## DEEP PALMPRINT IDENTIFICATION USING STACKED AUTO-ENCODER

# A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

# By SULAIMAN A.S. MILAD

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Electronics Engineering

NICOSIA, 2019

**SULAIMAN A.S. MILAD** DEEP PALMPRINT IDENTIFICATION USING STACKED **AUTO-ENCODER** 

NEU 2019

# DEEP PALMPRINT IDENTIFICATION USING STACKED AUTO-ENCODER

# A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By SULAIMAN A.S. MILAD

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Electronics Engineering

NICOSIA, 2019

# SULAIMAN A.S. MILAD: DEEP PALMPRINT IDENTIFICATION USING STACKED AUTO-ENCODER

Approval of Director of Graduate School of Applied Sciences

**Prof. Dr.Nadire CAVUS** 

#### We certify this thesis is satisfactory for the award of the degree of Masters of Sciences in Electrical and Electronics Engineering

**Examining Committee in Charge:** 

Assoc. Prof. Dr. Gözen Elkıran

Committee Chairman, Department of Civil Engieering, NUE

Asst. Prof. Dr. Boran Şekeroğlu

Department of Information System Engineering, NEU

Asst. Prof. Dr. Pinar Akpinar

Supervisor, Department of Civil, Engineering, NEU

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, last name: Signature: Date:

To my parents...

#### ACKNOWLEDGMENTS

All praises and thanks to my family. It is by their grace that I have been able to access this point in my life. I would like to express my sincere gratitude to my supervisor, Assist. Prof. Dr. Sertan Kaymak who has supported and directed me with her vast knowledge and also for his patience that ensured the completion of this thesis.

I dedicate my success to my parents, who always supported me in my studies.

Finally, I thank to my friends who supported me in every possible way.

#### ABSTRACT

Deep Learning has been widely used in various areas in computer vision. Those areas include image analysis, segmentation, classification, identification etc... this intelligent concept has shown a great efficacy in those areas by achieving states of arts in many areas where it was applied. Deep learning involves different types of deep networks that differ in architecture, working principles, and training algorithms. Stacked auto-encoder is one of the deep networks, that is basically a simple one as its depth can be controlled or extended based on the application complexity and difficulty where it is applied. This network is an encoder-decoder system that is used to extract some useful features from an image through its encoding unit and try to use those features in order to construct the same image at the as an output, through its decoding unit. In this work, we propose the use of a stacked autoencoder with two hidden layers in order to solve a computer vision problem, which is the deep humans identification using palmprints palm images. Network is trained and tested using images obtained from a public database available on the internet, which allowed the network to be capable of accurate identification of palmprints with small margins of errors. Experimentally, it was found that the designed stacked auto-encoder is efficient in solving a complex classification task such as the palmprint identification.

*Keywords:* Deep learning; deep networks; stacked auto-encoder; back propagation; palmprints identification; computer vision

#### ÖZET

Derin Öğrenme, bilgisayar vizyonundaki çeşitli alanlarda yaygın olarak kullanılmaktadır. Bu alanlar arasında görüntü analizi, segmentasyon, sınıflandırma, tanımlama vb. Vardır. Bu akıllı konsept, uygulandığı birçok alanda sanat devletlerine ulaşarak bu alanlarda büyük bir etkinlik göstermistir. Derin öğrenme, mimaride, calısma prensiplerinde ve eğitim algoritmalarında farklı olan farklı derin ağlar içerir. Yığılmış otomatik kodlayıcı derin ağlardan biridir, derinliği, uygulama karmaşıklığına ve uygulandığı yerde zorluğa bağlı olarak kontrol edilebildiği veya genişletilebildiği temel olarak basittir. Bu ağ, bir kodlama birimi aracılığıyla bir görüntüden bazı yararlı özellikleri çıkarmak için kullanılan ve kod çözme birimi aracılığıyla bir çıktı olarak aynı görüntüyü oluşturmak için bu özellikleri kullanmaya çalışan bir kodlayıcı-kod çözücü sistemidir. Bu çalışmada, avuç içi görüntülerini kullanarak derin insan tanımlaması olan bilgisayar görme problemini çözmek için iki gizli katmanı olan bir yığılmış otomatik kodlayıcının kullanılmasını öneriyoruz. Ağ, internette bulunan ve kamuya açık bir veri tabanından elde edilen görüntüleri kullanarak eğitilmiş ve test edilmiş olup, bu ağın küçük hata paylarıyla birlikte palmprintlerin doğru bir şekilde tanımlanmasını sağlamıştır. Deneysel olarak tasarlanan yığılmış otomatik kodlayıcının, palmprint tanımlaması gibi karmaşık bir sınıflandırma görevini çözmede etkili olduğu bulunmuştur.

Anahtar Kelimeler: Derin öğrenme; derin ağlar; yığılmış otomatik kodlayıcı; geri yayılımı; palmprints tanımlama; Bilgisayar görüşü

### TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
ABSTRACT	iii
ÖZET	iv
TABLE OF CONTENT	v
LIST OF TABLES	vi
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS	viii

### **CHAPTER 1: INTRODUCTION**

1.1 Introduction	1
1.2 Definition of the Problem	2
1.3 Objective and Scope of the Study	2
1.4 The Proposed Identification Process	2
1.5 Structure of the Thesis	4
	•••••

### **CHAPTER 2: LITERATURE REVIEW ON PALMPRINT IDENTIFICATION**

2.1 Palmprint Identification	5
2.2 Palmprint Features	5
2.3 Other Methods for Palmprint Identification	7
2.4 Neural Networks Methods for Palmprint Identification	9
2.5 Deep learning and Machine Learning Methods for Palmprint Identification	9

### **CHAPTER 3: DEEP LEARNING: STACKED AUTO-ENCODER**

3.1 Features Extraction and Segmentation in Image Processing.	
3.2 Patterns Recognition in Image Processing	14

3.3	Neural Networks	16
	3.3.1 Neural networks types	16
	3.3.2 Single layer perceptron	16

### **CHAPTER 4: PROPOSED METHODOLOGY**

4.1 Proposed Design for the Palmprint Identification System Using SAE	21
4.2 RGB to Grayscale Conversion	22
4.3 Image Denoising	22
4.4 Background Extraction	23
4.5 Images Addition	25
4.6 Intensity Adjustment	26
4.7 Edge Detection	26

### **CHAPTER 5: NETWORK SIMULATION AND PERFORMANCE**

5.1 Databases	
5.2 Proposed Network Architecture	
5.3 Network Training	31
5.4 Performance Evaluation	36
5.5 Results Discussion	
5.6 Results Comparison	

41
4

EFERENCES
-----------

### APPENDICES

Appendix 1: Image Processing Code	48
Appendix 2: Neural Networks Code	50

### LIST OF FIGURES

Figure 1.1: Palm features	3
Figure 1.2: Proposed system architecture for the recognition of plamprints	4
Figure 2.1: Geometric points and features of a palmprint	6
Figure 3.2: Architecture of artificial-neuron model (McCulloch and Pitts model)	8
Figure 3.3: Architecture of of Rosenblatt's Perceptron	19
Figure 4.1: Proposed palmprint framework	21
Figure 4.4: Grayscale conversion	22
Figure 4.5: Median filtering	23
Figure 4.6: Probing of an image with a structuring element	24
Figure 4.7: Background extraction	25
Figure 4.8: Image addition	25
Figure 4.9: Image adjustment	26
Figure 4.10: Canny edge detection	27
Figure 5.1: A sample of the palmprints of database 1	28
Figure 5.2: A sample of the palmprints of database 2	29
Figure 5.2: A sample of the palmprints of database	31
Figure 5.4: Sample of the training palmprint images	32
Figure 5.5: Training curve and reached mse during pre-training	34
Figure 5.6: Training curve and reached mse during fine-tuning	35
Figure 5.7: A sample of images used for testing the network	37
Figure 5.9: A sample of some misclassified images	38

### LIST OF TABLES

Table 5.1: Dataset 1, CASIA	29
Table 5.2: Dataset 2, PolyU	30
Table 5.3: Dataset 2 description and division	30
Table 5.4: Databases number of data for training and testing	32
Table 5.5: Network's training parameters	33
Table 5.6: learning results of the network	35
Table 5.7: Identification rate of the SAE during testing	37
Table 5.8: Results comparison	40

### LIST OF ABBREVIATIONS

ANN:Artificial Neural NetworkMLPMultilayer PerceptronBPNN:Back Propagation Neural networkSAE:Stacked Auto-encoderBPLABack Propagation Learning AlgorithmAE:Auto-encoderMSE:Mean Square Error

### CHAPTER 1 INTRODUCTION

#### **1.1 Introduction**

With the quickly improvement of the world's economy, the compounding of social request causes annihilation and savagery around the globe (Dai and Zhou, 2011). They have weighted the significance of security, and raised the developing interest for programmed identification frameworks. Then again, numerous strategies are connected in programmed identification frameworks, for example, get to controlling, ATM, PC information getting to, and so on (Cappelli et al., 2012).

For instance, a man dependably utilizes keys, passwords or access cards for character verification to get to the private records (or individual data), control puts or organized social orders. In any case, the keys and access cards might be lost by user himself or be unauthority replicated by others and the secret key may be overlooked by user himself or be known by other individuals (Dai et al., 2012).

Along these lines, individuals require an identification framework to distinguish the user's character without above detriments, or, in other words identification. Biometric identification framework has high productivity, high acknowledgment rate and agreeable to the user's working attributes, it is a programmed acknowledgment process dependent on an element vector got from a person's conduct or physiological qualities. The people's physiological attributes incorporate DNA, face, ear, fingerprint, walk, Iris, palmprints, voice, and so on (Dai et al., 2012; Cappelli et al., 2012; Jain and Demirkus; 2008).

A biometric identification framework should meet accuracy, speed, and asset necessity, be safe to the users, and be acknowledged by the expected individuals and vigorous to different assaults. A palmprint design is unique to the point that even twins have diverse palmprint designs; the example stays steady and settled for the duration of one's life. Contrasted with other physiological attributes, palmprint acknowledgment is considered the most feasible and dependable biometric acknowledgment strategy inferable from its benefits, for example, minimal effort, user invitingness, rapid and high accuracy (Jain and Feng, 2009).

#### **1.2 Definition of the Problem**

It is seen that an efficient tool to classify palmprints of humans in computer vision systems, nontime and resources consuming way will be beneficial for both security systems and researchers working in this field. Artificial neural networks (ANNs) have been used in computer vision by many researchers to various identification tasks, especially in human palmprints identification and estimation task (Dai and Zhou, 2011) (Cappelli et al., 2012). ANN is a data processing model which tries to imitate the way of human biological brain works. However, the previous ANNs models for palmprints identification have observed to not consider all features and moreover they consume long training time and less accuracy since they are based on the conventional backpropagation neural network. Thus, in this work, we attempted to investigate the use of deep network, named stacked auto-encoder (SAE) in the application of human palmprint identification. It is believed that this kind of deep network may result in a faster training time and a better accuracy due to its deep and hierarchical structure and training methods.

#### **1.3 Objective and Scope of the Study**

Based on previously mentioned researches, this study aims to employ a deep network, stacked auto-encoder (SAE) as an alternative in the identification task for the recognition and verification system of human's palmprints in a different way. The images of pamprints are first processed using image processing techniques such as filtering, and edge detection. Those methods help in extracting and highlighting the distinct features found in a palmprint. Hence, this makes the neural network classification task easier and faster as the unnecessary features are removed. Two large dataset of 772 subjects are used for training and testing the network performance. The total number of images of the two databases is 8102 which is fair enough for the deep network to converge and reach a very small error.

#### **1.4 The Proposed Identification Process**

Biometrics has been recently used for the human identification systems using the biological traits such as the fingerprints and iris scanning. Identification systems based biometrics show great

efficiency and accuracy in such human identification applications. The challenge in such systems is the segmentation of the region of interest; the region that has the key features that discriminate each human palmprint. Thus, this report aims to develop a human palmprint segmentation algorithm. The developed system uses images obtained from two public databases available on the internet (CASIA) (Sun et al., 2005) and PolyU palmprint database (Kumar, 2008). The proposed processing system is as follows: image filtering using median filter, image adjustment, image skeletonizing, edge detection using canny operator to extract features, clear unwanted components of the image.

Segmentation of the region of interest in a palm is a significant task in a human identification system bases on palmprint recognition. Therefore, in this work we plan on the identification of palmprints by the extraction of features in a palm using a deep neural network designed to extract the useful features. Figure 1 shows the most significant features in a palm.



Figure 1.1: Palm features (Sun et al., 2005)

Figure 1.2 shows the proposed architecture of the two-phase identification system that is designed to identify humans through their palmprints. As seen, the system is based on a stacked auto-encoder of two hidden layers and 4096 input neurons as the input images are of size 64\*64 pixels.



Figure 1.2: Proposed system architecture for the recognition of plamprints

#### **1.5 Structure of the Thesis**

This thesis is structured as follows:

- **Chapter one** describes in details the problem that is detected and studies in this thesis, as well as the scope, objective and the significance of the study.
- Chapter two includes theoretical background about palmprints identification. It also explains artificial neural networks and the most common types of these networks. Previous studies on palmprint identification with artificial neural networks are discussed in this chapter.
- **Chapter three** discusses the proposed identification system in addition to detailed explanation of the image processing techniques used in this work.
- **Chapter four** presents the training phase of the work in which network training performance is discussed
- **Chapter five** shows the network testing performance in addition to discussing the results and discussion of the thesis.
- Chapter six is a conclusion of the whole thesis.

#### **CHAPTER 2**

#### LITERATURE REVIEW ON PALMPRINT IDENTIFICATION

#### **2.1 Palmprint Identification**

Palm print acknowledgment is a biometric check procedure subject to the unprecedented instances of various traits in the palms of people's hands.

Palm print acknowledgment systems use a camera-based application, nearby related programming that methods picture data from a photograph of an individual's palm and ponder it to a set away record for that person. Palm prints are accomplices to fingerprints, including similar focal points. Like the case with unique mark checking, palm scanners use optical, warm or material procedures to draw out the focal points in the case of raised locales (called edges) and branches (called bifurcations) in an image of a human palm, nearby extraordinary unobtrusive components including scars, wrinkles and surface. Those three procedures rely upon evident light examination, warm release examination, and weight examination, separately. Palm scanners may require that individuals contact their hands to a screen or may be contactless.

Palm prints and fingerprints are consistently used together to enhance the exactness of unmistakable verification. An imprint, by uprightness of covering more skin locale, consolidates all the additionally recognizing focal points, making false positives everything aside from incomprehensible and at the same time making purposeful distortion generously progressively troublesome. In various conditions, for instance, criminal examinations, a full or midway palm print may at times be gotten when fingerprints are absent. A criminal may, for example, wear gloves to keep away from leaving fingerprints yet unexpectedly leave a partial palm print when a glove slips in the midst of the commission of a bad behavior.

#### **2.2 Palmprint Features**

As a rule, the geometry highlights, important line highlights and wrinkle highlights can be dictated by some picture preparing procedures from the picture.

Datum guide assurance: toward find the endpoints of every vital lines.

Properties of the central lines:

• Each central line meets the side of the palm at approx. right point when it streams out of the palm;

• The life line is situated at within part of the palm which bit by bit slopes to within the palm in parallel toward the start;

- Most of the existence line and the take line stream off of the palm at a similar point
- The endpoints are nearer to the fingers than the wrist.



Figure 2.1: Geometric points and features of a palmprint (Kora et al., 2009)

• Geometric highlights, for example, the width, length and territory of the palm. Geometric highlights are a coarse estimation and are moderately effectively copied. In themselves they are not adequately particular;

• Line highlights, vital lines and wrinkles. Line highlights distinguish the length, position, profundity and size of the different lines and wrinkles on a palm. While wrinkles are exceedingly unmistakable and are not actually copied, essential lines may not be adequately particular to be a solid identifier in themselves; and

• Point highlights or particulars. Point highlights or details are like unique mark Minutiae and distinguish, among different highlights, edges, edge endings, bifurcation and dabs (Kora et al., 2009).

#### **2.3 Other Methods for Palmprint Identification**

Palm print acknowledgment normally realizes a significant part of the equal coordinating characteristics that have allowed unique finger impression acknowledgment to be a champion among the most extraordinary and best reported biometrics (Jia et al., 2008).

Both palm and finger biometrics are addressed by the information presented in a contact edge impression. This information joins edge stream, edge characteristics, and edge structure of the raised section of the epidermis. The data addressed by these rubbing edge impressions allows an affirmation that relating districts of disintegration edge impressions either began from a comparable source or couldn't have been made by a comparative source (Aoyama et al., 2013).

Since fingerprints and palms have both uniqueness and constancy, they have been used for over a century as a trusted in kind of conspicuous verification. In any case, palm acknowledgment has been slower in getting the opportunity to be automated in light of a couple of confinements in enrolling limits and live-look at headways (Jain et al., 2016).

Palm recognizing verification, much equivalent to unique mark conspicuous confirmation, relies upon the aggregate of information presented in a pounding edge impression. This information joins the surge of the disintegration edges (Level 1 Detail), the proximity or nonattendance of features along the individual pounding edge ways and their game plans (Level 2 Detail), and the astounding inconspicuous component of a singular edge (Level 3 detail) (Jia et al., 2008)..

Palm acknowledgment advancement manhandles a segment of these palm features. Grinding edges don't by and large stream constantly all through a model and consistently result specifically properties, for instance, completing edges or parceling edges and touches (Jain et al., 2016). A palm acknowledgment system is planned to disentangle the surge of the general edges to dole out a request and after that think the points of interest detail — a subset of the total whole of information available, yet enough information to effectively glance through an immense

storage facility of palm prints. Points of interest are obliged to the region, course, and presentation of the edge endings and bifurcations (parts) along an edge way.

An assortment of sensor composes: capacitive, optical, ultrasound and warm, can be utilized for gathering the computerized picture of a palm surface; nonetheless, customary live-check systems have been ease back to adjust to the bigger catch regions required for digitizing palm prints. Difficulties for sensors endeavoring to achieve high-goals palm pictures are as yet being managed today (Zbontar and LeCun, 2016). A standout amongst the most widely recognized methodologies, which utilizes the capacitive sensor, decides every pixel esteem dependent on the capacitance estimated, made conceivable on the grounds that a zone of air (valley) has fundamentally less capacitance than a region of palm (edge).

Some palm recognition frameworks filter the whole palm, while others require the palms to be divided into littler zones to advance execution. Amplifying unwavering quality inside either a fingerprint or palm print framework can be incredibly enhanced via looking littler informational indexes (Wu et al., 2006). While fingerprint frameworks frequently segment vaults dependent on finger number or example order, palm frameworks segment their storehouses dependent on the area of a grinding edge zone. Inert inspectors are extremely gifted in perceiving the bit of the hand from which a bit of proof or dormant lift has been obtained. Seeking just this area of a palm vault instead of the whole database amplifies the unwavering quality of an inactive palm look.

Like fingerprints, the three principle classes of palm matching strategies are particulars based matching, relationship based matching, and edge based matching. Particulars based matching, the most generally utilized system, depends on the details focuses depicted above, particularly the area, bearing, and introduction of each point. Relationship based matching includes basically arranging the palm pictures and subtracting them to decide whether the edges in the two palm pictures compare. Edge based matching uses edge design milestone highlights, for example, sweat pores, spatial traits, and geometric qualities of the edges, and additionally neighborhood surface investigation, which are all exchanges to particulars trademark extraction. This technique is a quicker strategy for matching and beats a portion of the troubles related with removing details from low quality pictures (Wu et al., 2006).

#### 2.4 Neural Networks Methods for Palmprint Identification

Palmprint based individual confirmation has picked up inclination over other biometric modalities because of its simplicity of procurement, high client acknowledgment and reliability.

Authors in (But et al., 2008) presented a novel research about the identification of palmprint using an approach that uses textural information that can be found on palmprints. The authors extracted these features using contourlet transform (CT).

The works uses iterated directional filterbanks in order to extract the region of interest which represented the palm features and the two dimensional spectrums which was divided into four slices using the same method. The authors proposed algorithm was capable of capturing the local and global features in a palmrint using a method called normalized Euclidean distance classifier. This work was tested 7752 palm images and the experimental results showed that this method is feasible as it achieved an accuracy of 88.91%.

Moreover, another work by (Krishneswari, 2011) proposed a new method for palmprint authentication. This work was based on the extraction of intramodal features of palmprints using wavelet. Authors of this work used many phases of their system in order to extract the region of interest and then to extract the palm features. First, image acquisition and pre-processing are used, and the feature extraction and fusion using wavelets transform were applied to the extracted features. Finally, those extracted features were classified using neural network classifier. The system was tested on 200 users and the experimental results demonstrate the robustness of their work which achieved 93.1% accuracy.

Biometrics verification is a powerful strategy for consequently perceiving a man's character with high certainty.

#### 2.5 Deep learning and Machine Learning Methods for Palmprint Identification

Image processing and manual features extraction of images can be so complex and time consuming. Therefore, there was a big need of networks that can extract features from images automatically through its layers. This was the motivation of creating deep learning networks. The

depth of networks is made to extract low and high level features without any feature engineering techniques.

Many deep networks were created, however, the best deep network for features extraction from domain space is the deep convolutional neural network (DCNN). This is due to its depth in which convolutions, pooling, regularization, and normalization are applied to images, which allows the extraction of different levels of abstractions of input data.

In practice, the training of deep convolutional neural networks created from scratch is a tedious task. This is because CNNs are deep, which means many hyperparameters to be trained in addition to filters learning and weights update and calculation of errors which requires long time. Moreover, CNNs need large datasets in order to be trained and to not overfit. This can be an issue since it is relatively difficult to find some large datasets especially in the medicine field.

Recently, deep networks architectures are presented. Those networks are convolutional neural networks with different architectures and number of layers such as AlexNet, VGG-NET, GoogleNet etc... These networks are trained on ImageNet; a public dataset of millions of general images used to train the new models to classify 1000 classes. After training those models have obtained great generalization capabilities in classifying 1000 objects.

Chen et al. (2006) play out a two dimensional double tree complex transform on the preprocessed palmprints to disintegrate the pictures. Double tree complex changes are proposed to determine the shortcoming of customary wavelet change, or, in other words invariant, for example recognition. At that point they apply Fourier Transform on each subband and view the range size as features. At long last, SVM is utilized as a classifier.

Chen et al. (2006) remove a progression of nearby features (e.g. normal power) along a winding and utilize a period arrangement strategy called emblematic total guess to speak to the features and least separation to think about two component vectors.

Doi et al. (2003) respect the crossing point purposes of finger skeletal lines and finger wrinkles and the convergence purposes of the broadened finger skeletal lines and key lines as highlight focuses (Doi et al., 2003). Notwithstanding position data, the distracting edges between the vital lines and the broadened skeletal lines are likewise considered as features. They utilized root mean square deviation to gauge the contrasts between two features. Han extricated seven determined line profiles from preprocessed palmprints and three fingers and utilized wavelets to process low recurrence data (Han, 2004). This data is framed as another element vector, whose dimensionality is decreased by PCA. At last, summed up learning vector quantization and ideal positive Boolean capacity 12 are utilized to settle on ultimate choice. This work might be the main paper utilizing highlight level combination for palmprint recognition.

Hennings et al. (2004) utilize Log-Gabor channels to dole out line-content scores to various areas of palmprints. A particular number of districts with best line content scores are chosen to prepare relationship channels. They utilize ideal tradeoff manufactured discriminant work (OTSDF) channel as a classifier. Connection channel is a kind of classifiers, broadly considered by Kumar and his associates (Kumar et al., 2005).

To upgrade check execution, they make utilization of a few client particular strategies (e.g. user specific division and client particular limit). Koichi et al. additionally propose a relationship approach (lto, et al., 2006).

The sufficiency range of two portioned pictures is utilized to gauge their rotational and scale contrasts. One of the pictures is pivoted and scaled and after that their abundancy data in the recurrence area is evacuated. At last, band-restricted stage just connection (BLPOC) is utilized to register the likeness of two pictures. BLPOC just thinks about low to center recurrence data. Zhang et al. utilized complex wavelets to break down palmprint pictures and propose an altered complex-wavelet basic similitude (CW-SSIM) list for estimating the nearby closeness of two pictures (Zhang et al., 2007).

The general likeness of two palmprints is evaluated as the normal of all nearby altered CW-SSIM. CW-SSIM is initially proposed for assessing picture quality (Wang et al., 2004).

#### **CHAPTER 3**

#### **DEEP LEARNING: STACKED AUTO-ENCODER**

#### 3.1 Features Extraction and Segmentation in Image Processing

Segmentation is a partitioning of an image so that a particular region is extracted or segmented. However, this cannot be easily achieved, as it depends on some properties of the image or the region that should be detected such as edges, shapes, textures, intensities etc..

Over the past decades, different and many algorithm were developed for segmentation purposes in medical images (Fu and Mui, 1981) (Pal and Pal, 1993) (Koshana, 1994) (Lucchese and Mitra, 2001). Those approaches are all based on different properties of images. those properties can be the points, regions edges, objects or regions etc..

#### • Algorithms based on the points properties.

This algorithm is based on detecting a point in a homogeneous part of the image. This is achieved by analyzing some properties of the point such as colour, brightness, intensity and other characteristics. The drawback of this algorithm is the difficulties in selecting the important and useful features in images that have many homogenous segments of similar point characteristics. Many researches have used these approaches for segmenting medical images (Sharma et al., 2010)(Withey and Koles, 2007)(Zhang and Wang, 2000).

#### • Algorithms based on the edge detection.

This algorithm is very popular for segmentation, in particularly, in medical field where a certain region segment in the image needs to be extracted (Aroquiaraj and Thangavel, 2013) (Wu et al., 2015) (Sahakyan and Sarukhanyan, 2015). Edges in an image are the changes and discontinuities in intensities of the image pixels. Hence, this approach works mainly on the images which have brightness or intensity changes on its region edges. Thus, detecting these intensity changes can lead to segmentation of the region edges which for an object in an image.

Researchers have used various algorithms for segmenting the breast tumorous cells in histological images. The authors in (Erezsky et al., 2015) reviewed different segmentation

algorithms such as K-means, Watershed, and texture segmentation. These 3 techniques were applied to breast cell images and the signal to ration for each technique was calculated. Moreover, the authors proposed their own technique for breast cells segmentation which is based on detecting the properties of point connections. Moreover, the authors claimed that their proposed method yielded better segmentation results and lesser signal to noise ration compared to other discussed techniques.

Another breast cancer cell segmentation and contouring algorithm is proposed in (Mouelhi et al., 2011). In their work, an algorithm for segmenting the breast cancer cells is based on watershed and concave vertex graph as a next stage since the segmentation here occurs on many stages. At first, the malignant cells are detected using the geodesic active contour. Then high concavity points are taken from the cell contours to be then used for selecting the clustered cell regions only. Secondly, the touching cells regions are first segmented using watershed technique and then a concave vertex graph is constructed. This shows the inner edges and concave points which helps in separating cells regions. Finally, the authors of this work showed that their algorithm is very accurate in breast cancer cells segmentation without losing geometrical features.

An algorithm for the tumor cells detection breast cells microscopic images is proposed in (Phukpattaranont and Boonyaphiphat, 2006). The algorithm is comprised of two processing stages. The first one is the segmentation of breast cells using watershed mathematical process. Second, the breast cells are extracted or described using Fourier transform descriptors and the principal components analysis is performed to classify cells into normal or cancerous cells.

Moreover, authors in (Vahadane and Sethi, 2013) improved the watershed segmentation algorithm to detect breast cancer cells in histological images using nuclear segmenation. Their algorithm is based on many image processing techniques such as image enhancements and Ostu's tresholding in addition to the fast radial symmetry transform (FRST) for the nuclei extraction and foreground seeds generation.

Guassian smoothing is first used to remove the high frequency noise and the blurred nuclei segmentation. Then, background markers are used based on the image information to reduce the over-segmenation. FRST is also used to extract nuclei and to form foreground seeds. Finally,

post-processing takes place by using erosion and dilation which results in segmenting the cell nuclei.

#### **3.2 Patterns Recognition in Image Processing**

Pattern recognition is the process of developing systems that have the capability to identify patterns; while patterns can be seen as a collection of descriptive attributes that distinguishes one pattern or object from the other. It is the study of how machines perceive their environment, and therefore capable of making logical decisions through learning or experience. During the development of pattern recognition systems, we are interested in the manner in which patterns are modeled and hence knowledge represented in such systems. Several advances in machine vision have helped revamp the field of pattern recognition by suggesting novel and more sophisticated approaches to representing knowledge in recognition systems; building on more appreciable understanding of pattern recognition as achieved in the human visual processing.

Typical pattern recognition as the following important phases for the realization of its purpose for decision making or identification.

- Data acquisition: This is the stage in which the data relevant to the recognition task are collected.
- Pre-processing: It is at this stage that the data received in the data acquisition stage is manipulated into a form suitable for the next phase of the system. Also, noise is removed in this stage, and pattern segmentation may be carried out.
- Feature extraction/selection: This stage is where the system designer determines which features are significant and therefore important to the learning of the classification task.
- Features: The attributes which describe the patterns.
- Model learning/ estimation: This is the phase where the appropriate model for the recognition problem is determined based on the nature of the application. The selected model learns the mapping of pattern features to their corresponding classes.
- Model: This is the particular selected model for learning the problem, the model is tuned using the features extracted from the preceding phase.

- Classification: This is the phase where the developed model is simulated with patterns for decision making. The performance parameters used for accessing such models include recognition rate, specificity, accuracy, and achieved mean squared error (MSE).
- Post-processing: The outputs of the model are sometimes required to be processed into a form suitable for the decision making phase stage. Confidence in decision can be evaluated at this stage, and performance augmentation may be achieved.
- Decision: This is the stage in which the system supplies the identification predicted by the developed model.

There exist several approaches to the problem of pattern recognition such as syntactic analysis, statistical analysis, template matching, and machine learning using artificial neural networks.

Syntactic approach uses a set of feature or attribute descriptors to define a pattern, common feature descriptors include horizontal and vertical strokes, term stroke analysis; more compact descriptors such as curves, edges, junctions, corners, etc., which is termed geometric features analysis. Generally, it is the job of the system designer to craft such rules that distinguish one pattern or object from another. The designer is meant to explore attribute descriptors which are unique to identify each pattern, and where there seems to a conflict of identification rules such as can be observed in identifying Figure 6 and 9; they have same geometric feature descriptors save that one is the inverted form of the other, the system designer is meant to explore other techniques of resolving such issues (Yumusak and Temurtas, 2010).

Statistical pattern analysis uses probability theory and decision to infer the suitable model for the recognition tasks.

Template pattern matching uses the technique of collecting perfect or standard examples for each distinct pattern or object considered in the recognition task. It is with these perfect examples that the test patterns are compared. It is usually the work of the system designer to craft the techniques with which pattern variations or dissimilarities from the templates are measured, and hence determine decision boundaries as to accept or reject a pattern being a member of a particular class. Euclidean distance is a common used function to measure the distance between two vectors in n-dimensional space.

Template matching can either be considered as global or local depending on the approach and aim for which the recognition system is designed. In global template matching, the whole pattern for recognition is used to compare the whole perfect example pattern; whereas in local template matching, a region of the pattern for classification is used to compare a corresponding region in the perfect template.

Artificial neural networks, on the other hand, are considered intelligent pattern recognition systems due to their capability to learn from examples in a phase known as training. These systems have sufficed in lots of pattern recognition systems; the ease with which same learning algorithms can be applied to various recognition tasks is motivating.

In this approach, the designer is allowed to focus on determining features to be extracted for learning by the designed systems, rather than expending a huge amount of time, resources, and labour in understanding the whole details of the application domain; instead, the system learns relevant features that distinguish one pattern from the other (Yumusak and Temurtas, 2010).

#### **3.3 Neural Networks**

Artificial Neural Networks (ANNs) can be defined as a data processing model which tries to imitate the way of human biological brain works. There are many nodes (neurons) that linked or connected with each other through lines (weight) in ANNs; these neurons work with each other to find solution for specific tasks. The processes of neural networks (NN) consist of two steps; the first step is training or learning of neural network through use of data (examples) which can be carried out by using learning algorithm. Whereas, the second step is recalling; this step means testing the trained network for new given data (examples). However, the structure, properties of neurons and training methods are factors that affects classification of neural networks or specify the type of neural network. The most common types of neural network are listed below (Haykin, 2009; Du and Swamy, 2013; Kriesel, 2007; Tino et al., 2015; Gurney, 1997).

#### **3.3.1** Neural networks types

A Feed-Forward Neural Networks (FFNNs): are the most commonly used type of neural networks. FFNNs consist of three types of layers (inputs layer, hidden layer and output layer). the structure of FFNNs is sorted by the type of layers, such as the first layer is input layer and last layer is the output layer, whereas the middle layers (located between input and output layer) can be called as hidden layers, which can be one or more layers. Moreover, in FFNs, the neurons are connected to the following layer neurons by one-direction lines (weights). In other words, there is no feed-back connection in FFNN and the neurons of same layer are not connected with each other. The most common types of Feed-Forward neural networks are listed below (Haykin, 2009; Du and Swamy, 2013; Kriesel, 2007; Tino et al., 2015; Gurney, 1997).

- a) Multilayer perceptron
- b) Radial basis function network

Recurrent neural network: is a less conventional type of neural network. The architecture of this network allows feed-back connection between neurons. Further, minimum amount of feed-back connection between neurons in this network must one feed-back connection. Also in this network, the neurons of same layer can be connected with each other. The commonly used types of Recurrent neural network are listed below (Haykin, 2009; Du and Swamy, 2013; Kriesel, 2007; Tino et al., 2015; Gurney, 1997).

- a) Hopfield network
- b) Boltzmann machine.

#### **3.3.2 Single layer perceptron**

It is artificial neuron model that can be defined as a mathematical model of a biological neuron with several inputs  $(x_1, x_{j1})$  and one single output (y). Furthermore, McCulloch and Pitts model also can be referred as a simple neuron paradigm that gathers input patterns and assign them as input parameters through the associated parameters of the weights. In other words, linear threshold system is a neuron that can operates all the number of inputs from another units and form an actual values, this process is performed in accordance to the activation function. The transfer function performs mapping from the input (real values) to the output (into interval); this

mapping can be a linear or nonlinear. The sigmoidal function (hard-limiter) was used in McCulloch and Pitts model as transfer function, which referred by ( $\emptyset$ ). The synapses in artificial neuron model is referred as weights (w) which is the connection lines between inputs and neuron. Moreover, in McCulloch and Pitts model the values of the weight (w) and threshold ( $\theta$ ) were fixed. Artificial neuron model can easily classify inputs set into two various classes (which means the output is binary). The output (y) in artificial neuron or McCulloch and Pitts model is specified by summation of the dot product between weight and input parameters (w<sub>i</sub>. x<sub>i</sub>) with respect to the activation function  $\emptyset$  (Haykin, 2009; Du and Swamy, 2013; Gurney, 1997).

$$\mathbf{N} = \sum_{i=1}^{J_1} w_i \, x_i - \, \theta = w^T \mathbf{x} - \theta \tag{1}$$

 $\mathbf{y} = \boldsymbol{\emptyset}\left(\boldsymbol{N}\right) \tag{2}$ 



Figure 3.2: Architecture of artificial-neuron model (McCulloch and Pitts model) (Du and Swamy, 2013)

N = network of artificial neuron model, whereas, x is the input parameters.

w represents the weight or the connection lines between inputs and transfer function.

 $\emptyset$  is the activation function (sigmoidal).

 $\theta$  is the threshold which is an attribute uses to move the decision boundary away from the origin.

In 1957, the first perceptron (single-layer perceptron paradigm) was developed by Rosenblatt which was inspired by McCulloch & Pitts model and the idea of Hebb (Hebbian learning rule). Rosenblatt's Perceptron model has the capability to classify inputs set into more than two classes unlike artificial neuron (McCulloch & Pitts) model which can only classify inputs set into two classes. In single-layer perceptron model, different activation functions ( $\emptyset$ ) have been used such as a bipolar. Also, the weights (w) and thresholds or biases ( $\theta$ ) is calculated analytically or by a

learning algorithm. However, the output ('y) of single-layer perceptron can be written as fallowing (Haykin, 2009; Du and Swamy, 2013; Fausett, 1994; Tino et al., 2015).

1

$$N = w^T x - \theta \tag{3}$$

$$\mathbf{y} = \boldsymbol{\emptyset}\left(N\right) \tag{4}$$



Figure 3.3: Architecture of Rosenblatt's Perceptron (Du and Swamy, 2013)

Single-layer perceptron has capability only to find solution for linear separable problems. The weight between neurons can be adjusted by using learning algorithm (Rosenblatt's perceptron convergence theorem) and this can be driven through error equation  $(E_{t, j})$ . Moreover, the learning algorithm of perceptron can be written as following:

$$N_{t,j} = \sum_{i=1}^{J_1} x_{t,i} w_{i,j} (t) - \theta_j = w_j^T x_t - \theta_j$$
(5)

$$'\mathbf{y}_{t,j} = \begin{cases} 1 & N_{t,j} > 0 \\ 0 & otherwise \end{cases}$$
(6)

$$E_{t,j} = y_{t,j} - y_{t,j}$$
 (7)

$$w_{ij}(t+1) = w_{ij}(t) - n \ x_{t,i} \ E_{t,j}$$
(8)

N = network of single-layer perceptron, whereas,  $x_{t,I}$  is the *i*th input of the *t*th example.

 $w_{ij}$  is the *ith* weigh at the *t*th node (stand for connected lines between neurons).

 $\theta$  is the bias or threshold for neuron. while,  $\emptyset$  is the transfer or activation function.

 $E_{t,j}$  is denote to the error.

- $y_{t, i}$  is referred to the real output (desired).
- ' $y_{t, i}$  is the actual output (predicted from network) (Du and Swamy, 2013).

### CHAPTER 4 IMAGE ANALYSIS PHASE

#### 4.1 Proposed Design for the Palmprint Identification System Using Stacked Auto-Encoder

In this thesis, a stacked auto-encoder for the identification of plamrpints is presented. The proposed work is a combination of both image analysis and neural network classification. In the first phase image processing is employed in order to extract the rightful features that identify each palmprint. Different image processing techniques are used in a manner that the important features are extracted. Images are first enhanced using median filtered as they may have some noise. The filtered images then undergo some morphological techniques in which the background of the image is removed and then added to the filtered image. This addition operation results in a brighter image where features are clearer and smother. At last, the edges are extracted from input images using the method called edge detection using Canny operator.



Figure 4.1: Proposed palmprint framework

Image processing has been extensively used in various areas in medicine. Those areas include medical image diagnosis, segmentation, enhancement etc... image segmentation is needed in this field as it helps in detecting or contouring regions of interest in some images where a specific objects should be segmented. In this thesis, we apply image segmentation based images processing for the segmentation or detection of palm features in palmprint images. The system is

based mainly on different image processing techniques that end up by segmenting the some features that cud help in the classification of palmprints.

#### 4.2 RGB to Grayscale Conversion

The initial step is to change image to grayscale type. This transformation is done utilizing the luminosity technique which depends on the commitment of every shade of the three RGB colors (Pitas and Venetsanopoulos, 1990).



Original image

Grayscale image



#### **4.3 Image Denoising**

The median filter has been ended up being extremely helpful in numerous image processing applications. In a median filter, a window slides over the information and the median estimation of the pixel values inside the window is calculated to be the output of the filter. This nonlinear filter has more advantages comparing to linear ones since it helps in preserving edges. Moreover, it has the property of noise attenuation against the impulsive-sort noise (Pitas and Venetsanopoulos, 1990; Bovik et al., 2011).

Figure 4.3 represents an illustration of the median value calculation of an apple image.



Grayscale image

Median filtered image

Figure 4.3: Median filtering

#### 4.4 Background Extraction

Morphological operations are utilized for the separating of background of the image. Morphology can be characterized as image processing analysis tools that procedure images in light of shapes (Ortiz and Torres, 2004). These operations should be possible by applying a structuring element to an input image, bringing about an output image of the same size.

Morphological tasks tests a picture with a little shape or supposed an "organizing component". The structure component is a matrix involves 0's and 1's, the place the 1's are known as the neighbors The organizing component is put at all conceivable areas in the info picture so as to be contrasted and the comparing neighborhood of pixels. Thus, based on some operations it will be recognized whether the structure element fits or intersects within the neighbourhood as shown in Figure 4.4.



Figure 4.4: Probing of an image with a structuring element

Structure element has numerous shapes as indicated by its application. Image opening is utilized to extract the background of image in the proposed framework. Morphological opening is erosion trailed by dilation utilizing the same structure component for both operations. The opening method can uproot objects that can't contain the structure component all together then to separate the background (Ortiz and Torres, 2004; Pratt, 2001).

Dilation is utilized to individually expel or include a pixel at question limit dependent on organizing component shape and sweep. Erosion is a change of shrinking, which diminishes the grayscale estimation of the image (Priya et al., 2005; Jagadeesh et al., 2013). The formula of finding the output pixel in both operations is the maximum of input pixels neighborhood matrix.

Image opening is used to extract background in the proposed system. It is erosion followed by dilation using the same structure element for both operations. Generally, it is used to smooth the edges of an image, in addition to remove gaps where the structuring element cannot be contained in order then to extract the background (Ortiz and Torres, 2004; Pratt, 2001).



Median filtered image

Exracted background

Figure 4.5: Background extraction

#### 4.5 Images Addition

During this operation; the background image is added to original grayscale image. This addition operation is achieved by simply adding each pixel in the first image to its corresponding pixel in the second image (Pitas and Venetsanopoulos, 1990). This leads to an increase in the intensity of pixels; therefore, the region of interest (defective area) gets brighter since the pixels under this region originally have higher intensities than other pixels. Figure 4.6 illustrates the result image after adding the two images (original and background image).



Median filtered image

Exracted background



#### 4.6 Intensity Adjustment

The image obtained from the addition of original and background image undergoes intensity adjustment in which the information image's intensities are mapped to another scope of intensities in the output image. This should be possible by setting the low and high information intensity esteems that ought to be mapped and the scale over which they ought to be mapped (Figure. 4.7) (Gonzalez and Woods, 2002).



Added image

Adjusted image

Figure 4.7: Image adjustment

#### **4.7 Edge Detection**

Edge detection can be defined as an image processing technique for finding the boundaries of an object in an image. Basically, this technique works by detecting the discontinuities in the intensities of pixels. This allows to detect the edges which are the discontinuities in intensities between two pixels (Gonzalez and Woods, 2002; Ortiz and Torres, 2004). Edge detection can be used for segmentation of objects in images and also for data extraction from images. Many algorithms were proposed for edge detection; each is used based on the application. However, the most common used is called Canny edge detection in which image are filtered and then edges are detected.

In this work, Canny detector is used for detecting the edges of the flaws found in the adjusted images. Figure 4.8 shows the canny edge detection of the adjusted apple image.



Adjusted image



Edge detected (segmented) image



#### **CHAPTER 5**

#### NETWORK SIMULATION AND PERFORMANCE

#### 5.1 Databases

Images can be described as the "food" of the neural networks. More images mean the smarter and more accurate network will be. Thus, the first step in developing a neural based system is to find a good and public database which will be used for training the network. In this work, pamprint mages of different shapes are needed to train our system to be capable of identifying the humans by checking their palmprints. Therefore, the best palmprints database was chosen for this task. The first used database is called CASIA; a public Palmprint Image Database V1.0 (CASIA database) (Zhou et al., 2014).

This Palmprint Image Database contains 5,502 palmprint images captured from 312 subjects. For each subject, palmprint images from both left and right palms are collected. All palmprint images are 8 bit gray-level JPEG files by our self-developed palmprint recognition device (as shown in Figure 5.1).



Figure 5.1: A sample of the palmprints of database 1 (Zhou et al., 2014)

This database contains 5502 images for 312 different persons. Note that each of these subjects has 15-17 images of left and right palms.

<b>Table 5.1:</b>	Dataset	1,	CASIA
-------------------	---------	----	-------

Dataset 1	Number of subjects	Training	Testing
5,502	312	4000	1502

As seen in the table 5.1, the first employed database contains 5502 images which are split in two parts. One for training and contains 4000 images and other for testing and has 1502 images

Another database is also used for collecting more images to feed the neural networks. The second dataset is called The IIT Delhi palmprint image database (Sun et al., 2017), which consists of the hand images collected from the students and staff at IIT Delhi, New Delhi, India. This dataset is collected from a total of 235 users, and all the images are in bitmap (\*.bmp) format. All the subjects in the database are in the age group 12-57 years. Seven images from each subject, from each of the left and right hand, are acquired in varying hand pose variations. Each of the subject is provided with live feedback to present his/her hand in the imaging region. The touchless imaging results in higher image scale variations. The acquired images have been sequentially numbered for every user with an integer identification/number.

The resolution of these images is 800\*600 pixels and all these images are available in bitmap format. In addition to the original images, 150\*150 pixel automatically cropped and normalized palmprint images are also available. Figure 5.2 shows a sample of the database 2



Figure 5.2: A sample of the palmprints of database 2 (Sun et al., 2017)

Table below shows the description of the database 2.

Dataset 2	Number of	Number of	Total	Total	Total
	Right hands	left hands	number of	number of	Number of
	per person	per person	right hands	left hands	images
2600	6	6	1300	1300	2600

Table 5.2: Dataset 2, PolyU

This database consists of 2600 images of palms. Those images are collected from 460 persons in which 230 are left hands and 230 are right hands. Table 5.3 shows the number of images used for training and testing the network.

Dataset 2 Number of Total Training Testing Total subjects training testing images images 4 images of 2 images of 920 2600 235 1840 each person each person

Table 5.3: Dataset 2 description and division

#### **5.2 Proposed Network Architecture**

In this study, an original research for the identification of palmprints using deep learning is presented. A stacked auto-encoder is selected to be employed for the identification of palmprints in this work. This selection of SAE was the results of many researchers conducted in the field of palmprints identification where all networks were either backpropagation neural networks or convolutional neural networks. Thus, it was found that there is a big need of employing the stacked auto-encoder to investigate and evaluate its performance in identifying humans through their palmprints, which is considered as a tough classification task. Therefore, due to its difficulty, a pre-processing phase was used before feeding images into network. This may help in spotting the important and unique features of palmprints before being fed into network. This can help in the learning phase of the stacked auto-encoder which therefore results in a better identification performance and accuracy.

The employed stacked auto-encoder is designed for the proposed task is built of two autoencoders stacked together to create a bigger SAE composed of two hidden layers. The autoencoder was first trained layer by layer using greedy layer wise training until a network of two hidden layer, one input, and one output network is formed. Therefore, these trained autoencoders were all stacked together and the proposed stacked auto-encoder is formed.

The proposed architecture of the proposed network is presented in Figure 5.3.



Figure 5.3: Proposed stacked auto-encoder architecture for the identification of palmprints

Note that in this work, the images are first segmented using image processing techniques explained in chapter 4, before being fed into network as shown in Figure 5.3. As seen the input images are of size 256\*256 pixels, however after processing the images size is reduced to 64\*64 pixels for faster computations and processing.

#### **5.3 Network Training**

The training of the deep model SAE is discussed in this section. The simulation of this work was using Matlab 2017b version. Note that the SAE is trained on palmprint processed images with enhancement and segmentation. It is important to mention that SAE is trained on both databases used in this work as shown in table 5.4. As seen in table 5.4, the total number of images of both databases is 8102 images, in which 5840 and 2422 images are used for training and testing, respectively.

Databases	#data	#Subjects	Training	Testing
CASIA	5,502	312	4000	1502
IIT Delhi	2600	235	1840	920
Touchless				
Palmprint				
Database				
Total Number of images	8102	547	5840	2422

Table 5.4: Databases number of data for training and testing

For output classes the network has 547 neurons as the number of subjects is 547. Note that the network is first pre-trained as it is deep network, which means that each layer should be trained separately according to the greedy layer-wise training algorithm (Hinton, 2006). This is called pre-training and in this phase outputs are not labeled in order to let the network learn to reconstruct its input from its features extracted in its hidden layers. Therefore, in the pre-training phase the number of output neurons is equal to the number of input neurons which is 4096. Figure 5.4 shows a sample of the training palmprint images.



Figure 5.4: Sample of the training palmprint images

Once the networks finish pre-training, it is then fine-tuned using the conventional backpropagation algorithm. Here, the input images are labeled therefore; output neurons are 547 which means that network is being trained to classify the palmprints into 547 classes or subjects.

Table 5.5 shows the training parameters of the stacked auto-encoder during both pre-training and fine-tuning.

Learning parameters	Values (Pre-training)	Values (Fine- training)
Number of training images	5840	5840
Number of layers of the network	4	4
Number of hidden layers	2	2
Learning rate	0.22	0.19
Maximum number of iterations	100\100	400\400
Transfer function	Sigmoid	Sigmoid

	37. 14	· · ·	
Table 5.5:	Network's	training	parameters
	1		p

The learning curve of the network during pre-training is shown in Figure 5.5. It can be seen that the SAE training error decreases with the increase of number of iterations; however, it couldn't reach a very small error (0.0846).



Figure 5.5: Training curve and reached mse during pre-training

The Figure 5.6 depicts the learning of the network SAE during fine-tuning. It is also seen that the network error is diminishing sharply and it reaches a very small error of 0.0048 at iteration 400 which indicates a good learning results of the network during this stage. Note that the network required 2 minutes and 20 seconds to fine tune and converge.



Figure 5.6: Training curve and reached mse during fine-tuning

Table 5.6 shows the learning performance of the stacked auto-encoder during pre-training and fine-tuning.

Learning results	Pre-training	<b>Fine-tuning</b>
Number of training images	5840	2422
Training recognition rate	97%	100%
Minimum square error achieved	0.0846	0.00489
(MSE)		
Iterations required	100	400
Training time	77 secs	140 secs

**Table 5.6:** learning results of the network

Table 5.6 presents the performance of the SAE during pre-training and fine-tuning. It is seen that the network SAE performed very good during fine-tuning where it achieved a high recognition rate of 91.2% with a training time of 140 seconds and with small number of iterations (400). Moreover, it is seen that the network reached a very small error during fine-tuning (0.00489). However, the network couldn't achieve similarly in the pre-training since it achieved a high recognition rate (100%) with less time (77 secs) and higher error margins (0.0846) than that obtained during fine-tuning. This performance is good because the network performance is evaluated in the fine-tuning stage where it is trained to classify however, in the pre-training the network is just trying to get the good and right weights and extract the significant features that can be used on the fine-tuning.

#### **5.4 Performance Evaluation**

After training and convergence the network was tested in order to evaluate its capability if generalization on unseen images. The stacked auto-encoder was tested on 2522 images from the two databases. Figure 5.7 shows a sample of some testing palmprint that were used in testing the network.

Table 5.4 shows the identification results in the testing phase. It shows the accuracy of the trained stacked auto-encoder in generalizing the accurate identification of palmprint images that have not been seen in the training phase.



Figure 5.7: A sample of images used for testing the network

From Table 5.7, it can be seen that the stacked auto-encoder which was trained on 4000 processed palmprint images achieved has performed well with an accuracy of 93%.

Note that the accuracy is calculated as follows:

$$Accuracy = \frac{C}{N}$$

Where N is the total number of correctly classified images during the training and/or testing phases, white N is the total number of images.

Deep Network	Number of training images	Number of testing images	Error reached	Training time	Epochs required	Classification rates
SAE	4000	2422	0.0048	120 secs	400	93%

 Table 5.7: Identification rate of the SAE during testing

Figure 5.8 shows some misclassified images when model is tested. Note that these images were classified as humans they don't belong to.



Figure 5.9: A sample of some misclassified images

#### **5.5 Results Discussion**

A deep learning approach for the identification of palmprints using stacked auto-encoder is presented. Image processing methods are used to enhance and extract the useful features of a palm. Those features are then fed into a stacked auto-encoder which learns to classify these features into different subjects. This work is based on a stacked auto-encoder that is trained to classify the palmprint images to learn the useful features that can distinguish humans. This type of networks is trained using an algorithm called Greedy-layer wise training which is meant to train the network layer by layer in an unsupervised manner. This allows the network to gain the power of extracting of important features that will be used in the next training phase which is called Fine-tuning. Fine-tuning is to train pre-trained network to palmprint using the

conventional backpropagation learning algorithm by using the weights obtained for the pretraining phase.

The network was trained and tested on a large number of images collected from two different databases. The network performed well in training where it achieved an accuracy of 100% during fine-tuning; however this required a long training time of 140 seconds and 400 iterations. Note that the network reached a small minimum square error of 0.00489 during training

Moreover, the network performed well in testing where it achieved a relatively good classification rate of 93% which however can be improved.

#### **5.6 Results Comparison**

Palmprint has always been a hot topic for researchers to work on. Identification human through their palmprint is a tough task due to the various level features found in a palmprint. Thus, researchers have always tried to use intelligent networks that can classify the palmprints if the features are well processed and extracted. However, different approaches were conducted by researchers in order to tackle this problem. Table 5.8 shows a comparison of two different methods that were proposed to identify palmprints. The first one is a method that uses Contourlet transform (Butt et al., 2008) proposed to identify palmprints. Second one is combination of principal components analysis and a backpropagation neural network (Insanto et al., 2016) proposed for the same task. As seen in table 5.8, the proposed method in this thesis which uses image processing techniques to extract palm features and stacked auto-encoder as a classifier has outperformed other two methods in terms of accuracy.

Furthermore, for more comparison, a stacked auto-encoder of three hidden layers is built for the same task. Similarly, the three hidden layers stacked auto-encoder was created by training each layer separately and then stacking all layers to together to fine-tune it. The network was trained on the same images that were used for training the two hidden layers auto-encoder. Experimentally, it was seen that adding one more hidden layer is not efficient and doesn't contribute to improving the accuracy of the network. The network achieved less training accuracy in training (94%) and testing (91.2%) after adding one more hidden layer. Moreover,

the network required more training time of 200 seconds which is longer than the stacked autoencoder with two hidden layers.

As a result, it can be stated that adding more hidden layers doesn't end up with better accuracy during training and testing. Also, it required longer time and iterations. The reasons of that may be obviously the increase of depth of the network which requires more training parameters, which means longer training time. Moreover, the reason why the network couldn't achieve better accuracy after adding one more hidden layer may be the small number of data. The deeper the network, the more data it needs to be trained. Therefore, in order for this three hidden layers network to achieve better accuracy tan the previous one, data should be augmented.

Related researches	Methods used	Database used		Total number of users	Total number of samples	Accuracy
Palmprint identification using contourlet transform (Butt et al., 2008)	Contourlet transform	Pol Train 3881	yU Test 3881	460	7762	88.91%
Intramodal feature fusion using wavelet for palmprint authentication (Krishneswari, 2011)	Wavelet transform	Pol Train 5000	lyU Test 5000	200	1000	93.1%
Proposed Method	Stacked Auto- encoder (2 hidden layers)	Pol Train 1840	yU Test 920	235	2600	95%
Proposed Method	Stacked Auto- encoder (3 hidden layers)	Pol Train 1840	yU Test 920	235	2600	91.4%
'Personal Authentication Using Multifeatures Multispectral Palm Print Traits' (Rajagopal & Manoharan, 2015)	Ordinal code	CA Train 2000	SIA Test 600	312	2600	91%

Personal	Comp	CA	SIA			
Authentication Using Multifeatures Multispectral Palm Print Traits' (Rajagopal	Code	Train 2000	Test 600	312	2600	84.67%
& Manoharan, 2015)						
Proposed Method	Stacked	CASIA	4			
	Auto-	Train	Test			
	encoder(2 hidden	4000	1502	312	5502	93%
	layers)					
Proposed Method	Stacked	CASIA	A			
-	Auto-	Train	Test			
	encoder(3 hidden	4000	1502	312	5502	91.2%
	layers)					

Table 5.8: Results comparison (Continued)

### CHAPTER 6 CONCLUSION

In this thesis a deep network based stacked auto-encoder for the palmprints identification was carried out. The employment of deep network in particular stacked auto-encoder in such a field is considered as a challenging task as human's identification via their palms is security verification problem, in which even small margin of errors is not allowed. Thus, such a deep identification model is needed as it makes the human verification work faster and easier.

A stacked auto-encoder is employed in this work and it was trained and tested on 4000 and 2422 palmprint images, respectively. The network is first trained on images taken directly from the database, with processing and enhancing. The goal of processing was to enhance and extract the useful and unique features that can distinguish each human. This extraction of features can consequently help the deep model in classifying the images more accurately. As a result, this can lead to a better network learning and convergence. The images were enhanced using median filtering and adjustment of intensities, which results in a brighter image where features are more spotted on.

Experimentally, this network was tested on 2422 unseen palmprints and an evaluation of its performance was carried out. The performance of network was discussed and a comparison of the two network performance was shown, in terms of accuracy, error reached, training time, and number of iterations needed. The network was capable of achieving a good generalizing power where it reached an accuracy of 93%. After comparing the proposed model accuracy with other related works, it was seen that the proposed stacked auto-encoder model outperformed some works which use different classifier to identify palmprints.

In conclusion, the testing of stacked auto-encoder showed that it gained a good capability extracting the rightful features and an accurate of identification of new unseen palmprints images. Thus, it can be stated that the SAE can be a good classifier for the pamprints identification with a simple pre-processing stage before classification, with a small margin of errors. Moreover, it is seen that the processing of palmprints images using filtering and edge

detection has can help in in improving the learning of the stacked auto-encoder, which results in a better accuracy during the testing of the network.

#### REFERENCES

- Bovik, A. C., Huang, T. S., & Munson, D. C. (1987). The effect of median filtering on edge estimation and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2), 181-194.
- Butt, M.A.A., Masood, H., Mumtaz, M. (2008). Palmprint identification using contourlet transform. In *Proceedings of the International Conference on Biometrics: Theory, Applications and Systems* (pp. 1–5). Los Angeles: California.
- Cappelli, R., Ferrara, M., & Maio, D. (2012). A fast and accurate palmprint recognition system based on minutiae. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* (*Cybernetics*), 42(3), 956-962.
- Dai, J., & Zhou, J. (2011). Multifeature-based high-resolution palmprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 945-957.
- Dai, J., Feng, J., & Zhou, J. (2012). Robust and efficient ridge-based palmprint matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(8), 1618-1632.
- Doi, J. and Yamanaka, M. (2003). Personal authentication using feature points on finger and palmar creases. In *Proceedings of 32nd Applied Imagery Patten Recognition Workshop* (pp. 282-287). Berlin: Germany.
- Du, K. L., & Swamy, M. N. (2013). Neural networks and statistical learning. Springer Science & Business Media.
- Fausett, L. (1994). Fundamentals of neural networks: architectures, algorithms, and applications. Prentice-Hall, Inc.
- Gonzalez and Woods, Digital Image Processing, Second Edition, Prentice-Hall, pp. 108-120, 2002.
- Gurney, K. (1997). An introduction to neural networks. CRC press.

- Han, C. C. (2004). A hand-based personal authentication using a coarse-to-fine strategy. *Image* and Vision Computing, 22(11), 909-918.
- Haykin, S. S. (2009). *Neural networks and learning machines* (Vol. 3). Upper Saddle River, NJ, USA: Pearson.
- Hennings, P., & Kumar., B.V.K.V. (2004). Palmprint recognition using correlation filter classifiers", In *Proceedings of the 38th Asilomar Conference on Signal, Systems and Computers* (pp. 567-571). Pacific Grove: USA.
- Isnanto, R. R., Zahra, A. A., Haq, A. K., & Rozy, F. (2016). Palmprint Recognition Systems based-on Backpropagation Neural Network and Euclidean Distance using Principal Components Analysis (PCA) Feature Extraction. *International Journal of Software Engineering and Its Applications*, 10(11), 131-142.
- Jain, A. K., & Feng, J. (2009). Latent palmprint matching. *IEEE Transactions on pattern* analysis and machine intelligence, 31(6), 1032-1047.
- Jain, A. K., Nandakumar, K., & Ross, A. (2016). 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 79, 80-105.
- Krishneswari, K., & Arumugam, S. (2011). Intramodal feature fusion using wavelet for palmprint authentication. *International Journal of Engineering and Scientific Technology*, 3(2), 1597-1605.
- Kumar, A., & Wang, K. (2016, December). Identifying humans by matching their left palmprint with right palmprint images using convolutional neural network. In *Proceedings of the First International Workshop on Deep Learning and Pattern Recognition*. Mumbai: India.
- Lto, K., Aoki, T., Nakajima, H., Kobayashi K., & Higuchi, T. (2006). A phase-based palmprint recognition algorithm and its experimental evaluation. In *Proceeding of International Symposium on Intelligent Signal Processing and Communications* (pp. 2669-2672). Budapest: Hungary.

- Ortiz, F., & Torres, F. (2004). Vectorial morphological reconstruction for brightness elimination in colour images. *Real-Time Imaging*, *10*(6), 379-387..
- Priya, P., & Dsouza, D. A. (2015). Study of Feature Extraction Techniques for the Detection of Diseases of Agricultural Products. *International Journal of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering*, 3(7), 3-10.
- Pujari, J. D., Yakkundimath, R., & Byadgi, A. S. (2013). Reduced color and texture features based identification and classification of affected and normal fruits' images. *International Journal of Agricultural and Food Science*, 3(3), 119-127.
- Rajagopal, G., & Manoharan, S. K. (2015). Personal authentication using multifeatures multispectral palm print traits. *The Scientific World Journal*, 2015.

- S Aoyama, S., Ito, K., & Aoki, T. (2013). Similarity measure using local phase features and its application to biometric recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 180-187). Oregon: USA.
- Sun, Q., Zhang, J., Yang, A., & Zhang, Q. (2017, June). Palmprint recognition with deep convolutional features. In *Proceedings of the Chinese Conference on Image and Graphics Technologies* (pp. 12-19). Springer: Singapore.
- Sun, Z., Tan, T., Wang, Y., & Li, S. Z. (2005). Ordinal palmprint representation for personal identification. 2005. In *Proceedings of the International Conference of Visual Patterns Recognition* (pp. 363–368). San Diego: America
- Tino, P., Benuskova, L., & Sperduti, A. (2015). Artificial Neural Network Models. In Springer Handbook of Computational Intelligence (pp. 455-471). Springer, Berlin, Heidelberg.
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image* processing, 13(4), 600-612.
- Wu, X., Wang, K., & Zhang, D. (2006, November). Palmprint texture analysis using derivative of Gaussian filters. In *Proceedings of the International Conference on Computational Intelligence and Security*, 2006 (pp. 751-754). Guangzhou: China.
- Zbontar, J., & LeCun, Y. (2016). Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, *17*(1-32), 2.
- Zhang, L., Guo, Z., Wang, Z. & Zhang. D. (2007). Palmprint verification using complex wavelet transform. In *Proceedings of International Conference on Image Processing* (pp. 417-420). Texas: USA.
- Zhou, Y., Liu, Y., Feng, Q., Yang, F., Huang, J., & Nie, Y. (2014). Palm-vein classification based on principal orientation features. *PLoS One*, 9(11), e112429.

APPENDICES

#### **APPENDIX 1**

#### **IMAGE PROCESSING CODE**

close all

clc

PATTERNS = [];

for k = 1:4000

 $myFolder = 'C: Users TOSHIBA \ imp. \ documents \ Sleiman \ libie \ New \ Data \ IITD \ Palmprint \ V1 \ Segmented \ AllImages';$ 

filePattern = fullfile(myFolder, '\*.bmp');

jpegFiles = dir(filePattern);

baseFileName = jpegFiles(k).name;

fullFileName = fullfile(myFolder, baseFileName);

fprintf(1, 'Now reading %s\n', fullFileName);

img = im2double(imread(fullFileName));

```
% f=rgb2gray(img);
```

```
I=imresize(img, [256 256]);
```

% figure, imshow(I), title('gray image')

%pause

m=medfilt2(I);

% figure, imshow(m), title('median filtered image'); %%% using median filtering

% pause

```
background = imopen(m,strel('disk',10));
```

% figure, imshow (background);

% title('background');

%pause

%% ADD the original form the background

I2 = m+background;

% figure, imshow(I2),title('ADDED image');

% pause

%% increase the image intensity

% I3=I2;

I3 = imadjust(background);

I3 = imadjust(I2);

% figure, imshow(I3), title('adjustedimage');

%pause

d = edge(I3,'canny',0.35); %% .....%# Perform Canny edge detection.. The value increases ...the edges points decreases

% figure, imshow(d)

%pause

im1=imresize(img, [64 64]);

v=im1(:);

PATTERNS = [PATTERNS v];

end

```
save('DATA.mat','PATTERNS');
```

## **APPENDIX 2**

#### NEURAL NETWORKS CODE

load DATA1.mat;

load DATA2.mat;

train\_input=[PATTERNS(:,1:50),PATTERNS2(:,1:50)];

a=ones(2,180);

b=a(1,1:93);

c=a(2,1:93);

d=a(1,94:180);

e=a(2,94:180);

t1=[b;c\*0];

t2=[e\*0;d];

t1=t1(:,1:50);

t2=t2(:,1:50);

target=[t1,t2]; %the desired output data

%Autoencoder training using Greedy wise layer training:

### %%%% NOTE THAT HERE THE NUMBER OF INPUTS = NUMBER OF OUTPUTS.....NO OUTPUT

#### %%%% LABELING...THIS IS CALLED UNSUPERVISED LEARNING

net = newff(minmax(train\_input),[100 2500],{'logsig','logsig'},'trainscg');

#### % TRAINING THE NETWORK

net.trainParam.lr = 0.47; % Learning Rate.

net.trainParam.show = 300; % Frequency of progress displays (in epochs).

net.trainParam.epochs =100;% Maximum number of epochs to train.

net.trainParam.mc = 0.8 % Momentum Factor.

TARGETS=train\_input; % OUTPUTS = INPUTS

[net,tr] = train(net,train\_input, TARGETS);

train1=sim(net, train\_input);

Hidden\_weights = net.LW $\{2,1\}$ ;

inp\_weights=net.IW{1,1};
THEM LATER ON

%%%%SAVE THE inp -Hidden WEIGHTS TO USE

%-----

save('AE1','net');

#### %SAVE THE NETWORK

% plotWeights(net);

% Fine tuning of the Autoencoder

%%%HERE WE TRAIN THE NETWORK NORMALLY LIKE A BACKPROPAGATION NEURAL NETWORK

# %%NOTE THAT THE INPUT-HIDDEN WEIGHTS OF THIS NETWORK ARE EQUAL THE ABOVE

%% INP-HIDDEN WEIGHTS .... THE LAYER WEIGHTS ARE INITIATED RANDOMLY

net = newff(minmax(train\_input),[100 2],{'logsig','logsig'},'trainscg');

net.IW{1,1}=inp\_weights; %%%%%%input weights same as the aboove network weights(2500\*100)

net.LW{2,1}=0.01\*rand(2,100) randomly (100\*2) %%%%%% layer weights are initiated

% TRAINING THE NETWORK

net.trainParam.lr = 0.05; % Learning Rate.

net.trainParam.show = 300; % Frequency of progress displays (in epochs).

net.trainParam.epochs =400;% Maximum number of epochs to train.

net.trainParam.mc = 0.45 % Momentum Factor.

net.trainParam.goal = 0.001

[net,tr] = train(net,train\_input,target);

train = sim(net,train\_input);

#### %RECOGNITION RATE OF TRAIN DATA

%target max indices

[M,I\_t]=max(target);% row vector

% dimensions of target matrix

[u,v]=size(target);

%actual output matrix sim\_net

sim\_net=sim(net,train\_input);

[N,I\_sim\_net]=max(sim\_net);% row vector

% comparison of target and actual outputs

result = I\_t==I\_sim\_net;% row vector

% sum of all elements, 1s, to know how many corrects

corrects=sum(result);

%recognition rate,

w=double(corrects\*100/v); % let recognition rate be w

fprintf('train recognition rate is %d\n',w);