RÉPUBLIQUE DU CAMEROUN
PAIX-TRAVAIL-PATRIE
****
UNIVERSITÉ DE DSCHANG
****
ÉCOLE DOCTORALE

REPUBLIC OF CAMEROON
PEACE-WORK-FATHERLAND
****
UNIVERSITY OF DSCHANG
****
POSTGRADUATE SCHOOL

DSCHANG SCHOOL OF SCIENCES AND TECHNOLOGY

Fundamental Computer Science, Engineering and Applications Research Unit (*URIFIA*)

RESEARCH TOPIC:

A FINGERPRINT RECOGNITION SYSTEM BASED ON MACHINE LEARNING

Publicly defended thesis with a view of obtaining a diploma of
Masters in Computer Science

OPTION : **FUNDAMENTAL COMPUTER SCIENCE**
SPECIALITY : **NETWORKS AND DISTRIBUTED SERVICES**

BY:

YATAGHA ROMARICK
REGISTRATION NUMBER : CM-UDS-19SCI2681
BSc in Combined Mathematics & Computer Science

UNDER THE SUPERVISION OF :

Dr SOH MATHURIN
Senior Lecturer

Academic Year: 2020/2021

Dedication

To my father YATAGHA GEREMIE *and my mother* MAKENGUE BERTHE.

Acknowledgements

The completion of this dissertation would not have been possible without the support and nurturing of the Almighty God, who granted and continues to grant me health and wisdom, without which I wouldn't have come this far.

Throughout the writing of this dissertation, I have received enormous support and assistance:

- I would like to express my deepest acknowledgements to the jury, for the time and energy taken to appreciate this work.
- I would also like to extend my deepest gratitude to my supervisor Doctor SOH Mathurin, whose expertise was unavoidable in structuring the research questions and methodology. Your corrections and constructive criticism pushed me to sharpen my thinking and brought this work to the level it has.
- I very much appreciate my tutors and mentors Professor TAYOU Clementin and Doctor TCHOUPE Maurice for their valuable guidance and counseling throughout my studies. You provided me with the tools that I needed to choose the right direction and successfully complete my curriculum.
- Special thanks to Mr. Utkarsh-Deshmukh, MIT license and developer, for his advice, proposed methods and libraries.
- Special thanks to all teachers of the mathematics and computer science department of the university for the lessons taught. You all are a source of inspiration.
- My appreciation also goes out to my parents, Mr. YATAGHA Geremie and Mrs MAKENGUE Berthe for their encouragements and support throughout all my studies. I am deeply indebted to them.

- Many thanks to all my family members, my uncles, my siblings for their unwavering support and guidance
- I would also like to thank all my classmates for the constructive discussions and time spent together
- Finally, thanks to all those who participated, contributed and took part in the study and enabled this research to be possible.

Table of Contents

List of Acronyms

Acronyms	**Meaning**
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DL	Deep Learning
DPI	Dot Per Inch
FAR	False Acceptance Rate
FRR	False Rejection Rate
FVC	Fingerprint Verification competition
HTER	Half Total Error Rate
K-NN	K^{th} Nearest Neighbor
LR	Learning-Rate
NIST	National Institute of Standard Technology
RAM	Random Access Memory
ReLU	Rectified Linear Unit
ResNet	Residual Network
RF	Random Forest
SGD	Stochastic Gradient Descent
SQL	Structured Query Language
SVM	Support Vector Machine

List of Tables

List of Figures

Abstract

Biometric traits are used in security and authentication systems, as they confidently identify users. These traits are classified into two categories, the behavioral and the psychological biometric traits. The most used of these traits is Fingerprint, which is a psychological traits and makes a trending research topic. The process of authentication is time consuming and is best appreciated with real time response. It is shown that the average authentication time with a fingerprint based authentication system is about 1.4 seconds on a Sun ULTRA I workstation. In this project, we propose a fingerprint authentication system which uses a deep Convolutional Neural Network (CNN), to improve the overall authentication time. To do so, we start with some image enhancement and preprocessing techniques, within which fingerprint images are filtered, binarized and skeletonized. The resulting images are fed into a deep Convolutional Neural Network for classification and Minutiae feature extraction, which are used as feature vectors for the fingerprint template, before the matching process. With a 98% accuracy, we train the algorithm to correctly classify fingerprint images within a relatively small time frame (As compared with other techniques cited in the state of art), with a good and better overall authentication time: An average of 0.3 seconds per authentication, with a False Acceptance Rate of 0% and a False Rejection Rate of 12%. This helps a long way, in authenticating a large number of individuals within a small time frame

Keywords: Fingerprint, Deep Learning, Convolutional Neural Network, Minutiae, Authentication.

Systeme de reconnaissance d'empreinte digitale base sur le Machine Learning

Résumé

Les caractères biométriques sont utilisés lors de l'authentification dans les systèmes de sécurité, car ils identifient les utilisateurs de façon unique. Ces caractères sont classés en deux catégories, les caractères biométriques comportementaux et psychologiques. Le caractère le plus utilisé pour l'authentification est l'empreinte digitale, caractère psychologique, faisant l'objet de vives discussions au seing de la communauté scientifique. Le processus d'authentification prend du temps et est mieux apprécié lorsque les transactions se font en temps réel. Il est démontré que le temps d'authentification moyen avec un système d'authentification basé sur les empreintes digitales est d'environ 1,4 seconde sur une station de travail Sun ULTRA I. Dans ce projet, nous proposons un système d'authentification par empreintes digitales qui utilise un réseau neuronal convolutif profond, pour améliorer le temps d'authentification global. Pour ce faire, nous commençons par quelques techniques d'amélioration et de prétraitement d'images, dans lesquelles les images d'empreintes digitales sont filtrées, binarisées et squelettisées. Les images résultantes sont introduites dans un réseau de neurones convolutifs, pour la classification et l'extraction de caractéristiques (Minutiae), qui sont utilisées comme vecteurs caractéristiques d'empreinte digitale, avant le processus de vérification. Avec une précision de 98%, nous entraînons l'algorithme à classifier correctement les empreintes digitales dans un court laps de temps (par rapport aux autres techniques citées dans l'état de l'art), avec un meilleur temps d'authentification global: En moyenne 0.3 secondes par authentification, avec des taux de fausses acceptations et de faux rejets de 0% et de 12% respectivement. Ce qui est une grande aide lors de l'authentification d'un grand nombre d'individus dans un court laps de temps

Mots clés: Empreintes digitales, Apprentissage profond, Réseau de neurones convolutifs, Minutiaes, Authentification

Introduction

Contents

Artificial Intelligence (AI), is a branch of computer science that aims to equip software with the ability of analyzing its environment using either predetermined rules and search algorithms, or pattern recognizing machine learning models, and then make decisions based on those analysis [51]. This discipline can be further splitted and divided into different sub disciplines, based on the type of problem they tackle, and the methods they use to approach and analyze the problem. In this lining, we have the following areas of study of Artificial Intelligence: Machine Learning, Computer Vision, Deep Learning, Natural Language Processing...
Among the above mentioned domain of study of AI, the most prominent domain, which is going to be the heart of our discussion in this dissertation is Deep Machine Learning
Machine Learning teaches a machine how to make inferences and decisions based on past experience. It identifies patterns, analyses past data to infer the meaning of these data points to reach a possible conclusion without having to involve any human expertise [25]. Machine learning algorithms are often categorized as supervised or unsupervised:

- **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The algorithm is able to provide targets for any new input after sufficient training.

- In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data.
  Also, there is the **Semi-supervised machine learning algorithms** which falls somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training, and the **Reinforcement machine learning algorithms,** which allows the system to interact with its environment by producing actions and discovering errors and rewards.

Context of Study

The digital 21st Century in which we live in today is an era in which everything is digitalized, and security and privacy is becoming a major issue of great concern. Having understood this, Industries and Firms are investing important capital so as to be able to guarantee the security and privacy of their workers and users of their products. Also, they need at any given moment, to be able to trace and say with precision who had access to which service, when and where. It is as a response to this need that Artificial Intelligence and Machine Learning Techniques are greatly being implemented in the field of user authentication.
More to this, the traditional way of authenticating users using passwords and PINs is becoming obsolete, not only because of the ease with which intruders are now getting this information of PINs and passwords, but because users themselves forget these informations. To further portray this point of the obsoleteness of passwords and PIN's in authenticating users, is the Digital Consumer survey done in 2015, that clearly shows that, of 24000 people interviewed, close to 77% says passwords are cumbersome, and are willing to look at

some other alternatives [25]. Users are now authenticated using specific characters, which are particular and unique to each individual, such as fingerprint pattern, and Artificial Intelligence appears the ideal way of handling this biometric information which is subject to change as time goes on [51]

Problem Statement

Security is a major concern not only for organizations and industries, but also for citizens living in this interconnected world of ours. For this reason, devices are built with embedded captors and sensors, for biometric authentication. Fingerprint is one of the most dominant biometric traits that keeps spreading out because of its uniqueness, acceptability, and low cost [14]. Given the vast number of people that needs authentication within a small timeframe, biometric recognition systems are being developped and needs improvement in rapidity, efficiency and robustness. Particularly, in the field of Fingerprint Recognition, Artificial Intelligence methods are currently being implemented with image processing techniques to better the productivity of these systems; it is with respect to this that research articles are widely published.

To Solve the problem of securing our devices (smartphones, computers...), the problem of authenticating users and workers of industries, so as to grant access to authorized personnel only, many techniques have been used so far, among which the use of passcode and PINs to identify users. This technique of passwords is tedious, as good passwords, which are at the same time very complex to remember, are easily forgotten by users. Also, hackers could get the passwords of some users, and pretend to be these users in an identity theft. Biometric information is used to solve this problem of passwords, as the biometry of an individual is unique to him, and he cannot forget it as he carries it along with him each time (biometry is what you are). The problem at hand now is to build a complete authentication system in which one does not need any passcode or pass card, but using only biometric information: Fingerprints in our case. The Use of Artificial Intelligence in conceiving biometric authentication systems is from all indication, an interesting solution to better ensure security, rapidity and efficiency. AI is mostly used to distinguish between Human proposed model and machine-generated models. Intelligent algorithms can help accelerate the process of authentication.

For this, we propose a complete biometric authentication system with a Learning algorithm, that classifies each fingerprint pattern as belonging to one of three fingerprint classes as discussed in chapter. This significantly reduces the number of comparisons that must be done in the database before a match is found, or the fingerprint is rejected; Hence, reducing the time required to authenticate one individual, which is a key parameter in every authentication system

Methodology and Plan of Work

Methodology

In order to solve the above stated problem of providing a complete fingerprint biometric authentication system and accelerating the process of authentication, a thorough study of the existing literature in the field of fingerprint recognition systems is obliged. Also, Machine learning techniques such as Neural Networks and Deep Learning must be mastered.

In our methodology, we start with observation and pattern identifications, continuing with the proposition of our theory or model, and concluding with suggestions of new theories and ways forward, after giving the results obtained from the proposed model. This because our research project will only grow from already set foundations, which are standard existing techniques such as Median filters for image preprocessing, and defined libraries for minutiae extraction.

We start with some image enhancement and preprocessing techniques, within which fingerprint images are filtered, binarized and skeletonized. The resulting images are fed into a deep Convolutional Neural Network for classification and Minutiae feature extraction, which are used as feature vectors for the fingerprint template, before the matching process. For the matching process, the euclidean distance between minutiae points is used as coordinates for the feature vector, and the orientation angle of the delta point is used as rotational invariance reference.

Plan of Work

This present dissertation is structured as follows:

chapter 1: Literature Review. In this chapter, we present some definitions of concepts of Artificial Intelligence, and a brief history of the evolution of biometric recognition. We also present the state of art and some interesting works that have been published in the field of fingerprint recognition with AI techniques, on which we lay foundations of this project, presenting their results, accomplishments and limits.

chapter 2: Implementation of a Convolutional Network. In this chapter, we will talk of the various methods we use to get our results. Definition of terminologies and explanation of some basics is also presented in this chapter, including the architecture of our proposed system.

chapter 3: Results and Discussion. Experimental results and discussions are presented in this chapter. A comparative study of our results and some other results presented in the state of art is also given here. The dataset and conditions in which we carry out the experimental results are also provided in this chapter.

Finally, the general conclusion highlights the difficulties met in the couse of writing this dissertation, along with the perspectives for further research

I

Chapter

Literature review on fingerprint recognition algorithms

Contents

I.1. Introduction

Biometric traits are intensively being used by industries and firms to identify and authenticate users. This is because biometric identification is based on what the user is (fingerprint, iris pattern. . .), than what the user knows (passwords, PIN. . .), which can be easily hacked. Some of the biometric traits used in identification and authentication are fingerprint, iris pattern, voice tonality. . . Among all these traits, fingerprint recognition is the most used biometric identification system, as the survey "the Future of Biometrics"published by The Acuity Market Intelligence United State shows that biometric trait covers up to 73% of all the biometric traits used in biometric authentication[13]. According to this same survey, not only is fingerprint the most used biometric trait, but it appears to be the most secured of all biometric traits used in biometric recognition, with the least Acceptance Rate and Rejection Rate, as shown in table I below

Methods	**FRR(%)**	**FAR(%)**
Fingerprint Recognizer	3-7	0.001-0.1
Face Recognizer	10-20	0.1-1
Iris Recognizer	2-10	$\geq$ 0.001
Palm Print Recognizer	1-2	1-2
Voice Recognizer	10-20	2-5

Table I – *Implication of error rate of biometric authentication systems*

Fingerprint is an impression or mark made on a surface by a person's fingertip, suitable to be used to identify individuals because of the unique pattern of lines and ridges on the fingertips; Their uniqueness and lasting quality makes them one of the best ways to identify a person. Fingerprint recognition is therefore the process of uniquely identifying and authenticating each individual, based on the print of his fingers. Though they are unique and hence suitable for authentication, fingerprint recognition is faced with obstacles, such as they can be collected from somewhere, and 3D-printed on artifacts, and used in identity theft attacks. Also, well trained algorithms can now generate bio-data, precise enough to be used in identity theft attacks.
In this chapter, we bring out some of the fingerprint recognition algorithms that have been developed and implemented, with their limits. But first, the history of fingerprint recognition systems.

I.2. History of Fingerprint Recognition using Artificial Intelligence and Machine Learning

While they might not be prehistoric, biometrics have been around for thousands of years. Throughout the last few millennia, biometrics have gone from rough methods of classification to being authenticators of identity using a wide range of modalities, among which Artificial Intelligence Techniques [3]
The first account of biometrics can be dated as far back as 500 BC.

Biometric recognition in general and fingerprint in particular was already used in the 1850's, though not coupled with Artificial Intelligence Techniques as illustrated in this piece of writing: "In 1858, Sir William Herschel, working for the Civil Service of India, recorded a handprint on the back of a contract for each worker to distinguish employees from other who might claim to be employees when payday arrived"[28]

It is only later in the 1969s that biometric recognition became semi automated. The first semi-automatic face recognition system was developed by Woodrow W. Bledsoe under contract with the US Government [28]. The wonders of Artificial Intelligence only got embedded in Biometrics in the last decades, making biometrics recognition algorithms extremely accurate. Some writing shows that about 1200 million electronic passports in circulation in the mid-2019 provide a huge opportunity to implement face recognition with Artificial Intelligence at international borders [17]. Artificial Intelligence and biometric Recognition is becoming more and more omniscient in our day to day life, in social media (Facebook, Instagram...), access to buildings and infrastructures.

A Recognition Algorithm (Fingerprint, Iris or Facial) is appreciated and criticized based on some three parameters, The False Acceptance (FAR), The False Rejection Rate (FRR) and the Half Total Error Rate (HTER) evaluated as follows [13]

$$FAR = \frac{Number\_of\_accepted\_impostors}{Number\_of\_impostors\_transaction}$$

$$FRR = \frac{Number\_of\_rejected\_clients}{Number\_of\_client\_transaction}$$

$$HTER = \frac{FAR + FRR}{2}$$

A good biometric recognition system is one in which these values are minimized. The Performances of fingerprint recognition Algorithms can also be evaluated in terms of processing time and recognition accuracy [21].

Some of these algorithms coupled with Machine Learning Techniques are detailed in the following sections

I.3. Neural Network and Image Classification Algorithms in Fingerprint Recognition

Neural network algorithms are a set of algorithms, modeled after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, and labeled or unlabeled input. A neural Network is composed of several nodes, which are places where computations are done. A node combines input from the data with a set of coefficients or weights, that assigns significance to inputs with regard to the task the algorithm is trying to learn. The products of the inputs and weights are summed, and this sum is passed through an activation function to determine to what extend input signal should affect the ultimate outcome [34]. A diagrammatic representation of a node is given in figure 1.

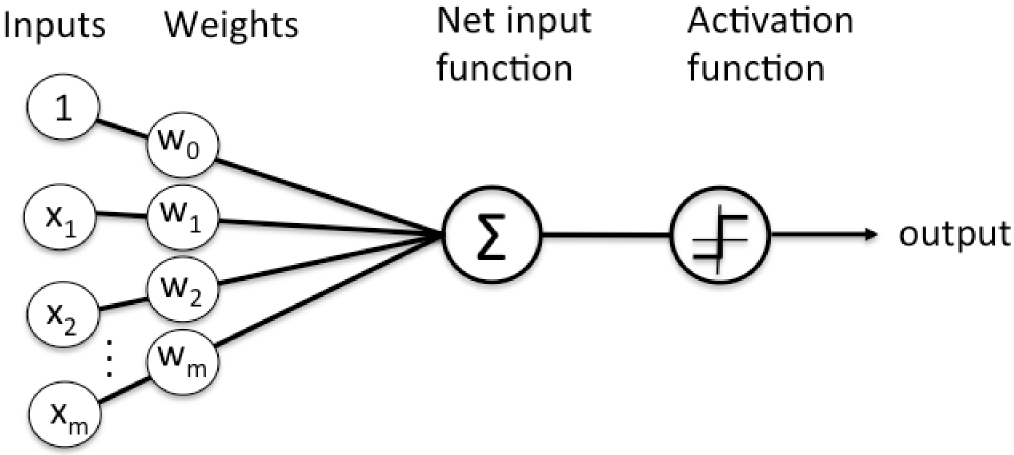


Figure 1 – *A diagrammatic representation of a node*

We talk of image classification when a computer can analyze an image and identify the “class”. A class is essentially a label, for instance, a “car”, or an “animal”[52]
Image Classification and enhancement is the most used technique

in fingerprint recognition, and several articles have been published, one adding building blocks to another, reshaping the philosophy of fingerprint recognition through image analysis, classification, enhancement and optimization techniques. The main advantage of image classification algorithms is their small response time during authentication. This is because during enrollment, the fingerprint images are collected and classified into groups and categories, which reduces the number of fingerprints to be compared during authentication, since the fingerprint we are looking for must belong to one and only one of the classes registered during enrollment: The system we propose in this project is based on this philosophy.

Fingerprint identification is one of the most well-known and publicized biometrics, because of their uniqueness and consistency over time. Fingerprints have been used for identification for over a century, and is improving in efficiency mainly because of advancements in computing capabilities and availability of data [29]. Machine learning and neural network algorithms have outclassed many previous techniques for object detection and classification [26].

Huong et Al. propose in [33] 4 machine learning classification algorithms to classify fingerprints into the three classes of fingerprints: Arch, Loop or Whorl. These algorithms are the Random Forest (R.F.), the Support Vector Machine (S.V.M), the k-Nearest Neighbor (k-NN) and Convolutional Neural Network (CNN).

In this work, after some preprocessing and noise removal such as Gaussian Filter and Edge Detection with Canny algorithm, Huong et Al. went ahead to feed their algorithms with datasets from three different sources, Fingerprint Verification Competition (F.V.C), National Institute of Standard Technology (NIST) and BG-IMG. The metrics they use to evaluate the performance of classifiers are Accuracy, Precision and Recall.

Results of their classifier shows that R.F. has the best accuracy (96%), that SVM and CNN models have the best accuracy (95%).

One of the major drawback of the work proposed here is the training time (About 7 Hours in average) and resource consumption.

In article [27] Pavol et Al. propose a fingerprint recognition system using artificial neural network as feature extractor, made of 3 classic stages: enhancement and coarse fingerprint classification, ex-

traction of features, and finally the Matching and comparison stage. For image enhancement and pre-processing, they use Gabor Filters, with local thresholds, so as to consider neighboring pixels when enhancing a particular pixel of the fingerprint image. For extraction, a simple Neural Network of 1 input layer, 200 hidden neurons and one output layer is used, and experiment is carried out on the FVC 2002 dataset. On this dataset, the recognition rate reached a value of 67% success. Tests were carried out on a local machine with Intel Core i7-4800 MQ at 2.7GHz and 16GB Ram, in contrast with the work proposed in [33] by Huong et Al. who used parallel computation framework on a 12 powerful computers in a cluster, to obtain their experimental results.
Pavlo Tertychnyi et al [36] used convolutional neural network for determining and identifying low quality fingerprint. In this approach, they consider five categories of low quality fingerprints. namely: dry, wet, damaged, dotted and blurred. This approach shows superiority over Support Vector Machine Classifier, with 84% accuracy.

In contrast with previous works, [27, 9], Su et Al. suggested a deep learning technique for pore extraction, a Level-3 Minutiae point in their article [49]. They use a convolutional neural network of nine weighted layers in which starting first seven layers are convolution layers, and the last two layers are fully connected layers. Fully connected two-way layers are used to prepare the features for the subsequent binary classification, which is done by a classifier. The algorithm is checked on 1480 fingerprint images, and achieves a pore detection rate of 88.6%, and an equal error rate (EER) of 3.66%.

Engelsma et Al proposed a technique not based on the traditional minutiae extraction and matching, but a deep network that learns for itself a fixed-length feature vector, that best identifies and differentiate fingerprints as they say in [23]. Their experimental results are carried out on the FVC DB4, and their results are compared with two commercial matchers Verifinger v6.3 and Innovatrics v2.03. This comparison shows that they have higher search accuracy and faster large scale search than the Verifinger. Also, their system is equiped with the homomorphic encryption, which is used to prevent and combat attacks. The relative disadvantage found in this art is the complexity of the deep network that is proposed, and its training time.

In [30] Shervin et Al. used a pre-trained Residual Neural Network, and Transfer Learning in Fingerprint recognition. This pre-trained ResNet was trained on ImageNet dataset. Shervin et Al. used this model, by applying several augmentation techniques to increase the number of training samples, including horizontal and vertical flip, random crops and small distortions [30]. This model is then fine-tuned for fix number of epoch and evaluated on a test set. Experimental results are carried out on the PolyU fingerprint database, provided by Hong Kong Polytechnic University containing 1480 images of fingerprints. Their results show that they outclass existing model such as the Scattering network [44] and the Gabor-Wavelet [19] in Accuracy. The complexity of the network used (50 Convolutional and pooling layers) is to be noted, with the resources used in training the algorithm before being used by Shervin and co-researchers (Scalable GPUs)

Rui Xie et Al in [41] proposed a neural network based on image quality estimator that takes gray scale fingerprint image as input and generates a continuous image quality score. This Artificial Neural Network is trained with the back-propagation algorithm. The proposed quality estimator divides the images into the blocks of size 16×16, which unfortunately increases the computational cost [21]. In [11], Atanu Chatterjee et al. proposed a backpropagation Neural Network technique for fingerprint identification. This technique uses thinned minutiae features for matching purposes. To remove false minutiae, post processing is also performed. This technique is tested on smaller size databases and achieves accuracy around 95%. But, it is to be noted that this post processing for minutiae removal is also time consuming, and adds the overall authentication time, which is a drawback. Stephane Kouamo et Al in [48] proposed an artificial neural network based fingerprint classifier for e-learning applications. Their idea is to apply a backpropagation algorithm to train the multilayer feedforward neural network for fingerprints. Hidden layers of the Neural Network are used to perform comparisons of the template by calculating probabilities, which makes system invariant to rotation and translation. The major drawback of their system is that training time is high.

I.4. Support Vector Machine in Fingerprint Recognition

Support Vector Machine is a supervised machine learning algorithm that is also used for classification and regression problems.
In [26], Sozan Abdulla proposed a three steps recognition system using support vector machine and neural network. The first step being to remove noise from the image which may be corrupt from fingerprint image capture, using binarization technique to convert grayscale to black and white image. the second being to extract minutiae on the harvested fingerprint image, and storing it in a database as template. And in the last step, the feed forward neural net and the support vector machine algorithm are trained on this information stored in the database.
To evaluate the performance of their algorithms, a dataset of 380 fingerprint images, sized 170 × 170 is used. The effectiveness of the proposed algorithm is shown by the improvement in fingerprint verification accuracy. The results in [26] shows that the recognition rates for training neural network are all approaching 100%. Recognition rate in percentage on testing data is about 92% for the neural network, and about 85% for the SVM.
In [42], S. Adebayo Daramola et al. suggested a verification system whereby a biometric input sensor is used to acquire digital images from human fingerprint, which is used in preprocessing stage. After the preprocessing stage, they use a SVM feature extraction algorithm to extract robust features from the preprocessed image. After this is the training process, where models are generated for each of the users, and finally the verification algorithm based on support vector machine used to decide if fingerprint matches or not. For experimental results, 100 fingerprint images are tested, among which 80 are correctly matched, while 20 images are mismatched, with a False Rejection Rate of 0.2.
Lianhua Liu et al in [24] proposed a support vector machine based fingerprint quality assessment method that uses five features which are extracted from an image. This method classifies the fingerprint

images into three classes: high, average and low quality, and achieved accuracy of 96.03%. In this method, feature extraction increases processing time. Fingerprint segmentation increases the preprocessing time.

I.5. Deep Learning and Convolutional Neural Network in Fingerprint Recognition

Deep learning networks are distinguished from the single-hidden-layer neural network by their depth [34]. It is simply a large artificial neural network, with multiple layers between the input and output layers.

A vast majority of fingerprint algorithms based on Deep Learning algorithms are focused on optimizing and improving only part of recognition process, as the process of fingerprint recognition is divided into stages: The segmentation of the fingerprint image, the estimation of the useful information on the image, the minutiae extraction, and the description and comparing of the extracted minutiae.

J. Ezeobiejesi et Al proposed in [16], a conventional paradigm for extracting a variable length feature representation comprised of the minutiae and their associated descriptors in a given fingerprint image [16].

Bhavesh Pandya et al in [8] used a convolutional neural network for fingerprint classification. In this method, fingerprint images are preprocessed using histogram equalization, Gabor filtering and image thinning and fed as input to their convolutional neural network for classification. This method achieves 98.21% accuracy when tested on smaller fingerprint database [21]

I.6. Limits and Vulnerabilities of Fingerprint Recognition Systems

Extensively used in securing mobile phones, doors and computers, fingerprint based authentication systems needs to be done rapidly. Also, these systems are subject to attacks, as hackers will try to exploit any vulnerability of the authentication system. Therefore, Researchers and Engineers needs to be aware in any development in the security of the systems they build. Under this section, we present an overview of vulnerabilities of fingerprint based recognition systems, and eventual solutions to these vulnerabilities and threats.

As time went on, this field of Fingerprint Recognition using Machine Learning Techniques knew great improvement in terms of efficiency, response time and security. But, as technology and knowledge keeps evolving, the proposed solutions quickly presented their limits and drawbacks. The First we are going to talk about is that presented by Philip Bontrager et Al in [38].
In this work, they propose an algorithm capable of generating what is known as "Masterprint", which can be seen as a universal fingerprint pattern, used in dictionary attacks of fingerprint recognition systems. Masterprints are real or synthetic fingerprints that can fortuitously match with a large number of fingerprints, thereby undermining the security afforded by fingerprint systems [38]. Philip Bontrager et Al used the advantage of multiple partial enrollment of fingerprint, when fingerprint sensor is small, as that of mobile phones. They went ahead and presented a series of scenarios of possible attacks on the biometric system, from the point of attack (Access point), among which the repudiation or the refusal of one party to have accessed the system using his credentials

These shortcomings can be prevented by storing and transmitting only encrypted biometric information, using cryptographic algorithms such as the Vernam's Algorithm, and Asymetric Cryptosystems. In paper [4] Mahesh Joshi et Al devised 16 vulnerability points from which attacks can be conducted, among which biometric sensor (AP2), fingerprint image interception (AP3)..., Two vulnerabilities are studied at the sensor level, where Kanich et Al used the idea that treated skin disease on fingertips modifies the structure of the dermis, eventually modifying the fingerprint of an individual, contradicting the fact that each individual has a unique fingerprint. They further presented how fingerprint spoof (Fake) could be obtained from users' fingerprint left and latter collected somewhere, and used to attack fingerprint recognition systems
Also, as mentioned in the review section of this chapter, most fin-

gerprint recognition system suffers from extra computational time, as most of them have to resize the input image to a particular size during preprocessing stage. We must also mention that being based on Machine learning techniques, the sensitiveness of these training algorithms on the training data and parameters is a limitation, since any biasedness in the training data will lead to false results, and any change in training data will lead to change in the obtained results

Of all the machine learning techniques used in fingerprint authentication, and taking into consideration the success rate at which these algorithms are being implemented, we use a deep convolutional neural network algorithm in the biometric recognition system proposed in this dissertation, with the main objective being to contribute in accelerating the overall authentication time.

II

Chapter

Implementation of a feed-forward convolutional neural network: ResNet

Contents

II.1. Introduction

A security system can be divided into two stages: Identification and authentication. The subtle difference between the two terms is that identification is the process of going through a database and finding the existence of some predefined user, while authentication is the process of demonstrating the authenticity of the user, that he is really the person he claims to be. This two-phase process is generally used to combat identity theft attack.

The system we propose combines the two identification steps in one.

To ease the identification step of the system, we use level-1 features to characterize and classify all fingerprints stored in our database. We extract level-2 features to check the authenticity of the user's print.

II.2. Tools and Programming Languages

II.2.1. Python

Python is a popular and widely supported general purpose and high level programming language. Because of this, many similar project have been built using this programming language, and it is easy for us to lay hands on any useful documentation.

Python code takes less time to write and has a larger selection of pre-built libraries and packages for scientific computing such as machine learning and image processing, which are relevant to our work in this dissertation. Some of these libraries include mexnet, OpenCV and NumPy, which are extensively used in the implementation of our proposed system

II.2.2. Anaconda

Anaconda is a distribution of the python programming language for scientific computing such as data science and machine learning, that aims to simplify package management and deployment.

We prefer the Anaconda distribution because it is somehow complete and suited for beginners in data science and machine learning as it comes with about 250 packages already installed, and includes a GUI Anaconda Navigator, as graphical alternative to the command line interface (CLI). More to this, the Anaconda Navigator provides access to some essential applications such as Spyder and Jupiter Lab by default.

II.2.3. Spyder & Jupiter Lab

Spyder is a free and open source scientific environment written in python and used for python programming. Advantages of this IDE include interactive execution, deep inspection and beautiful visualization capabilities of its scientific packages. More to this, Spyder features a good combination of the advanced editing, analysis, debugging and profiling functionality of a comprehensive development tool with the data exploration.

The Jupiter Lab is a web-based user interface and environment used for python development. We use it in our project for debugging purpose. It is highly used in debugging because of its ability to execute and present the result of each line of code, giving an easy possibility for tracing the flow of a program and correcting errors.

II.2.4. MySQL Database

A database is needed in this project to store the extracted templates of the fingerprints of users, so as to later on proceed with authentication. For this, we use the MySQL database, which is an open source relational database management system. A relational database is used because it organizes data into related tables, which helps in structuring the data.

II.3. Types and Characteristics of Fingerprints

As mentioned earlier, fingerprints are unique patterns made by friction ridges, which appear on the pads of the fingers and thumbs. This is an example of the print left when an inked finger is pressed onto paper. These friction ridge patterns that makes up the fingerprint are grouped into three distinct types: **Loop, Whorl and Arch**, which are known as level-1 feature characteristics

II.3.1. Level-1 feature characteristics of fingerprints

Loops are prints that recurve back on themselves to form a loop shape. Divided into radial loops and ulnar loops, account for approximately 60 percent of pattern types [20]
Whorls form circular or spiral patterns, like tiny whirlpools. There are four groups of whorls: plain (concentric circles), central pocket loop (a loop with a whorl at the end), double loop (two loops that create an S-like pattern) and accidental loop (irregular shaped). Whorls make up about 35 percent of pattern types

Arch create a wave-like pattern and include plain arch and tented arch. Tented arch rise to a sharper point than plain arch. Arch make up about five percent of all pattern types. These types are illustrated in 2 below

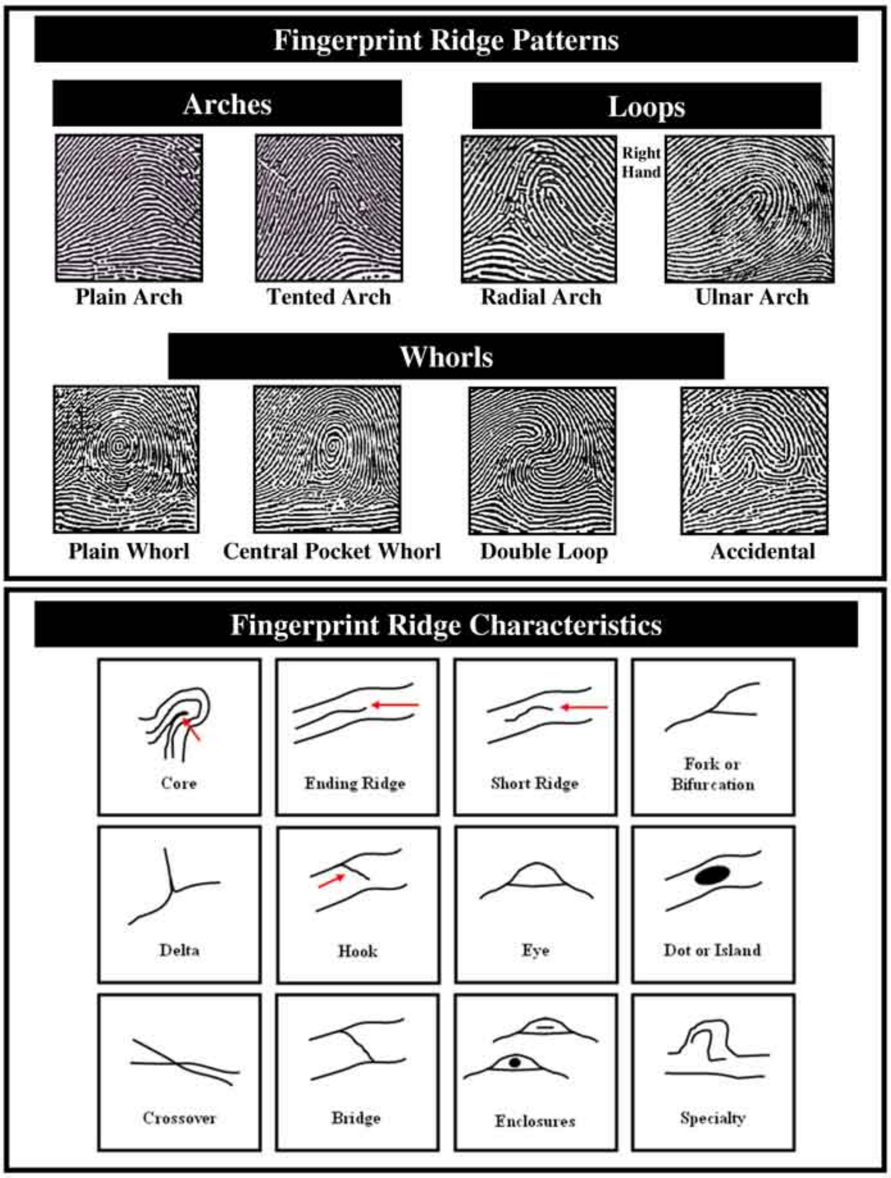


Figure 2 – *types and characteristics of fingerprints*

Every fingerprint is unique and can be classified as either loop, whorl or arch. What makes the uniqueness of each fingerprint are the spatial characteristics of the ridge pattern as indicated on figure 2 and the way the loops, whorls and archs align on each finger.

Fingerprint analysis looks at unique differences like the distance between different ridges, the heights of archs, the shape of whorls and the length of loops: These are the level-2 feature characteristics of fingerprints

II.3.2. Level-2 feature characteristics of fingerprints

Fingerprint Core is defined as the innermost turning point where the ridge forms a loop
Fingerprint delta is defined as the point where these ridges form a triangular shape (Leonard, 1988).
A bifurcation is a point in the fingerprint pattern at which two ridges meet. Bifurcations have the appearance of branch points between curved lines.
A ridge ending as indicated on the image, is the point at which the ridge ends, and becomes a valley
A short ridge is a ridge that is relatively very short, as compared to all other ridges of the fingerprint
A bridge is a short ridge, that is used as interconnection between two long ridges. At each of the endings of such ridge, is a longer ridge.
The Crossover is a point of intersection between two long ridges.
Other level 2 characteristics of the fingerprint are The **Hook**, the **Eye**, the **Speciality**, the **dot or island** and the **Enclosures.**
The ridge characteristics that we use in this text to uniquely identify fingerprints are the **bifurcation and the Ridge Ending**. This because these features can be easily spotted out on fingerprint images. These points are illustrated in figure 2

II.3.3. Level-3 feature characteristics of fingerprints

Level 3 fingerprint details are more precise details, used by print examiners when they do not want to rely only on a certain number of points on a fingerprint image, and their positions, but want to go further in exploring particular pore information details. It is noted that not all sensors are used to capture level-3 feature characteristics, as they are highly detailed. Only high resolutions sensors, usually as from 1000dpi are used to reliably extract level-3 features, and are generally not perceptible at naked eyes. The level-3 features are the pores, the edge contours, the incipient ridges, the fingerprint scars, the ridge path deviation, the width and the shape.

II.4. Architecture of a biometric system

A biometric authentication system is mainly made of two modules, the enrollment and the matching module, as shown in figure 3

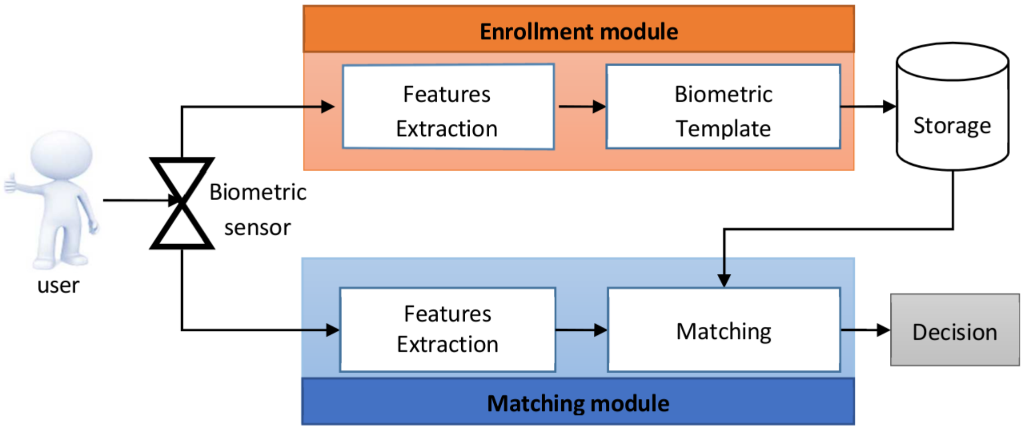


Figure 3 – *Architecture of a Biometric system*

- **Enrollment Process:** This is where the biometric information is collected from the users for the first time. This data is processed for quality enhancement, after which some relevant features are extracted, to generate a compact representation called template, that efficiently resumes the biometric characteristics. This generated template is then sent to the storage system, which can be a database containing some extra biographic information such as the name or gender.

- **Matching Process:** When the system is requested to authenticate a person, it extract the relevant features and generates a template, as during enrollment. After this, the initially stored template is revoked to be compared with the newly obtained template. This comparison aims to confirm or not that the two templates originate from the same person

The flowchart diagram of the proposed fingerprint system is given in figure 4, and this chapter is devoted to the explanation and implementation of the various components or parts of this system, starting with the presentation of tools and programming languages used.

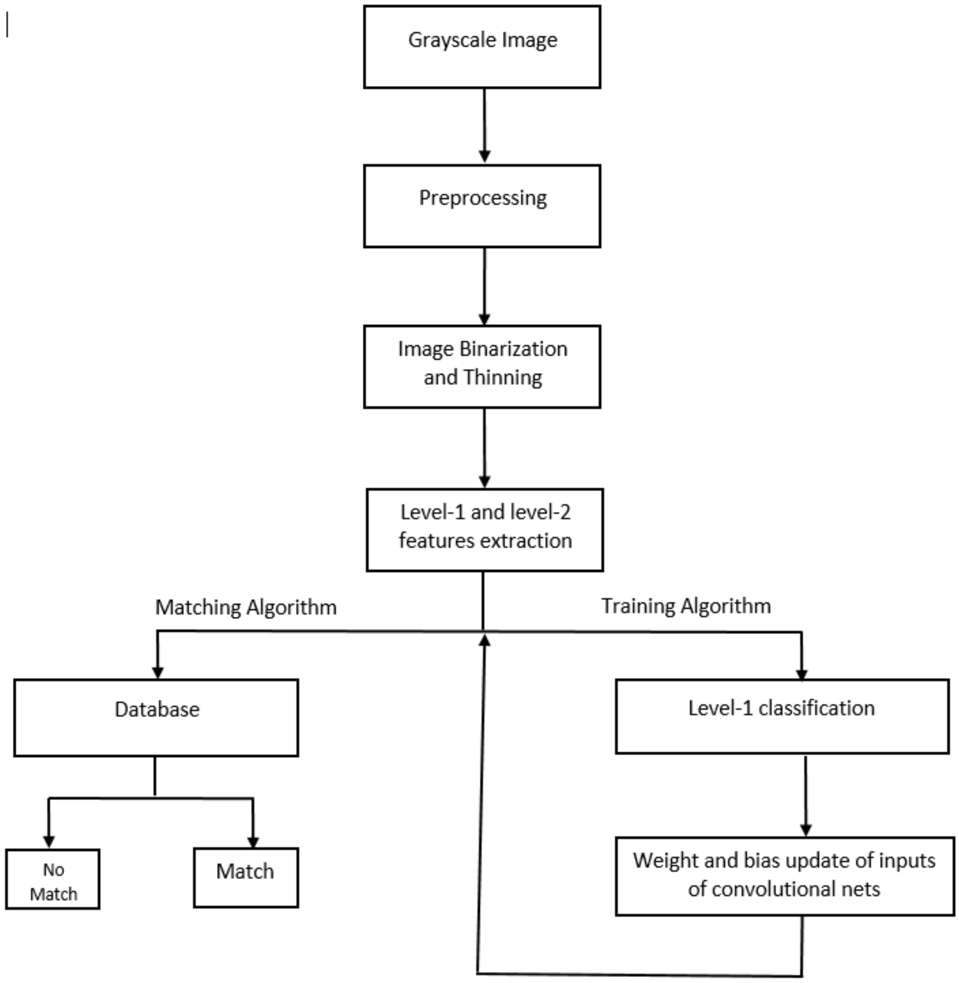

Grayscale Image
Preprocessing
Image Binarization and Thinning
Level-1 and level-2 features extraction
Matching Algorithm
Training Algorithm
Database
Level-1 classification
No Match
Match
Weight and bias update of inputs of convolutional nets

Figure 4 – *Flowchart of proposed fingerprint system*

II.5. Image Pre-processing and Binarization

II.5.1. Image Preprocessing

Image pre-processing is the use of computer algorithms to better the quality of an image. The objective of this stage is the removal of undesired distortions and noise eventually present on the input image, and the enhancement of some important features present on the image, so that the Learning algorithm can easily depict the presence of these features, which in our case are minutiae points. This image pre-processing technique is crucial for any algorithm that learns from image feature, as we don't want to bear the risk of teaching the wrong information to our algorithm because of noisy images.

We use the image filtering because with this preprocessing technique, the important features which are the positions of the minutiae points on the fingerprint images are highlited. During authentication, the input images fed into our algorithm can be of different sizes, depending on the scanner used to acquire the fingerprint image. We therefore establish a base size for all images used as input to our system.

The two-step algorithm of image filtering are:

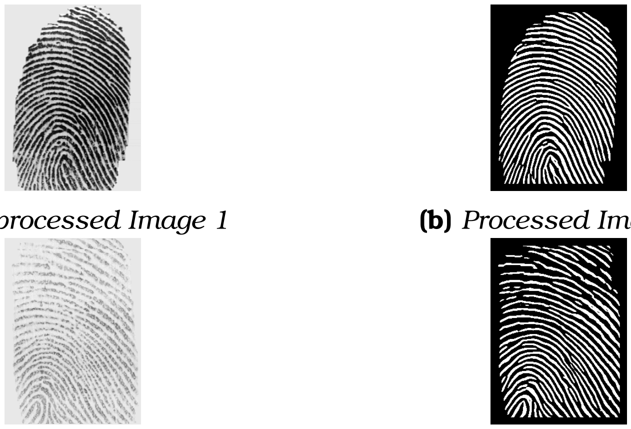
- Transform the pixel value of a given location (x,y) by using this value, and the values in some neighborhood pixels.
- Repeat this same transformation for every other pixel in the image from right to left, and top to bottom.

The Median Filtering, an effective nonlinear filtering is used in our system because of its two main advantages [39]:

- The median is a more robust average than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly.
- Since the median value must actually be the value of one of the pixels in the neighborhood, the median filter does not create new unrealistic pixel values when the filter straddles an edge. For this reason, the median filter is much better at preserving sharp edges than the mean filter.

The median filter is a non-linear digital filtering technique, often used to remove noise from an image or signal [50].

Implemented in python IDE spyder, we have as output of the preprocessing stage the images indicated in figure 5. The first input image is of good quality, and the second input image of bad quality



(a) *Unprocessed Image 1* **(b)** *Processed Image 1*

(c) *Unprocessed Image 2* **(d)** *Processed Image 2*

Figure 5 – *Output of Preprocessing Stage*

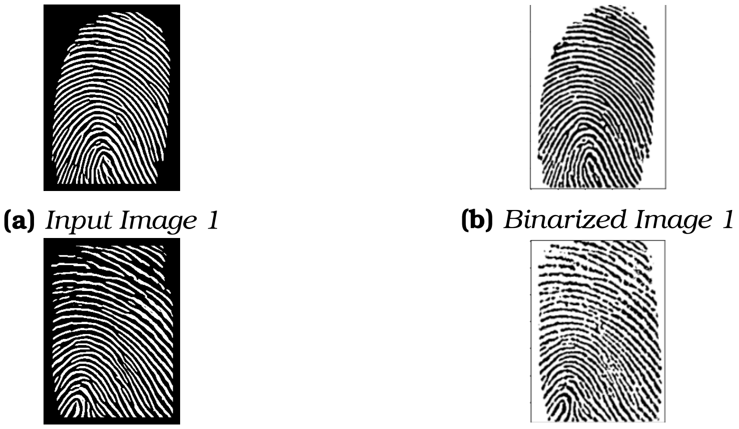
II.5.2. Image Binarization

Image Binarization is the process of taking a numeric grayscale image and converting it to black-and-white binary image [12]
The basic principle of binarization is to compare the pixel intensity which ranges from 0 to 255 with a defined threshold, and setting the pixel whose intensity is less than threshold to 0 or to 1 otherwise. Therefore, the threshold is particularly important for binarization. There are two variants of thresholds, which are the global thresholds and local thresholds.
Global threshold is one in which one single threshold value is used for the whole image and **local threshold** is one in which the threshold is calculated locally by adapting the average or mean intensity of the sweeping window.
This is resumed by the formula

$$B(x,y) = \begin{cases} 1, & \text{if } E(x,y) \geq th \\ 0, & \text{otherwise} \end{cases}$$

where *th* is the local threshold, $E(x,y)$ the input enhanced input image, and $B(x,y)$ the binarized output image.

Implemented in python IDE spyder, we have as output of the preprocessing and binarization stage the images indicated in figure 6



(a) *Input Image 1* **(b)** *Binarized Image 1*

(c) *Input Image 2* **(d)** *Binarized Image 2*

Figure 6 – *Output of Binarization stage*

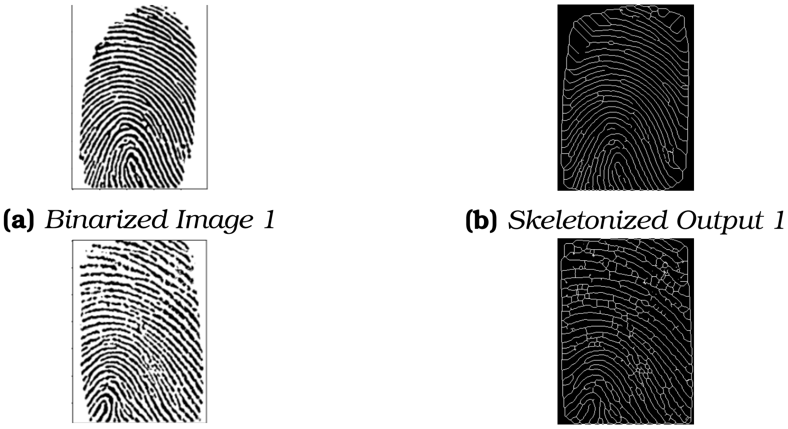
II.6. Image Skeletonization and Thinning

Thinning is the deletion of outline pixels of a connected component while maintaining their basic structure and connectivity. In this stage of thinning, we will reduce the thickness of the ridge line in the binary image to one pixel, by deleting pixels at the edge and contours of ridge lines.

The quality of the binary image has a great influence on the thinned image. If there's any one pixel disconnection in the binary image, the thinning process may end up leaving small breaks, bridges between ridges of the fingerprint image, which in turn may lead to the extraction of false level-2 minutiae of ridge ending. In this work, we use a model algorithm, inspired from that proposed by Zhang et Al in [56], which satisfies the following four conditions of a good thinning algorithm[18]

- The resultant skeleton should be 1 pixel-wide, with no redundant pixels
- Connectivity of components must be preserved
- Excessive erosion/deletion of points must be avoided
- The skeleton should be centered inside the component

Implemented in python IDE spyder, we have as output of the skeletonization and thinning stage the images indicated in figure 7



(a) *Binarized Image 1* **(b)** *Skeletonized Output 1*

(c) *Binarized Image 2* **(d)** *Skeletonized output 2*

Figure 7 – *Output of Image Skeletonization and Thinning*

II.7. Deep Convolutional Network for fingerprint classification

As indicated in section 4, fingerprint information that can be used to identify an individual can be collected or extracted at three levels: **Level-1, level-2 and level-3 fingerprint features**.
Level-1 information extracted from fingerprints can't really help in uniquely identifying an individual, but can help a long way, in reducing the number of fingerprints that must be checked, to say if or not, user's fingerprint is in database [24]. Level-1 feature permits us to classify and label fingerprints into three classes, shown in figure 8.

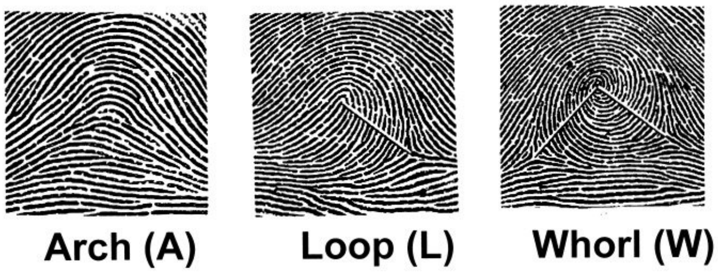


Figure 8 – *Level-1 Feature information of fingerprint*

All fingerprints in our database have been classified by our learning algorithm into one of these three classes. When we propose to our system to authenticate a user, it extracts the level-1 features of his fingerprint and predicts its class. Say for example, it appears that this user's fingerprint is WHORL. Then, we need not verify his bio data against all fingerprint of our database, but only against fingerprints initially classified as WHORL prints in our database. This helps us gain time and efficiency in our recognition system.
Level-2 features of fingerprint are more precise than the level-1 feature, and permits to uniquely associate one fingerprint to one individual. Also known as minutiae, level-2 feature appears to be the most used characteristics for automated fingerprint verification, and the two most widely used minutiae features which we use in this project are the ridge bifurcation (Image a) and ridge ending (Image b) as shown in figure 9

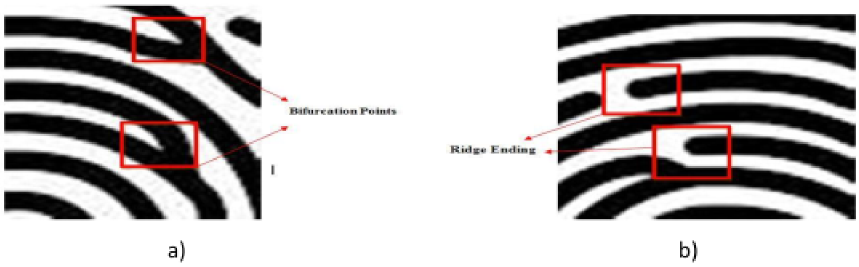


Figure 9 – *Level-2 Feature information of fingerprint*

II.7.1. Convolutional Layers

Convolutional layers are the layers where filters composed of kernels are applied to the image, to extract features; They are characterized by the number of filters they use and the sizes of these filters. The convolution operation in figure 10 below takes a 2-dimensional matrix of height and width 3 as input, and a 2×2 filter, and outputs a matrix reduced in size: a 2×2 matrix. The reduction in size of the output matrix of a convolutional layer is generalized by: If the input shape is $n_h \times n_w$ and the convolution kernel shape is $k_h \times k_w$, then the output shape will be $(n_h - k_h + 1) \times (n_w - k_w + 1)$[5]

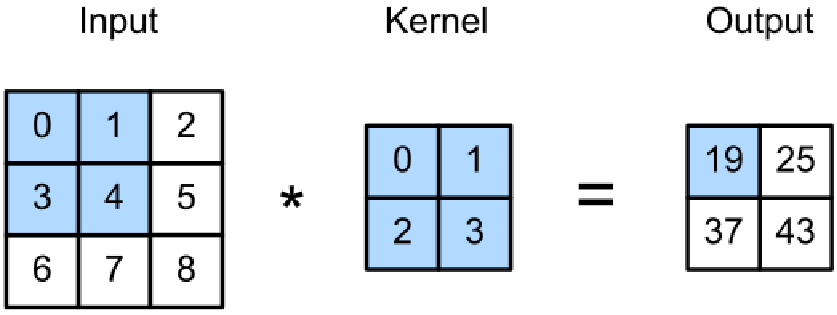


Figure 10 – *Two-dimensional convolutional operation [5]*

In the two-dimensional convolutional operation, the convolution begins with the window positioned at the top-left corner of the input tensor and slides across the input matrix, from left to right and from top to bottom. When the convolution window slides to a certain position, the input sub-matrix contained in that window and the kernel matrix are multiplied element-wise and the resulting values are summed up to give a single scalar value. This result gives the value of the output matrix at the corresponding location. The element-wise multiplication is evaluated as follows:

$$0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3 = 19$$

$$1 \times 0 + 2 \times 1 + 4 \times 2 + 5 \times 3 = 25$$

$$3 \times 0 + 4 \times 1 + 6 \times 2 + 7 \times 3 = 37$$

$$4 \times 0 + 5 \times 1 + 7 \times 2 + 8 \times 3 = 43$$

The first Convolutional layer of our algorithm receives input from the input layer. In every convolutional operation, we are given a set of inputs and we calculate the value of the current input based on all its previous inputs and their weights

The convolution layer ensures the spatial relationship between pixels by learning image geatures using small squares of input data. The objective of this layer is to extract high level features such as edges from the input image. Applying a convolution to an image is synonym to decreasing the image size, bringing all the information in the receptive field of the kernel together inot a single pixel

II.7.2. Pooling Layers

Pooling or Downsampling is simply an operation in which the size of the representation produced by the convolutional layer is reduced. The aim of this process is to speed up training process and reduce the amount of memory consumed by the network, by reducing the redundancy present in the input feature. In our algorithm, the max pooling is used, which is one in which the maximum value within a prescribed window is pooled or used in the output matrix.

A benefit of max pooling is that it forces the network to focus on a few neurons, instead of all of them, which has a regularizing effect on the network, making it less likely to overfit the training data and hopefully generalize well [15].

Unlike the convolutional operation which evaluates the output from the previous layer and a given kernel, the pooling layer operations does not have any parameter, and hence no kernel. The pooling operation in figure 11 below shows the output of max pooling, with a pooling window shape of 2×2

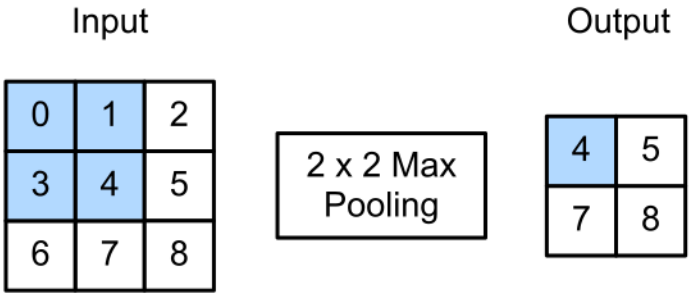


Figure 11 – *Example of Maximum Pooling operation [5]*

Like the convolution operation which begins with the window positioned at the top left of the input matrix and slides left to right and top to bottom, the pooling operation does same. The output is obtained from the maximum value of each window, when it is slid.

$$max(0,1,3,4) = 4$$
$$max(1,2,4,5) = 5$$
$$max(3,4,6,7) = 7$$
$$max(4,5,7,8) = 8$$

II.7.3. Rectified Linear Unit

At the end of the convolution operation, the output of the filter and the input image is summed with a biased term, and passed through a non-linear activation function, which introduces non-linearity into our network. This is because out input data, which is a fingerprint image is non-linear; Pixels of fingerprint image can't really be modeled linearly with ease. To account for this non-linearity, we use the Rectified Linear Unit (ReLU) function. The ReLU function is a quite simple function, that assigns 0 to each negative value, and that is invariant for positive values.

$$f(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ x, & \text{otherwise} \end{cases}$$

This function has the advantage and is preferred over other activation functions such as the sigmoid or hyperbolic tangent functions, because it overcomes the vanishing gradient problem, allowing models to learn faster, and perform better [9]

II.7.4. Fully Connected Layer

The fully connected layer in CNNs is usually placed before the output layer of the classification algorithm network, to flatten results (The rows of a matrix are concatenated to form a long feature vector) before final classification of input image.

II.7.5. Softmax Activation Function

Softmax is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each

value are proportional to the relative scale of each value in the vector. This function enables components of the output to be within the interval (0, 1), and will all sum up to 1, as required by probabilistic theories.
The standard softmax function is given by

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

where $\sigma(x_j)$ is the resultant scaled value of the individual value x_j

II.8. Training Process

Training a neural network means finding the appropriate Weights of the Neural connections and the kernels, using a feedback loop called Gradient Backward propagation. Initially, these weight are initialized randomly, and this is why at the beginning of the training process, the network performs badly, and goes better and better, while the loss function decreases.

II.8.1. Batch Normalization

Batch Normalization is an effective technique that consistently accelerates the convergence of deep networks, and ensures a better stability. Batch norm improves the learning speed of neural networks and provides regularization, while avoiding overfitting [40]. For example, in classifying a fingerprint image into either a loop or an arch or a whorl, the algorithm might consider the length of the ridges forming the fingerprint, which ranges from 10 to 150, and the orientation angle of the ridges ranging from 0 to 360. These differences in ranges can be problematic, as the value with the higher range might bias our models into giving them inflated importance. To normalize our data in this project, we scale it to a range from 0 to 1, using:

$$x_{normalized} = \frac{x - m}{x_{max} - x_{min}}$$

where x is the data point or value to normalize, m the mean of the data set, x_{max} the highest value in the dataset, and x_{min} the lowest value in the dataset

Batch normalization with the phenomenon of internal covariate shift helps in adjusting the issue of vanishing gradient in that it makes the input data to fall within a specified range of unsaturation of the activation functions, especially in the case of the sigmoid and hyperbolic tangent function.

II.8.2. Loss Function and Minimization: Stochastic Gradient Descent

A loss function measures how far an estimated value is from its true value. While experimenting and trying to improve a model, it is the loss function that tells if one is getting closer to the solution or not.

In this project, we use the multi-class cross-entropy loss. **Multi-Class** because each fingerprint sample image must be classified as one of the three classes, and **cross-entropy** because it calculates a value or score that summarizes the average difference between the actual and predicted probability distributions for all classes in the problem. In order to minimize this loss function, the stochastic gradient descent is used.

Gradient descent is an iterative algorithm, that starts from a random point on a function and travels down its slope in steps until it reaches the lowest point of that function [46]. It is an optimization algorithm used to find the values of parameters or coefficients of a function f, that minimizes the cost function

II.8.3. Parameters and Hyper-Parameters: Learning rate, Batch size and epoch

Learning rate (lr) is one of the most important hyper-parameter to tune when training neural networks. It controls how much data we take in iteration before updating and minimizing the loss function. If this value is too big, one may "jump" the minimum value of the loss function, and if it's too small, it might take too much time to converge to the minimum of the loss function. For the training of the model proposed in this dissertation, the learning rate is set to

0.001
Batch size is a parameter that indicates the number of training examples utilized in one iteration. It is the number of samples processed before the model is updated. A batch size of 100 is used to train the network proposed in this project.
The number of epochs is a hyperparameter that defines the number of times that the learning algorithm will work through the entire training dataset. It is the number of complete passes that the algorithm must make through the training dataset. For the training of the proposed model, the number of epochs is set to 45

II.8.4. Backpropagation

After each iteration of the batch size, one needs to update the weights of the neurons of the network. Since this update depends on the current output at the last neurons, the weight update takes place starting from the last (output) neurons, and comes back right till the input neurons. This process is known as backpropagation, and is efficient when the network is feed-forward.

A feedforward neural network is one in which the nodes never form a cycle, and in which data moves through them in only one direction. This kind of neural network has an input layer, hidden layers and an output layer. It is the first and simplest type of artificial neural network. If the network has cycles, the transmission of the error computed at the output layer to input layer might be disolved during the backpropagation algorithm.

The back propagation algorithm in neural network computes the gradient of the loss function for a single weight by the chain rule. Consider the back propagation neural network example in figure 12

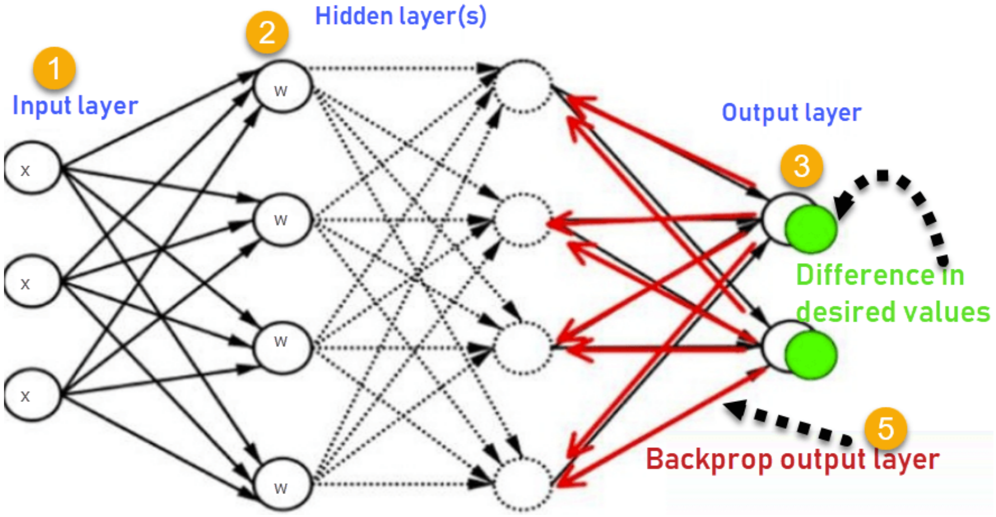


Figure 12 – *Illustration Backpropagation Algorithm*

The training process of a neural network is generalized by the following the algorithm:

- **Step 1:** Inputs X arrive through preconnected path
- **Step 2:** Calculate the output for every neuron from the input layer, to the hidden layers to the output layer
- **Step 3:** Calculate the error in the ouput. This is generally done by:

  $$Error = ActualOutput - DesiredOutput$$

- **Step 4:** Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased: backpropagation algorithm
- **Step 5:** Keep repeating the process until the desired output is achieved

II.9. Level-2 feature extraction: Terminations and Bifurcations

This process of level-2 feature extraction is carried on an image that has been binarized and squeletonized. A 3×3 window is used to swipe the squeletonized image, looking for all possible configuration of bifurcations and Terminations. These possible configurations are show in figure 13 for the terminations, and in figure 14 for the bifurcations

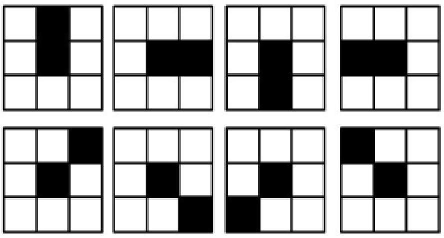


Figure 13 – *All 3 × 3 window possibilities for a Ridge Termination in a binarized and skeletonized fingerprint image [20]*

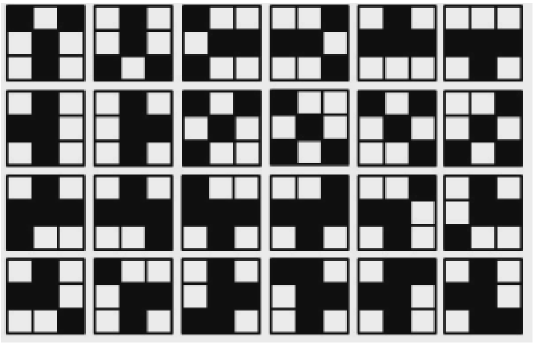


Figure 14 – *All 3 × 3 window possibilities for a Bifurcation in a binarized and skeletonized fingerprint image [20]*

Implemented in python and using some proposed libraries, we have the outputs in figure 15, where the blue circles represent the bifurcations, and the red circles represent the terminations

(a) *Input image of the level-2 extraction process*

(b) *Result of Level-2 feature extraction*

Figure 15 – *Output of level-2 feature extraction stage*

For each of the Termination points above, 3 parameters are used in obtaining the fingerprint's template that is stored into the database.

These parameters are the x-coordinate of the point on the image, the y-coordinate of the point on the image, and the direction angle of the ridge constituting the termination.
For each of the Bifurcations, 5 parameters are stored as part of the fingerprint's template: The x-coordinate and the y-coordinates of the point, and the three angles of the three ridges forming the bifurcation point.

In the above image, there are some false minutiae points that can easily lead to a false acceptance, or a false rejection. Three principles stated in [20] are used to eliminate some of these false minutiae points. These principles are:

- **Principle 1:** If the distance between a termination and a bifurcation is smaller than D_1, then these two minutiae could be false minutiae. We should remove them. Experimentally, D_1 is set to 10
- **Principle 2:** If the distance between two terminations is smaller than D_2, then these two minutiae could be false minutiae. We should remove them. Experimentally, D_2 is set to 6
- **Principle 3:** If the distance between two bifurcations is smaller than D_3, then these two minutiae could be false minutiae. We should remove them. Experimentally, D_3 is set to 6

II.10. Matching Algorithm

It is said in [22] that for a given fingerprint, 10 minutiae points are necessary to uniquely identify it and find a match in a database of fingerprints.
After the classification of images and extraction of minutiae points, the coordinates of these points are used to generate a feature vector, which is then used in a matching algorithm.The Matching problem is a quite difficult one [7], as the issue of rotation and Translation invariance must be taken into consideration, so that if the same fingerprint image with translated or rotated coordinates for the minutiae points is given to the system, it should be able to identify this

image as belonging to a recognized individual.

II.10.1. Translation invariance

For this, we implement a modified version of the Minutia-based matching principle proposed in [7], in which the relative distances between minutiae points are used. Hence, there is a match between two minutiae points if the Euclidean distance between these two points is less than or equal to a given threshold value.

The feature vector used during the matching stage of the system proposed in this project is made of 10 coordinates, which are the pair wise distances between the first 5 Termination points and the first 5 Bifurcation points, from left to right and from top to bottom. These Euclidean distances are calculated using

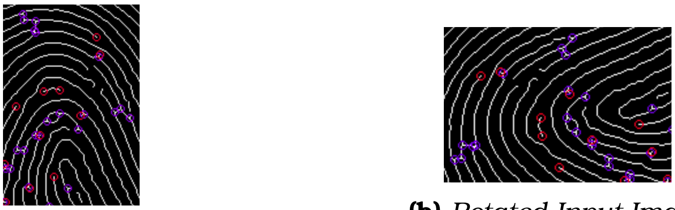
$$d(A,B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Where A and B are two minutiae points with spatial coordinates (x_1, y_1) and (x_2, y_2) respectively

II.10.2. Rotation invariance for a one to one authentication

The problem of rotation invariance is as follows: The minutiae points of the fingerprint image are extracted from top to bottom, and from left to right. If the fingerprint image is rotated, then it is possible that the order in which the minutiae points were obtained during enrollment is different from that obtain during authentication. And hence, the templates which are feature vectors might not be the same, leading to a False rejection of a user.

Consider the figure 16 which illustrates the problem of rotation invariance. The two subfigures are images of the same fingerprint, but figure b has been rotated by an angle of 90^o. Considering these two images, the two first Bifurcation points that are registered in the first image (figure a) are different from those registered when the image is rotated by 90^o (figure b). Hence, the first coordinate of feature vector in figure a which is the distance between the two first minutiae points might be different from that of figure b, leading to a false rejection.



(a) *Input Image*

(b) *Rotated Input Image*

Figure 16 – *Illustrating example of the problem of rotation invariance*

To go round this problem, we consider the delta point as rotational reference. During enrollment, the direction angle of this point is extracted from image and stored in database. During authentication, if we find out that the direction angle θ of proposed image for authentication is different from the angle α stored as template in database, then we understand that a rotation of angle $360 - |\theta - \alpha|$ needs to be done on input image.
This technique works well in the case of a one to one authentication, when we want to check if two fingerprint images have the same feature vectors or not, irrespectively of any translation. This techniques will be time consuming and cannot be implemented for a one to many identification, as each time the angles of the delta points are different, the authentication fingerprint has to be rotated, and features re-extracted.

Because of time constraint, we could not implement a solution when the fingerprint image is rotated for a one to many authentication. Hence, the results shown in the next chapter for one to many authentication are under the assumption that the fingerprint images are always oriented in the same direction (Vertically)

III

Chapter

Results and Discussions

Contents

The results obtained and presented in this chapter are under two hypothesis

Hypothesis 1: During authentication, all fingerprint images have the same direction as they had during enrollment. Rotation Invariance is not taken into consideration in this present work, and can be the object of further research.

Hypothesis 2: During authentication, a complete fingerprint image translated or not is submitted to the system. Not a partial image of fingerprint is given to the system for authentication.

III.1. Results

III.1.1. Programming Environment and Dataset

Our experimental results are carried out using two datasets, the popular FVC2002 DB4 (100 different fingers each represented by 8

fingerprint impressions) containing some challenging cases of fingerprints, and a Kaggle like datasets of 150 images, making a dataset of 950 fingerprint images.
All tests are carried out on an Intel Core i7-4910MQ CPU 2.90GHz machine, running Ubuntu 19.04
Algorithms are written in python 3.0, using the spyder IDE and jupyter notebook.

III.1.2. User Interface

A simple Graphical User Interface is used to test the system proposed as shown in figure 17. This Interface is made up of three sections, the first which shows the path from which we select the fingerprint image that is to be authenticated, the second which shows the real fingerprint image to be authenticated, and the third column shows the skeletonized image from which minutiae points have been extracted, and the results of the verification process: If a match was found in the database or not.

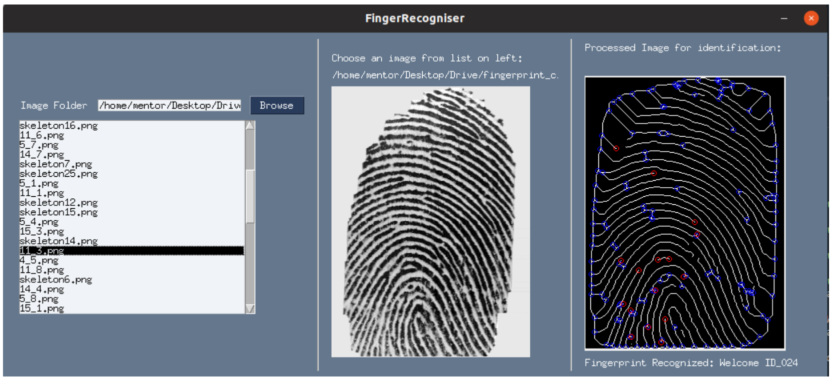


Image Folder /home/mentor/Desktop/Driv

```
skeleton16.png
11_6.png
5_7.png
14_7.png
skeleton7.png
skeleton25.png
5_1.png
11_1.png
skeleton12.png
skeleton15.png
5_4.png
15_3.png
skeleton14.png
11_3.png
4_5.png
11_8.png
skeleton6.png
14_4.png
5_8.png
15_1.png
```

Choose an image from list on left:
/home/mentor/Desktop/Drive/fingerprint_c

Processed Image for identification:

Fingerprint Recognized: Welcome ID_024

Figure 17 – *Authentication User Interface*

III.1.3. Classification Results

With the proposed network, we could obtain a classifier with an accuracy of 98% during training, and within a reasonable time frame, of about 2 hours of training as shown in figure 18

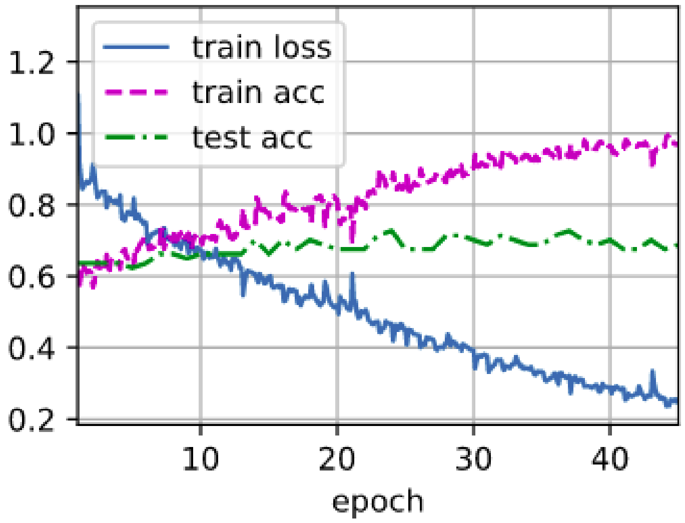


Figure 18 – *Training Results*

Making the hypothesis that the three types of fingerprint images are uniformly distributed in a database, applying the technique proposed above should theoretically reduce by a factor of 3, the time needed for a complete authentication. However, this is different from the results obtained in practice.
Table II shows the parameters used in training, with the results obtained

Metrics	**Values**
Training Images	809
Validation Images	77
Batch Size	100
Number of Epoch	45
Training Time	2 hours 15 mins
Learning Rate	0.001
Training Accuracy	98.8%
Testing Accuracy	72.1%
Loss Value	0.15

Table II – *Training Parameters and Results*

III.1.4. Matching Evaluation and Accuracy

For the accuracy and matching evaluation, we enrolled a total number of 310 fingerprint in a database, distributed into three tables, 110 Arch fingerprints for table arch, 50 Loop fingerprints for loop table and 50 Whorl fingerprints for the whorl table.

In order to evaluate the average overall authentication time, we first use 210 enrolled fingerprint images. The system loops through all 210 fingerprint images, indicating the match result of each of the fingerprints: Found or not Found. The loop ends and the results of the 210 fingerprint images are obtained after 1 minute and 27 seconds. Then, we consider 210 impostor fingerprint images not enrolled in the database, and the system loops through all 210 images, indicating the match result of each of the fingerprints. The loop ends after 50 seconds. This gives an average global time for the 210 images of:

$$T_{avg} = \frac{87+50}{2} = 69 seconds$$

This gives an average time for an authentication of:

$$T_{avg} = \frac{69}{210} = 0.328 seconds$$

In order to evaluate the False Acceptance Rate and the False Rejection Rate, we consider two scenarios. In the first scenario, we use 210 fingerprint images already enrolled. Of these 210 enrolled images, 12 fingerprint images could not find a match in the database, probably because of misclassification, or extraction of false minutiae points. This results in a False Rejection Rate of 5.7%, as:

$$FRR = \frac{Number\_of\_rejected\_clients}{Number\_of\_client\_transaction} = \frac{12}{210} = 0.057$$

In the second scenario, we considered 210 impostor fingerprint images, which were never enrolled in the database, and 0 match was found. This leads to a False Acceptance Rate of 0%, as

$$FAR = \frac{Number\_of\_accepted\_impostors}{Number\_of\_impostors\_transaction} = \frac{0}{210} = 0$$

And finally, the Half Total Error Rate is given by:

$$HTER = \frac{FAR+FRR}{2} = \frac{0+0.057}{2} = 0.028$$

III.2. Discussions

III.2.1. Critical Study

As mentioned in the state of art, Huong Thu et Al proposed in [32] 4 fingerprints classifiers, based on Random Forest, on Support Vector Machine, on Convolutional Neural Network, and on K-Nearest Neighbor.

A comparative study of Platforms or computers on which they implement the experimental results of their classifier and ours is given in table III

Computer Characteristics	State of Art [32]	Used in Present Work
Operating System	Windows 10	Ubuntu 19.04
C.P.U.	Corei3	Corei7
CPU frequency	3.0 GHz	2.9 GHz
RAM	2GB	8 GB

Table III – *Comparative Study of Computer Platforms*

A comparative study of classification results obtained in [32] with ours is given in table IV

Evaluation Metrics	R.F.	S.V.M.	C.N.N.	K-NN	Proposed System
Training Time	5H15	7H40	8H32	5H28	2H30
Training Accuracy (%)	96.25	96.76	95.82	91.90	98.8

Table IV – *Comparative Study of Classification Results*

A comparative study of Matching results obtained in [55] and in [5] with ours is given in table V

Evaluation Metrics	Article [55]	Article [5]	Proposed System
FAR (%)	1.22	0	0
FRR(%)	9.23	20	5.7
HTER(%)	5.225	10	2.8

Table V – *Comparative Study of Matching Results*

III.2.2. Difficulties

Some major difficulties we had in this dissertation are **Could not have access to some articles on research gate** We could not lay hands on some good articles with interesting results. mainly because they were of restricted access. Some articles on Research Gate are only accessible through your Research Institution via an institutional email account, which is only given to students in phd. This caused us some non-negligeable prejudices

Classification Dataset We could not find a dataset classified with respect to the various fingerprint classes. We had to download the public FVC database, and do a manual classification of fingerprint images as either an Arch, a Whorl or a Loop. It is possible that this manual classification was error prone, and might have contributed to the loss in classification accuracy obtained during the training process, and hence reducing the overall accuracy and efficiency of proposed system.

Could not design very complex and efficient model, due to the lack of training hardware: GPU It is only in the past decades that AI and Deep Learning have become a captivating research domain, with promising results in real world problems such as image classification and computer vision [5]. Much of this recent progress in deep learning has been triggered by an abundance of data arising from cheap sensors and internet-scale applications, and by significant progress in computation, mostly through GPUs [5]. Unfortunately, we did not have any GPU on which we could train and test any complex, but accurate model. This lead us only to explore solutions that can be trained and deployed easily.

General Conclusion

Contents

Summary

We proposed ourselves in this present dissertation to build a complete fingerprint authentication system, using machine learning techniques to accelerate the process and augment precision and accuracy.

To achieve this, we started with some preprocessing on fingerprint image, where we resize and better the quality of input image, for good accuracy in classification. This image is then binarized and skeletonized, for minutiae feature extraction. Skeletonized image is then fed into a CNN, which decides in which set of fingerprint images should we look for a possible match. After the classification stage, we now extract level-2 minutiae points which are in turn used as parameters of feature vector used as fingerprint template.

To accelerate the process, we made use of some open and public libraries such as opencv, skimage, numpy and FingerprintFeature-Extraction.

Perspectives

After the studies presented in this project, we think that if further research should be conducted under this same topic, attention should be given to the following:

Tacke down the problems of rotation invariance and partialness of fingerprint images during one to many authentication: To complete the proposed system and make it usable, one must tackle down the problems of rotation invariance, for a same fingerprint to be recognized by the system if it is rotated, or not in the same direction as it was during enrollment. Also, System should be able to recognize partial fingerprint image of an already enrolled image

Use Transfer Learning techniques on standard deep neural nets designed for other purposes, to do same thing as in this work: Transfer Learning is a technique in machine learning that consists of storing and using knowledge gained while solving one problem in some other related problem. One could use a modern network already trained such as DenseNet, GoogLeNet or Network in Network (NiN) to classify fingerprint images before the matching stage. This can improve on the precision and accuracy of the system, as these networks are trained on larger datasets, with good and sophisticated materials and resources such as scalable GPUs.

Explore the line of unsupervised training: Engelsma et Al. in [23], used unsupervised learning in the process of fingerprint authentication, and had better results than the available SDK Innovatrics and VeriFinger. To the best of our knowledge, this field of unsupervised learning applied to biometric recognition is not very much explored by researchers. We think that, using unsupervised learning in biometric authentication can help in learning parameters like the pressure that the user exerts on the scanner, the way a particular user slides his finger on the scanner. These parameters can help a long way in fighting against attacks such as spoofs and machine generated fingerprints.

The image preprocessing and Feature extraction: If there's any one pixel disconnection in the binary image, the thinning process may end up leaving small breaks, bridges between ridges of the fingerprint image, which in turn may lead to the extraction of false level-2 minutiae. The minutiae extractor algorithm needs to improved, so as to reduce the number of false minutiae point extracted

Most of the fingerprint images are the loops. Is it not preferable to just identify level 2 features without classifying level

1 features: Some research in article [20] shows that loops are the most common type of fingerprint, and that on average, 65% of all fingerprints are loops. This information can surely be used to improve on the time needed to identify a fingerprint as either a loop, a whorl, or an arch. One could try to answer the question: To what extend or how large should a dataset be for level-1 classification not to be necessary.

Bibliography

[1] S. Pankanti A. K. Jain, Lin Hong and R. Bolle. An identity-authentication system using fingerprints. *In proceedings of the IEEE*, 85(9), 1997.

[2] S. Prabhakar A. K. Jain and S. Parkanti. Fingercode: A filterbank for fingerprint representation and matching. *Computer Vision and Pattern Recognition and Computer Society Conference*, 1(1), 1999.

[3] Biometrics Evolution Amanda Douglas. A brief history of biometrics. https://bioconnect.com/a-brief-history-of-biometrics/, Feb 14, 2020. Visited on December 17 2020 at 08:05.

[4] Sudipta Banerjee Arun Ross and Cunjian Chen. Some research problems in biometrics: The future beckons. *International conference on Biometrics*, 1(1), June 2019.

[5] Mu Li Aston Zhang, Zachary C. Lipton and Alexander J. Smola. Dive into deep learning. *DEEP LEARNING*, 1(1), 2021.

[6] Ali Ismail Awad. Machine learning techniques for fingerprint identification: A short review. *Biometrics Conference Paper*, 1(1), 2012.

[7] Foudil Belhadj. Biometric system for identification and authentication. *Computer Vision and Pattern Recognition, Ecole Nationale Superieure en Informatique Alger*, 2017.

[8] cosma G. Bhavesh Pandya. Fingerprint classification using a deep convolutional neural network. *4th IEEE International conference on Information Management Oxford, United Kingdom*, 1(1), 2018.

[9] Jason Brownlee. A gentle introduction to the rectified linear unit. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/, 2019. Visited on April 14 2021 at 16:57.

[10] K. Cao and A. K. Jain. Fingerprints indexing and matching: An integrated approach. *Biometrics International Joint Conference*, 1(1), 2017.

[11] Mandal S. Chartterjee A. Fingerprint identification and verification system by minutiae extraction using artificial neural network. 1(1), 2010.

[12] THE CRAFT OF CODING. Image binarization (1) : Introduction. https://craftofcoding.wordpress.com/2017/02/13/image-binarization-1-introduction/, 2017. Visited on March 25 2021 at 10:55.

[13] Awad Egawa S. and Baba K. A short review on machine learning techniques used for fingerprint recognition. *Springer, Heidelberg*, 294(1), 2012.

[14] Emerj. Overview of ai in biometric authentication. https://emerj.com/ai-sector-overviews/ai-in-biometrics-current-business-applications. Visited on December 04 2020 at 05:58.

[15] Alejandro Escontela. Convolutional neural networks from the ground up. https://towardsdatascience.com/convolutional-neural-networks-from-the-ground-up-c67bb41454e1, 2018. Visited on April 14 2021 at 17:57.

[16] J. Ezeobiejesi and B. Bhann. Latent fingerprint image segmentation using deep neural network. *Deep Learning For Biometrics*, 1(1), 2017.

[17] THALES headquarters. Biometrics: Definition, trends, use cases, laws and latest news. https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/inspired/biometrics, December 4, 2020. Visited on December 17 2020 at 09:01.

[18] Archana Praveen Kumar Himanshu Jain. A sequential thinning algorithm for multi-dimensional binary patterns. *IEEE Transactions on Image Processing*, 2017.

[19] H. Sadeghi HM Jirandeh and MSJ Rad. High-resolution automated fingerprint recognition system based on gabor wavelet and svm. *International Journal of Scientific and Engineering Research*, 5, 2014.

[20] Laura Hospital. Which fingerprint is most common? https://www.perkinselearning.org/accessible-science/activities/which-fingerprint-most-common, 2015. Visited on July 24.

[21] Laxman Singh Jay Kant Pratap Singh Yadav and Zainual Abdin Jaffery. Evaluation of acceleration algorithm for biometric identification. *Journal of Critical Reviews, Department of Electrical Engineering, Jamia Millia Islamia, Central University, New Delhi*, 1(1), 2020.

[22] Dae-Seong Jeoune, Chan-Myeong Han, and Yun-Kyoo Ryoo et Al. Minutiae-based matching scheme for fingerprint identification using the grassfire spot matching. *International Journal of Applied Engineering Research*, 12(9), 2017.

[23] Anil Jain Joshua Engelsma, Dai-Cai. Fingerprints: Fixed length representation via deep networks and domain knowledge. *Michigan State University, USA*, 1(1), 2017.

[24] Stan Z. Li. Encyclopedia of biometrics: I - z. *Springer Science & Business Media*, 2(7), 2009.

[25] Sami Luukkonen. Are passwords becoming obsolete? https://www.forbes.com/sites/valleyvoices/2015/10/12/are-passwords-becoming-obsolete/?sh=4c1e2c3cb5e8, 2015. Visited on March 1, 2021 at 07:20.

[26] Sozan Abdulla Mahmood. Fingerprint recognition system using support vector machine and neural network. *International Journal of Computer Science Engineering and Information Technology Research*, 4, 2014.

[27] Pavol Marak and Alexander Hambalik. Fingerprint recognition system using artificial neural network as feature extractor: Design and performance evaluation. *TaTra Mountains Mathematical Publications*, 2016.

[28] Stephen Mayhew. Explaining biometrics: History of biometrics. https://www.biometricupdate.com/201802/history-of-biometrics-2, 2019. Visited on December 17 2020 at 08:16.

[29] Stephen Mayhew. What is fingerprint identification? https://www.biometricupdate.com/201205/what-is-fingerprint-identification,

2021. Visited on May 30, 2021.

[30] Shervin Minae and Elhan Azimi. Fingernet: Pushing the limits of fingerprint recognition using convolutional neural network. *New York University, University of Californial, Riverside*, 1(1).

[31] Tom M. Mitchell. The discipline of machine learning. *Machine Learning Department School of Computer Science, Carnegie Mellon University Pittsburgh*, 1(1), 2006.

[32] Huong Thu Nguyen and Hong The Nguyen. Fingerprints classification through image analysis and machine learning methods. *Artificial Intelligence Laboratory, Institute of Information Technology, and Data Science Russia*, 2019.

[33] Huong Thu Nguyen and Long The Nguyen. Fingerprints classification through image analysis and machine learning method. *Algorithm, Artificial Intelligence Laboratory, Institute of INformation Technology and Data Science*, 2019.

[34] Chris Nicholson. A beginner's guide to neural networks and deep learning. https://wiki.pathmind.com/neural-network#ai, 2020. Visited on March 04 2021 at 09:08.

[35] M. Asante O. Appiah and H. Acquah. Improved approximated median filter algorithm for real-time computer vision. *King Saud University-Computer and Information Sciences*, 2020.

[36] Ozcinar C. Anbarjafari G Pavlo Tertychnyi. Low quality fingerprint classification using deep convolutional neural network. *IET biometrics*, 1(1), 2018.

[37] D et Al. Peralta. A survey on fingerprint minutiae based local matching for verification and identification: Taxonomy and experimental evaluations. *Information Sciences 315*, 2015.

[38] Nasir Menon Philip Bontrager Aditi Roy Julius Torgelius and Arun Ross. Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution. *New York University, Michigan State University*, 1(1), 2020.

[39] S. Perkins R. Fisher. Median filter. https://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm, 2020. Visited on March 28 2021 at 22:24.

[40] Martin Riva. Batch normalization in convolutional neural networks. https://www.baeldung.com/cs/batch-normalization-cnn, 2019. Visited on June 24 2021.

[41] Qi J. Rui Xie. Continuous fingerprint image quality estimation based on neural network. *The International Symposium on Intelligen Signal Processing and Communication systems*, 1(1), 2010.

[42] Tola Sokunbi S. Adebayo Daramola and A. U. Adoge. Fingerprint verification system using support vector machine. *International Journal on Computer Science and Enginnering*, 5(7), 2013.

[43] Amirali Abdolrahidi Shervin Minae, Elhan Azimi. Fingernet: Pushing the limits of fingerprint recognition using convolutional neural network. *New York University, University of California, Riverside*, 1 (1), 2019.

[44] A. Abdolrashidi Shervin Minaee and Y. Wang. Iris recognition using scattering transform and textural features. *Signal Processing and Signal Processing Education Workshop, IEEE*, 2015.

[45] D. Song and J. Feng. Fingerprints indexing based on pyramid deep convolutional feature. *Biometrics International Joint Conference*, 1 (1), 2017.

[46] Aishwarya V. Srinivasan. Stochastic gradient descent-clearly explained!! https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31, 2019. Visited on June 24 2021.

[47] Analytics Step. 6 major branches of ai? https://www.analyticssteps.com/blogs/6-major-branches-artificial-intelligence-ai. Visited on November 28 2020 at 06:50.

[48] Tangha C. Stephane Kouamo. Fingerprint recognition with artificial neural network. *Application to E-learning Journal of Intelligent Learning Systems and Applications*, 1(1), 2016.

[49] Wong W.J. Su H., Chen K. and S. Lai. A deep learning approach towards pore extraction for high-resolution fingerprint recognition. *IEEE International conference on Acoustics, Speech and Signal Processing*, 1(1), 2017.

[50] Wikipedia Team. Median filter. https://en.wikipedia.org/wiki/Median_filter, 2020. Visited on March 28 2021 at 22:07.

[51] Techopedia. What does artificial intelligence mean? https://www.techopedia.com/definition/190/artificial-intelligence-ai, 2020. Visited on November 28 2020, at 06:07.

[52] ThinkAutomation. What is image classification in deep learning? https://www.thinkautomation.com/eli5/eli5-what-is-image-classification-in-deep-learning/, 2020. Visited on March 04 2021 at 08:14.

[53] M. Arif Wani and Adif Iqbal Khan. Supervised deep learning for fingerprint recognition. *Biometrics Conference Paper*, 1(1), 2020.

[54] webopedia. Modeling. https://www.webopedia.com/definitions/modeling/, 2020. Visited on December 18 2020 at 07:19.

[55] Naser Zaeri. Minutiae-based fingerprint extraction and recognition. *Open Access Publisher*, 2011.

[56] T. Y. Suen Zhang and Ching Y. A fast parallel algorithm for thinning digital patterns. *Communication of the ACM*, 27(3), 1984.