ORB with BF Matcher	0.005 s	0.002 s	0.01 s
Feature-Based ORB with ANN	0.005 s	0.007 s	0.02 s
Feature-Based SIFT with BF Matcher	0.09s	0.08 s	0.28 s
Feature-Based SIFT with ANN	0.09 s	0.32 s	0.5s
End-to-end CNN Model	-	-	0.14 s

CHAPTER 5

CONCLUSION

Since the need for identity verification is increasing day by day, the robustness of one of the most used features of biometry that is fingerprint recognition has a critical importance.

This thesis mainly focuses on the verification of human identity with a fingerprint. To do this, two different approaches for fingerprint verification are taken into consideration. The first one is the feature-based method, based on basic and advanced image processing techniques, uses ORB and SIFT algorithms as feature detectors and Brute Force and ANN methods as feature matcher. The second approach is the popular end-to-end convolutional neural network-based model. These models are implemented and evaluated with the MCYT-100 dataset. For the training of the CNN model, fingerprint images are augmented with random zoom and orientation values. For testing the robustness of the algorithms against the synthetic distortions, three types of distortions are applied to generate synthetically distorted fingerprint images.

Performance results of the algorithms are obtained in four different scenarios that are without distortion and with three different distortion types. The performance evaluations show that feature-based methods with the SIFT algorithm give better results than the ORB algorithm. The feature-based algorithm with SIFT and BF matcher and end-to-end CNN method became prominent for all types of datasets and gave better results. For the robustness and consistency of the algorithms, CNN model is better to external noises due to its low variations for different types of dataset results. Besides, this study has shown that significantly less time is required for the ORB descriptor algorithms. And low testing duration of end-to-end CNN model makes it greatly suitable for real-time applications.

REFERENCES

Adán, M., A. Adán, A. S. Vázquez, and R. Torres (2008, April). Biometric verification/identification based on hands natural layout. *Image Vision Comput. 26*(4), 451–465.

Aguilar, G., G. Sanchez, K. Toscano, M. Salinas, M. Nakano, and H. Perez (2007, July). Fingerprint recognition. In *Second International Conference on Internet Monitoring and Protection (ICIMP 2007)*, pp. 32–32.

Alonso-Fernandez, F., J. Fierrez, J. Ortega-Garcia, J. Gonzalez-Rodriguez, H. Fronthaler, K. Kollreider, and J. Bigun (2007, Dec). A comparative study of fingerprint image-quality estimation methods. *IEEE Transactions on Information Forensics and Security 2*(4), 734–743.

Chollet, F. et al. (2015). Keras. https://keras.io.

Duchi, J., E. Hazan, and Y. Singer (2011, July). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res. 12*, 2121–2159.

Elmouhtadi, M., S. E. fkihi, and D. Aboutajdine (2018). *Fingerprint Identification Using Hierarchical Matching and Topological Structures*, pp. 393–408. Cham: Springer International Publishing.

Grother, P. and E. Tabassi (2007, April). Performance of biometric quality measures. *IEEE Trans. Pattern Anal. Mach. Intell. 29*(4), 531–543.

Huang, G., Z. Liu, L. v. d. Maaten, and K. Q. Weinberger (2017, July). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269.

Jain, A. K., S. Prabhakar, L. Hong, and S. Pankanti (2000, May). Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing 9*(5), 846–859.

Jain, A. K., A. Ross, and S. Prabhakar (2004, Jan). An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology 14*(1),

4–20.

Lin Hong, Yifei Wan, and A. Jain (1998, Aug). Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*(8), 777–789.

Liu, Y., B. Zhou, C. Han, T. Guo, and J. Qin (2018, 10). A method for singular points detection based on faster-rcnn. *Applied Sciences 8*, 1853.

Lu Jiang, Tong Zhao, Chaochao Bai, A. Yong, and Min Wu (2016, July). A direct fingerprint minutiae extraction approach based on convolutional neural networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 571–578.

M. Bolle, R., J. Connell, and N. Ratha (2002, 12). Biometric perils and patches. *Pattern Recognition 35*, 2727–2738.

Maltoni, D., D. Maio, A. K. Jain, and S. Prabhakar (2009). *Handbook of Fingerprint Recognition* (2nd ed.). Springer Publishing Company, Incorporated.

MathWorks (2019). *What Is a Convolutional Neural Network?*

Ortega-Garcia, J., J. Fierrez, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. HernÃ¡ez, J. Igarza, C. Vivaracho-Pascual, D. Escudero, and Q. Moro-Sancho (2003, 12). Mcyt baseline corpus: a bimodal biometric database. iee proc vis image signal process spec issue biom internet. *IEE Proceedings - Vision Image and Signal Processing*, 395 – 401.

Ortega-Garcia, J., J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J. . Igarza, C. Vivaracho, D. Escudero, and Q. . Moro (2003, Dec). Mcyt baseline corpus: a bimodal biometric database. *IEE Proceedings - Vision, Image and Signal Processing 150*(6), 395–401.

Pato, J. N. and L. I. Millett (2010). Biometric recognition: Challenges and opportunities.

Peralta, D., I. Triguero, S. García, Y. Saeys, J. M. Benítez, and F. Herrera (2017). Fingerprint classification with a new deep neural network model: robustness for different captures of the same fingerprints. *CoRR abs/1703.07270*.

Recogtech, A. B. I. (2019). 5 common biometric techniques compared.

Ruxin Wang, Congying Han, and Tiande Guo (2016, Dec). A novel fingerprint classification method based on deep learning. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 931–936.

Saleh, A. M. A. (2014). Enhanced secure algorithm for fingerprint recognition. *CoRR abs/1402.4936*.

Solutions, B. (2016). Fingerprint recognition.

team, O. (2000). Open source computer vision library. *open source computer vision and machine learning software library*.

Wang Yani, Wu Zhendong, Zhang Jianwu, and Chen Hongli (2016, Oct). A robust damaged fingerprint identification algorithm based on deep learning. In *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1048–1052.

Wayman, J. (2001, 01). Fundamentals of biometric authentication technologies. *Int. J. Image Graphics 1*, 93–113.

Zhu, Y., X. Yin, and J. Hu (2018). Robust fingerprint matching based on convolutional neural networks. In J. Hu, I. Khalil, Z. Tari, and S. Wen (Eds.), *Mobile Networks and Management*, Cham, pp. 56–65. Springer International Publishing.