République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université 8Mai 1945 – Guelma
Faculté des sciences et de la Technologie
Département d'Electronique et Télécommunications

Mémoire de fin d'étude

pour l'obtention du diplôme de Master Académique

Domaine : **Sciences et Technologie**
Filière : **Electronique**
Spécialité : **Instrumentation**

Palmprint Recognition Using Deep Learning

Présenté par :

Himeur MOHAMMED

Sous la direction de :

Dr. Doghmane HAKIM

Juillet 2021

Remerciements

J’exprime ma profonde gratitude à **Dr. Hakim Doghmane**, enseignant au Département d’Électronique et Télécommunications et mon encadreur pour me guider dans cette recherche. Son expérience et son aide scientifique m’ont été essentielles. Sa disponibilité ainsi que l’attention qu’il a portée à ce travail ont été un atout précieux dans le suivi et l’avancement de mes travaux de thèse.

Je doit aussi remercier mon frère **Himeur Chems Eddine**, Ingénieur de recherche a l’Institut de Recherche en Informatique de Toulouse (***IRIT***) pour son assistance dans cette recherche, ses expertises dans le domaine d’intelligence artificielle et spécialement dans les réseaux neurones a était bénéficiaires pour mon travail et ma compréhension du domaine en générale.

Finalement, je remercie ma famille qui ma soutenu a travers mon cursus universitaire, et exprime ma gratitude a tous les enseignants qui m'ont aider durant mes années à l’Université 8 Mai 1945, Guelma, et spécialement au Département d’Électronique et Télécommunications pour leurs efforts, connaissances et patiences.

Abstract

Biometry is the use of biological features in order to identify individuals; biometric systems aim to make such identification automatic, fast and highly reliable; a biometric system uses one or multiple modalities to verify the identity of a user, a highly recommended biometric modality is the palmprint. Palmprint recognition systems are highly reliable because of the properties of palmprints from their distinctiveness to their stability over time, and also their accessibility since there are multiple ways to acquire a palmprint image. The recognition process goes through multiple steps from the image acquisition, pre-processing, feature extraction and finally matching (or classification); each step has a plethora of ways it can be done by. The recognition of the palmprint is best done using deep learning, using the convolutional neural network AlexNet to extract the features and then classifying the palmprint accordingly, the system overall returns good results and high accuracy.

Contents

List of figures

List of tables

Introduction

Identity authentication has been important since the old times, for centuries people used multiple methods to verify identities, such as signatures which has been used since the very first days of human history to verify transactions, and they are still used to this day along side other developed authentication methods such as fingerprints, palmprints, face recognition and other biometric methods. Using palmprints has developed over the last few decades and since, palmprint recognition systems has been in demand as their market share grew. There are multiple ways to build a palmprint recognition system, a good method is to employ deep learning and it’s advanced algorithms. In this study, we are going to give an overview of biometry and it’s different systems while highlighting palmprint recognition, we are also studying deep learning and then employing it to build a robust and highly reliable palmprint recognition system.

1. Introduction

Fraud, forging, impersonation, imposing and other vile acts that only serve to steal, trespass and endanger people have always been, and has never ceased to grow, either in old times or modern day societies, and especially in these times of technological advancement, when most of personal information, banking statements and transactions can be stored or done online, and many other fields now use electronics to operate. Security is paramount, and access to private places, files or assets has to be protected and restricted to the right people whom these things are meant for, and here biometry is introduced, as it aims to identify a person using their biological features, biometry will give a reliable solution against fake identification, and ensure restricted access and high security. In this chapter we will explain biometry in detail and display it's different modalities.

2. Biometry by definition

Biometrics is the science of identifying or verifying an individual based on their biological characteristics. it tends to be used in biometric systems that aim to make recognition automatic and reliable. The idea of identifying individuals by their physical characteristics actually dates back to the 19th century, when Alphonse Bertillon introduced the first steps of forensic science, which used biometric identification, his method, called Bertillonage, allowed the French police to identify criminals through several physiological measurements, however this method was manual and was performed only by experts.

Biological characteristics are distinguishing features that are unique to each person which makes biometrics reliable. Characteristics can be divided into two groups: physical and behavioral characteristics and are called biometric modalities. Physical characteristics are the most commonly used, they are physical features such as fingerprints, palmprints, iris, hand geometry, ...etc, while behavioral characteristics are based on the actions of the person, let's say the most usual ones such as gait, facial expression, ...etc.

Biometrics is the best and most efficient way to verify a person's identity because ID cards and badges can be forged, passwords can be guessed or generated, and keys can be lost or stolen, while biometric data is not reproducible, which guarantees security against theft, loss and forgery.

3. Biometric Modalities

A biometric modality is a distinctive characteristic of an individual, it is used to verify the identity of a person. Any biometric modality has the following properties:

- **Universality:** it can be found in every individual.
- **Distinctiveness:** it is unique for each individual, you will not find two individuals with the same biometric representation.
- **Stability :** for it to be reliable, it must be stable over time, it does not change over time, and also regardless of acquisition conditions.
- **Collectability:** it can be acquired (scanned, recorded...etc).
- **Performance:** it must ensure fast, robust and accurate recognition.
- **Acceptance:** it should be relatively easy to use.
- **Circumvention:** it must be hard to falsify, it gets more reliable as it gets harder to reproduce by others.

The following table shows some examples of biometric modalities:

Physical modalities	Behavioral modalities
Fingerprints	Keystrokes
Palmprints	Gait (way of walking)
Hand geometry	Signature
Face	Voice
Ear	
DNA	
Iris	

Table. 1.1: Some biometric modalities.

3.1. Physical Modalities

These are some biometric modalities that rely on physical (morphological) features to establish a distinctive identification:

- **Fingerprints :** These are probably the most used biometric modality. Bometric systems based on this modality are quite popular and can be found in a lot of places, they can be used for both identification and verification. The method used to verify fingerprints is by matching it’s minutiae details, it is quite the robust method, however it faces some difficulties when it comes to image acquisition since it requires contact scanning, humidity and hygiene can affect the performance of the system.

Figure. 1.1 : A sample of fingerprints.

- **Palmprints :** The use of palmprints is considered a popular method in biometric systems, Palmprints are rich in features and can be recognized either from their main lines and wrinkles which can be acquired even from low resolution images, or from it’s minutiae details, but those require high resolution images of the palmprint. Palmprint image acquisition can be done either using scanners that require contact, or without contact using digital cameras.

Figure. 1.2 : A palmprint image

- **Hand Geometry :** Based on the fact that every person has a distinct hand shape that doesn’t significantly change with time, the hand geometry identification was adapted, and it is quite the popular method. Hands are recognized either by checking the geometry of the whole hand, or by only checking two fingers, it bases on the length and shape of the fingers and knuckles**.**

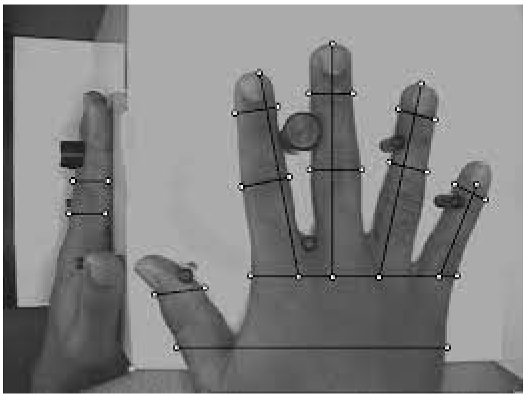


Figure. 1.3: Identification Using Hand Geometry.

- **Face :** This biometric technology uses images of a person's face to verify their identity, it is considered a robust method to recognize individuals, however, a face itself as a modality can change with age, or with makeup and plastic surgery, and even different facial expressions can have an effect on the recognition process. There are two types of facial recognition systems, ones that recognize a person in a known environment, mostly used in authentication, and other systems that recognize an individual from a group of people in a random environment.

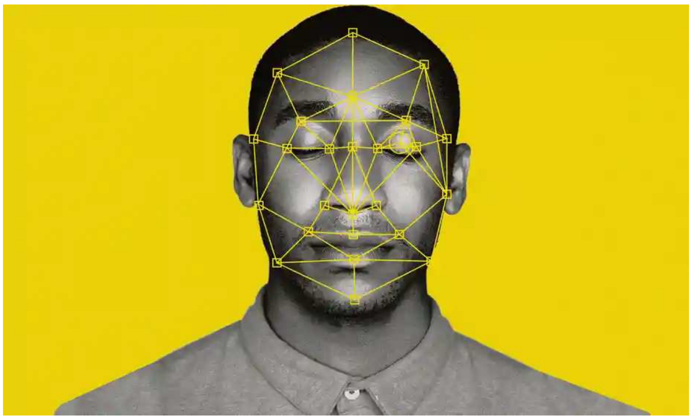


Figure. 1.4: Facial Recognition.

- **Ear :** It is a fairly new method of recognition, it was suggested that the shape of the ear is distinct from a person to another. An approach used in ear recognition is verifying the max-line, which is a line that has both its end points on the edges of the ear.

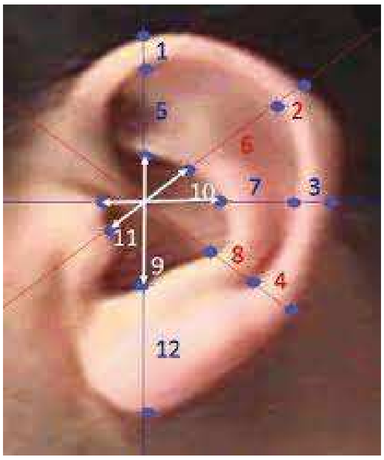


Figure. 1.5: Ear Recognition.

- **DNA :** it is short for Deoxyribo Nucleic Acid, it is unique for every individual and it can be found in every single cell of the body, which makes it a very reliable biometric modality when a positive identification is required, however it's not cost effective since it requires a lot of testing and resources, it's mostly used in legal and law enforcement fields.

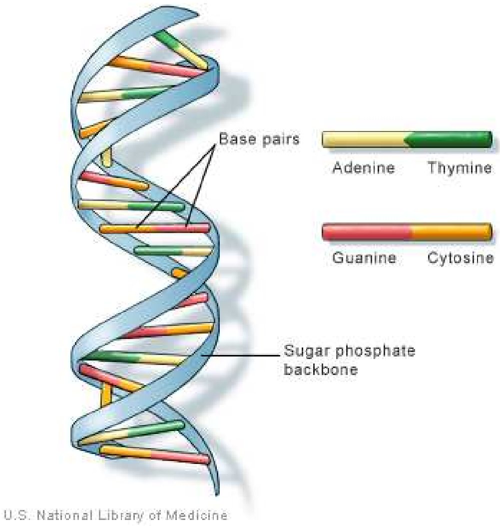


Figure. 1.6: Identification Using DNA.

- **Iris :** It is probably the most effective biometric technology, iris is rich in textures and it's patterns are distinct from person to another, not to mention that this technology is safe to use, and it gives results at high speeds with great accuracy as well.

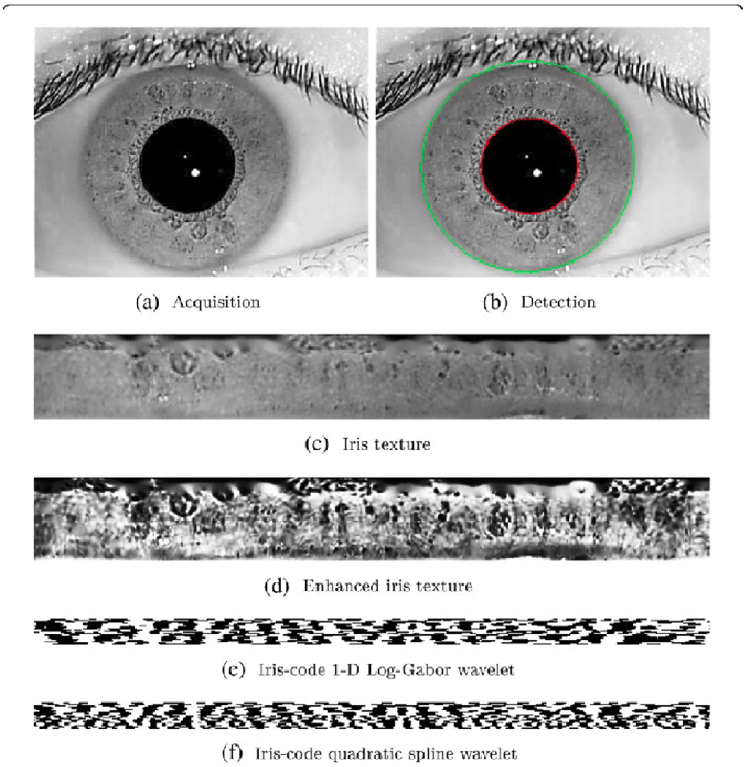


Figure. 1.7: Steps Of Iris Recognition.

3.2. Behavioral Modalities

These are the biometric modalities that rely on an individual's behavior to establish their identity :

- **Keystrokes :** It is one of the latest technologies in biometry, keystroke dynamics works by analyzing the way an individual types, taking as factors the time and speed of typing of the user, however it’s not too stable, as the mood of the person can affect it and also the fact that the way of typing changes as the person gets used more and more to typing.

Figure. 1.8: Keystrokes.

- **Gait :** It is the method of recognizing individuals through their way of walking, it is true that every person has their special way of moving, their body language, it can be a good way to identify individuals, however, it can be flawed since there are some conditions that affect the gait of a person, an injure that would cause the person to limp, or not walk comfortably can affect the system when it comes to recognition, still though it can be reliable in some cases.

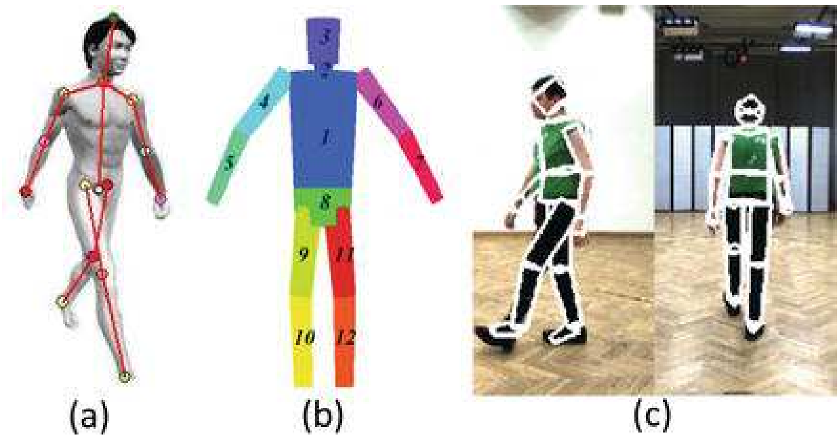


Figure. 1.9: Identification Using Gait.

- **Signature :** Since centuries, signatures have been used as a method of authentication, so it makes sense to adapt them into a behavioral biometric modality. Biometric systems based on signature recognition not only verify the shape of the signature, but they analyze the way of writing, how it fast it was, how the “t” is crossed and how the “i” is dotted, and that helps verify the identity of the signer as well.

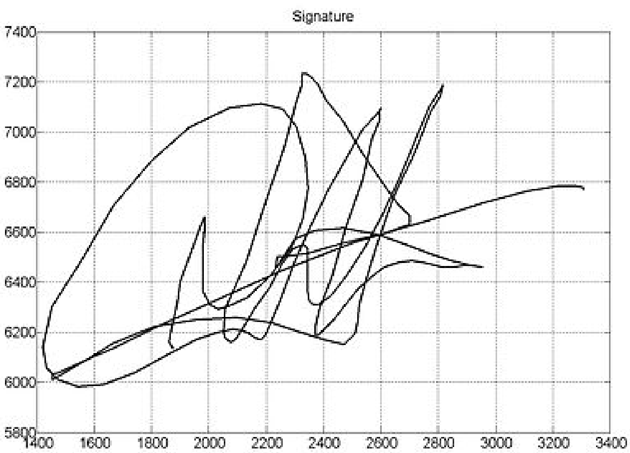


Figure. 1.10: Signature Recognition.

- **Voice :** it can be considered as one of the most convenient biometric modalities, the voice is distinct, and it is also nearly universal. A voice recognition system works by analyzing the waveforms and air pressure patterns when a person talks, the problem however with it, is that a voice can be mimicked, recorded, or in some very common cases the person can have a cold or a sore throat, and that would gravely affect the performance of the system.

Figure. 1.11: Vocal Recognition.

3.3. Comparison between biometric modalities

Naturally, not all biometric modalities would give the same degree of reliability, each one has it’s advantages and disadvantages, they are rated through their properties (which were mentioned previously, Universality, Distinctiveness, Stability, Collectability, Acceptance, Performance and Circumvention) , the following table that originates from Jain 2004 [7] evaluates each property with 3 levels of quality: high, medium and low using the following notations• • •, • • and • respectively.

Biometric modality	U	D	S	Co	A	Ci	P
Face	• • •	•	• •	• • •	• • •	•	•
Iris	• • •	• • •	• • •	• •	•	•	• • •
Fingerprint	• •	• • •	• • •	• •	• •	• •	• • •
Hand geometry	• •	• •	• •	• • •	• •	• •	• •
Palmprint	• •	• • •	• • •	• •	• •	• •	• • •
Keystroke	•	•	•	• •	• •	• •	•
Odor	• • •	• • •	• • •	•	• •	•	•
Retina	• • •	• • •	• •	•	•	•	• • •
Signature	•	•	•	• • •	• • •	• • •	•
Voice	• •	•	•	• •	• • •	• • •	•
Hand vein	• •	• •	• •	• •	• •	•	• •
DNA	• • •	• • •	• • •	•	•	•	• • •

Table. 1.2 : Comparison Between Biometric Modalities [2].

As it can be observed, behavioral modalities do not score too well compared the physical ones, we can see that their distinctiveness is low and so is their performance, whereas fingerprint and palmprint along with hand geometry score between high and medium in every property, making them the most reliable modalities.

4. Biometric Systems

A biometric system is used to recognize individuals based on their biometric data using artificial intelligence techniques. It can be modal or multimodal if it relies, respectively, on one or more biometric modalities.

Biometric systems have 3 main components, which are the input device, the biometric software and the database. The input device depends on the modality on which it is based, a scanner is the best word to describe it. it enrolls the inputs which are then used by the software, which is the main core of the system, as it processes the inputs and converts them into digital data, then compares its features with the database, the software is so important because the accuracy of the whole system mainly depends on its quality, as for the database, it is used to store data such as the features extracted from input samples which are then, used for comparison. There are three processing modes for biometric systems: enrollment mode, verification mode and identification mode.

4.1. Enrollment Mode

This is considered the learning phase for the system, it collects biometric information through scanners, processes it and creates feature vectors and saves them into the database, these vectors would later on be used as templates to compare with when the system is in verification or identification mode. Some systems require this mode, while some others don't use need it.

4.2 Verification Mode

In this mode the system is active, after acquiring the input image, it goes through processing and extracts it's features which would give it a feature vector, the system then compares that feature vector with the ones saved in the database, the system is doing a one-to-one comparison, so it will authenticate the identity if it finds a match, otherwise it will reject it.

4.3 Identification Mode

The system here is doing a one-to-N comparison, which means it will look to associate the identity with a person, it will compare the scanned features with the models saved in the database, if it finds a match, it will identify the person.

5. Performance and Evaluation of Biometric Systems

When a system is working in real time, its performance is affected by environmental variations such as humidity and temperature, noises, and even the quality of the input devices, as well as other changes that occur over time, collecting all these variations would make it almost impossible to get the same values in the feature vectors for the same person. Therefore, the matching algorithm is required to compare between the samples and compute their matching score, and then compare that score with the predefined acceptance threshold (the percentage of similarity to decide if it's a match). Following this method, would create the possibility of false acceptance along with false rejection, and so the performance of a biometric system is measured by two error values: False Rejection Ratio (FRR) and False Acceptance Ratio (FAR).

False rejection is the number of times the system rejects an authorized user and FRR is the ratio of the false rejection to the number of times the system is used for identification. In the other hand, False acceptance is the number of times the system accepts an unauthorized user and FAR is the ratio of the false acceptance to the number of times the system is used for identification [3]. The acceptance threshold is set to depending on the FRR and FAR, having a low acceptance threshold would mean the system would require high degree of similarity to call a match, which would lead to a low FRR, but it would also increase the FAR, the security would be relatively low, on the other hand, if the acceptance threshold is set to a high value, it will decrease the FAR ensuring high security, but it would also increase the FRR which would make it a bit annoying if it keeps rejecting authorized users possibly unreliable if it the false rejection rate is too high, and so the acceptance threshold is set according to the need, or what would be preferred, either low FRR or low FAR.

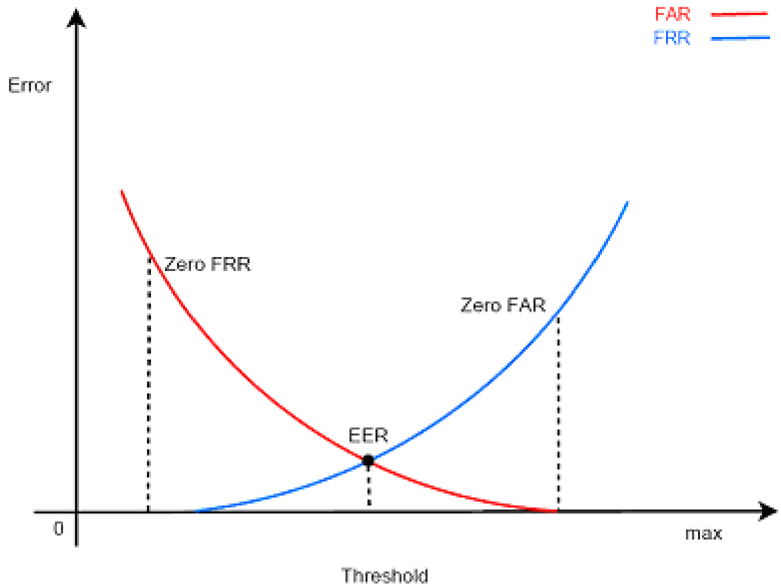


Figure. 1.12: Biometric System Performance Curve [3].

EER is the Equal Error Ratio, it's the point where false acceptance rate and false reject rate are equal, it can be considered the point of balance, but it's not necessarily the ideal, as stated before the threshold is set according to the desired security level, usually keeping FAR very low is the priority, as a few false rejections can be tolerated, but a false acceptance might have dire consequences.

6. Applications of Biometry

Biometric systems assure security, they are quite practical and can come in handy in a lot of situations, there are a lot fields where biometry can be applied, some of the applications of biometry:

- **Legal Applications :** Justice and Law Enforcement make use of all kinds of biometric systems, they rely a lot on fingerprints, palmprints, DNA, face recognition… etc, either to track down criminals, or to identify them. The advancement of biometry has always served law enforcement well, as it’s use has actually began there.
- **Governmental Applications :** It can be used in border control and airports, biometric systems would serve to automate the process of border crossing, it can also be used in healthcare, it would help secure and guarantee quick access to medical records to doctors, which would help a lot in the management of medical establishments.
- **Commercial Applications :** there are a lot of examples for commercial applications, as of present day, it has been introduced into most personal devices such as smartphones and computers, increasing their level of security and protecting the privacy of the users. It is also used in private security, as the biometric systems can be implemented nearly anywhere from a house or car door to a covered button. Biometry has also proven itself useful in the finances, as it can make transactions much more secure, it helps a lot in banking and account management, securing personal accounts and banking information, which would help a lot against fraud.

The market of biometric systems has grown so fast in the last 2 decades, and it keeps on growing as there has been an increase in the share of private sector market, there has been a need for biometric solutions especially for phone manufacturers. The global biometrics market reached a value of 28.2 billion USD in 2020, but it has been affected by COVID-19 as well, the fingerprints devices market has been expected to drop by 1.2 billion USD in 2020 [5].

In the other hand, the palm print is one of the best performing biometric modalities, we chose it as the subject of this research, given its reliability in terms of distinction, stability and performance compared to other modalities. Moreover, when using palmprints, you have a region of interest (ROI) which is the the core of the palm itself as shown in figures, the region has the principle lines and wrinkles of the hand, delta points, minutiae features and the geometry feature.

There are two types of approaches for palmprint recognition, high resolution approach (400 dpi or more) and low resolution approach(150 dpi or less), the high resolution allows for a very precise recognition process, it can be mainly used by forensics to identify criminals and so on, whilst low resolution is not so precise compared to the first one, but it is reliable enough for commercial and civil use such as access control.

A scanner is required to acquire a palmprint, an example of scanners is the CCD-Based Palmprint scanner developed by the Hong Kong Polytechnic University, it gives high resolution images, and it is also accurate since it has pegs that fixates the hand’s position on the scanner, the image captured by the CCD (Charge-Coupled Device) camera is converted into a digital image and then it gets processed.

Figure. 1.13: A CCD-Based Palmprint Scanner [3].

Figure. 1.14: Contactless Palmprint Scanner.

6.1. Palmprint Image Acquisition

As already stated, a scanner is required to acquire a palmprint image, CCD-Based Palmprint scanners aren't the only way, we can also use digital cameras, video cameras and digital scanners to obtain the image. Unlike the CCD-Based scanner, digital and video cameras allow for contactless palmprint acquisition, which would serve in a different approach, contactless scanning would help minimize the hygiene problem, especially in the times of pandemics, such as the present one COVID-19, or in places where high levels of hygiene are required such as hospitals. Digital scanners are highly effective in collecting palmprint images and they assure high precision, and their cost is relatively low, however their downside is that they cannot be used for real-time verification because of its long scanning times, Figure. 1.15 Shows 2 palmprint images, one collected with a CCD-Based (a) scanner and the other with a digital scanner (b).

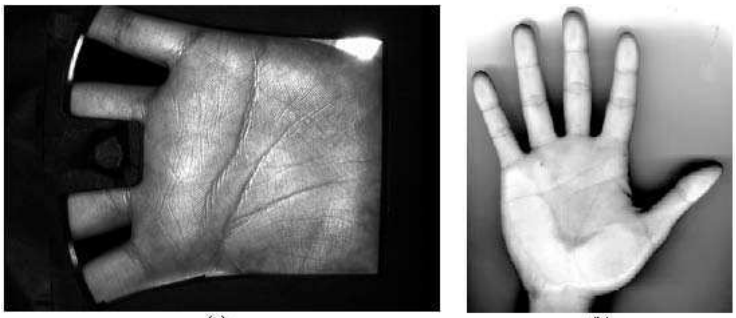
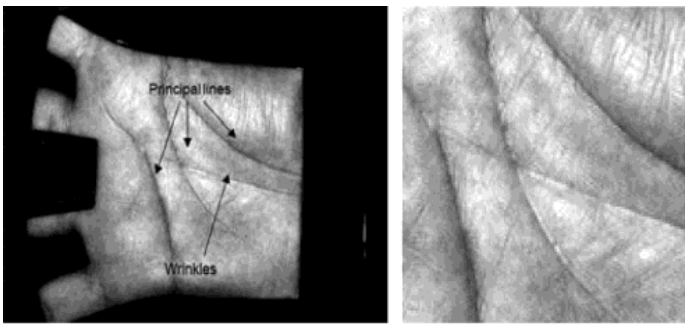

(a) (b)

Figure. 1.15: Palmprint Images Of Different Types [3].

When It comes to the data type, 2D palmprint images are the most used as they are the easiest to access and process. There are other types of images as well such as 3D, multi-spectral and minutiae images, the figures (1.16 to 1.18) Demonstrate these data types.



(a) (b)

Figure. 1.16: 2D Palmprint Image [4].

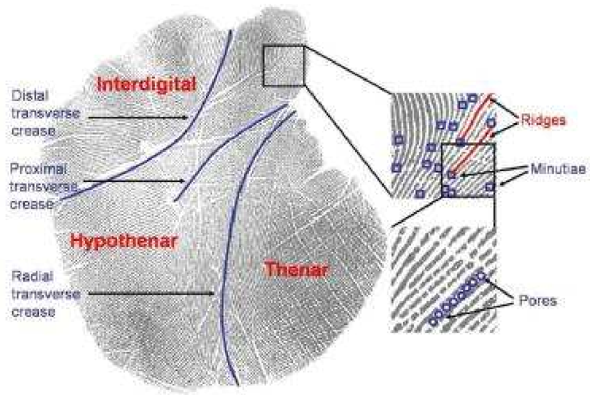


Figure. 1.17: Minutiae Details In A High Resolution Image [4].

Figure. 1.18: 3D Palmprint Images [4].

There are many effective acquisition algorithms, each one is built to suit it's main application making it highly reliable in the ideal scanning conditions, however, the real world environment is changeable, and most of the time the ideal conditions would not be met, which makes some algorithms unreliable for palmprint recognition. The solution to this problem is setting up different databases that simulate real world conditions, then evaluate the results and see if the algorithm is suitable for use, or if it needs some modification.

6.2. Pre-processing Step

Preprocessing is the most important step in palmprint recognition, as it influences the subsequent steps (feature extraction and matching). The quality of the pre-processing has an important impact on the results of the recognition process. It consists of five steps:

- Binarization of the image.
- Extraction of the palm contour.
- Detection of key points.
- Establishment of a coordination system.
- Extraction of the central parts.

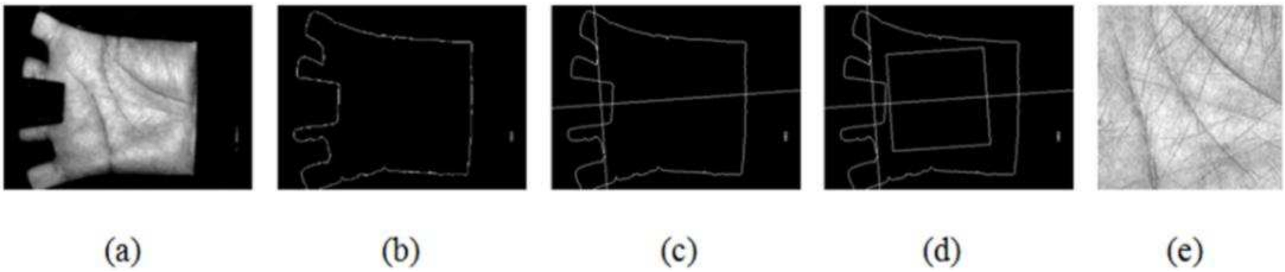


Figure. 1.19: Steps Of Image Pre-processing [4].

The main objective of preprocessing is therefore to extract the region of interest (ROI).

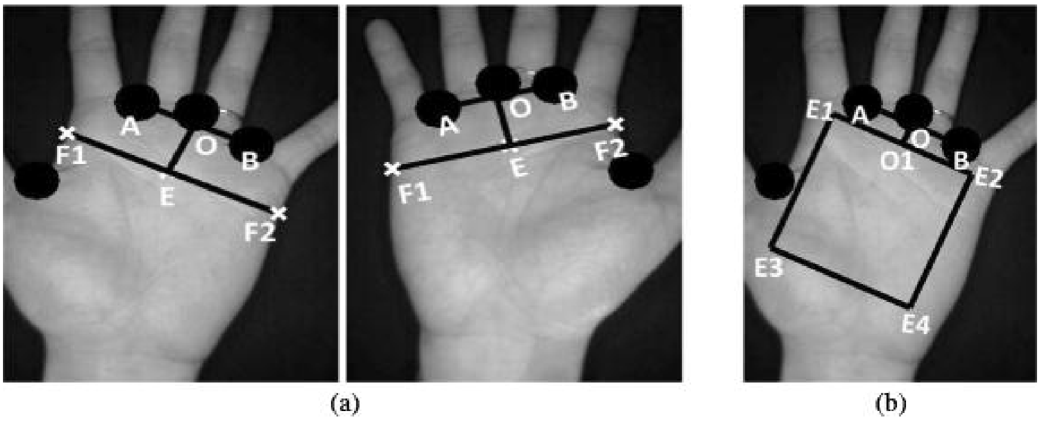


Figure. 1.20: ROI Location Using The Ratio Method [4].

6.3. Feature Extraction Step

After the acquisition and pre-processing the palmprint image, the next step is extracting the features from the ROI, features are of course different between individuals, good extraction and processing will assure high performance when it comes to matching. Over the recent years, feature extraction algorithms have developed quite rapidly, which gives a variety of accurate and effective ways to record features. The main features in palmprint images are: texture, orientation, discrimination and frequency, there are also some novel features that emerged over the years such as Laplacienpalm feature [8] , Local relative variance (LRV) [9], Gabor magnitude and phase (GMP) information [10], and many others such as Convolutional Neural Networks (CNN) [11] that help improve the accuracy of the extraction.

6.4. Palmprint Matching (or Classification) Step

The final step of palmprint recognition is matching, the goal here is to figure out if the palmprint at hand exists in the database and who it belongs to, that is based on the features extracted previously, the algorithm compares the features of the palmprint and tries to find the best match for it, of course with a defined percentage of incertitude.

The reliability of the system depends on the precision of the matching, it should have a very low, nearly null chance of it making a false identification, and it also should be able to easily identify correct ones without having to repeat the scanning multiple times.

7. Conclusion

In this chapter, we have given a detailed definition about biometry and its advantages in ensuring the security for both individuals and groups, we also explained biometric modalities and how they are chosen, and which modalities give the best results and are most reliable, we also gave generalities on biometric systems, how to evaluate their performance, and their applications, and finally we explained the properties of palmprints in detail, and why we chose the palmprint as the biometric modality to use in this research.

References

[1] S. Guennouni, A. Mansouri et al., "Biometric Systems and Their Applications", 10.5772/intechopen.84845. (2019) 4-11.

[2] N. Charfi et al., "Biometric recognition based on hand shape and palmprint modalities", Ecole nationale supérieure Mines-Télécom Atlantique, (2017)12-17.

[3] S. K. Panigrahy, "A Secure Template Generation Scheme for Palmprint Recognition Systems", Department of Computer Science and Engineering National Institute of Technology Rourkela Rourkela, Orissa, 769 008, India, (2008)3-12.

[4] D. Zhong, X.Du et al., "Decade progress of palmprint recognition: a brief survey", Neurocomputing (2018), doi: https://doi.org/10.1016/j.neucom.2018.03.081, 4-12.

[5] Research on the biometrics market, ABI Research, 06 May 2020. Latest visit July 2021.https://www.abiresearch.com/press/covid-19-pandemic-pummels-biometrics-market-causing-device-revenues-drop-us2-billion-while-forcing-investment-surge-ai-face-recognition-applications/

[6] G.S. Badrinath, P. Gupta et al., "Palmprint based recognition system using phase difference information", Future Generation Computer Systems, 28 (2012) 287-305.

[7] A.K. Jain, A. Ross et al., "An introduction to biometric recognition", IEEE Transactions on Circuits and Systems for Video Technology, 14 (2004) 4-20.

[8] H.G. Wang, W.Y. Yau, A. Suwandy, E. Sung, Person recognition by fusing palmprint and palm vein images based on "Laplacianpalm" representation, Pattern Recognition, 41 (2008) 1514-1527.

[9] X. Pan, Q.Q. Ruan, Palmprint recognition using Gabor-based local invariant features, Neurocomputing, 72 (2009) 2040-2045.

[10] M.R. Mu, Q.Q. Ruan, Region covariance matrices as feature descriptors for palmprint recognition using Gabor features, Int. J. Pattern Recognition Artificial Intelligence, 25 (2011) 513-528.

[11] Y. LeCun, L. Bottou et al., “Gradient-based learning applied to document recognition”, PROC of IEEE, (1998) 1-42.

1. Introduction

In the process of building a biometric system, we need to make a suitable recognition algorithm, the algorithm would go through a number of steps after acquiring an input, in the case of palmprint recognition, it goes through pre-processing, feature extraction and finally matching.

The classic programming methods fall short when it comes to biometric recognition and object detection, they are mostly straightforward and need exact values, which makes them unreliable in these cases since they work with estimations, so trying to build a recognition algorithm using these methods would only end up in a long and unoptimized program.

Machine learning introduces a solution, instead of using long and hard to maintain algorithm, we use it's techniques to build a program that can recognize palmprints and distinguish them. This chapter explains machine learning in general, and focuses on the algorithms to be used in this research.

2. Machine Learning

A frequent question is, what is machine learning? Or what's the difference between machine learning and deep learning? In general we could label machine learning as a kind of artificial intelligence, and in the same way deep learning is included in machine learning.

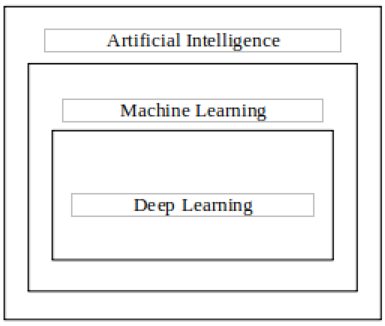


Figure. 2.1: Deep learning as a kind of machine learning.

As for the first question, machine learning is a modeling technique that involves data, or in other words programming machines to learn from data, in order to build models representing the classes that the computer delivers after learning from the data. The latter can be documents, images, sounds, videos, etc.

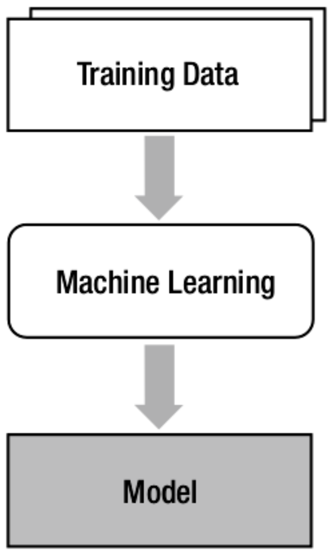


Figure. 2.2: The Process Of Machine Learning[1].

An example about models, is the e-mail spam filter, writing an auto-filter program without would be hard, and the end product wouldn't be reliable as it will require regular updates because the spammers would change their patterns when they that their e-mails are detected. Applying machine learning techniques would make it much easier, the program would detect patterns and keep update the filter on flight. So the process here is like this: the program gets a lot of e-mails as training data, it goes through the learning process as it detects patterns, and in the end it comes out with the best spam filter which is the model in this case.

Of course machine learning isn't just employed for problems like spam mail, it can be used to solve much more complicated problems, it is highly reliable in image and audio recognition, object detection and even handwriting recognition, for example if you see letters or numbers handwritten you would easily recognize them, but how would a machine do it? Traditional modeling techniques wouldn't do and there are not analytic models for this, what if we make the computer learn the same way a human being is has learned to distinguish numbers and letters? that would work if we come out with the right learning rule, and that is the concept of machine learning, it works to achieve a model out of training data when we can't really use equations and laws.

3. Deep Learning

Machine learning is about computers having the ability to perform tasks without being explicitly programmed, but the computers would still think and act like machines, their ability to perform some complex tasks such as gather data from pictures or videos or audio, still falls far from what humans are capable of doing.

Deep learning models introduce an especially sophisticated approach to machine learning, and are set tackle these challenges because they have been modeled after the human brain, a type of deep learning models is a neural network, for example: deep neural networks are built to permit data to be passed between nodes in highly connected ways, similar to how brain neurons work, this results in a non-linear transformation of the data.

While it takes tremendous volumes of information to 'feed and build' such a system, it can begin to come up with immediate results, and there's relatively no need for human intervention once the system is up and running.

4. Types of Machine Learning

Machine learning has multiple types and techniques, each one is used to solve a specific problem, the collection of these techniques can be divided into 3 main types depending on the degree of human interference in the learning process, the types are: supervised, unsupervised and reinforcement learning. We will explain each type while highlighting its most important algorithms.

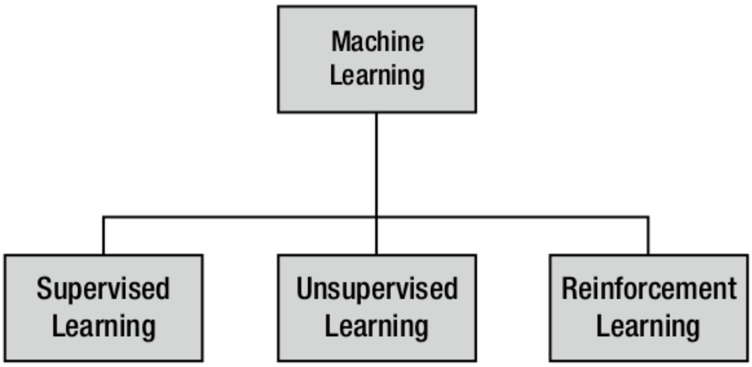


Figure. 2.3: Different Types of Machine Learning [1].

4.1. Supervised Learning

As the name suggests, human supervision is a part of this training technique. Basically, the data sets are fed into the system in labeled as (Input; Correct Output), meaning that in the learning process, the system works on the given data and gives an output, then compares it with the correct output, calculates the error and works on rectifying it, the process goes on until it reaches optimal results.

Again it is named supervised because it already knows the correct output, it learns how to get to it, and then it employs the results in other situations, a few examples on where we could use supervised learning are object recognition, speech recognition (Classification) and predicting numeric values like prices (Regression).

Supervised learning has multiple algorithms, such as K-nearest neighbors, linear regression, neural networks and logistic regression. We chose to focus on 3 important classification algorithms, K-NN, SVM and neural networks.

4.1.1. K-Nearest Neighbor

The principle of this algorithm is pretty straightforward, basically, the example in hand is classified depending on the class of it's nearest neighbors, it is usually useful to take multiple neighbors into account. Since the classification process requires the examples to be loaded in the memory, it is referred to sometimes as Memory-Based Classification, the following figure depicts the main principle of this algorithm.

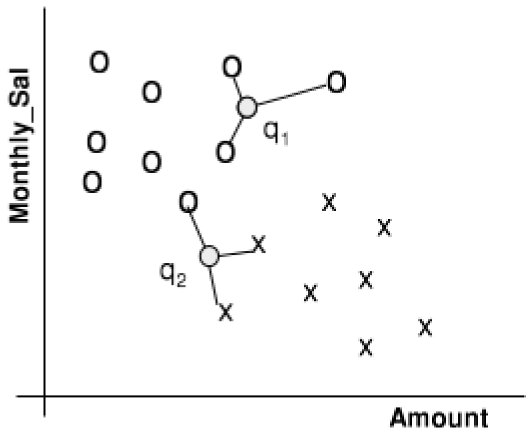


Figure. 2.4: K-NN Classification [6].

4.1.2. Support Vector Machines

Support Vector Machines are a set of methods used for regression, classification and outliers detection. SVM does multiple complex data transformations in order to find and draw the optimal line to separate and classify the data. It was developed in the framework of statistical learning theory.

4.1.3 Neural Networks

Neural Networks are one of the most used algorithms in Machine Learning as a whole, and especially in Deep Learning. A neural network is an architecture of input and output nodes that in its formation mimics the association of the neurons in a human brain, with such complex architecture it can be used to solve a lot of linear and main non-linear problems, this makes it very effective and a very important technique in image processing and object recognition.

4.2. Unsupervised Learning

In this type, the system works on unlabeled data and it tries to detect patterns and common features to classify the data without any human interference, it is quite useful when working on statistics. It has multiple algorithms such as Clustering and Association Rule Learning, we are not using unsupervised learning, so we won't be focusing on it's algorithms.

4.3 Reinforcement Learning

In reinforcement learning, the system here is called an Agent, it learns by observing the environment and then performing actions depending on the situation, after that it gets either rewarded or penalized depending on outcome of the action taken, if for example it takes an action and gets a penalty, it would update it's action policy to avoid it in the future. This type of learning is used mostly in robotics, it is employed in robots that learn how to walk.

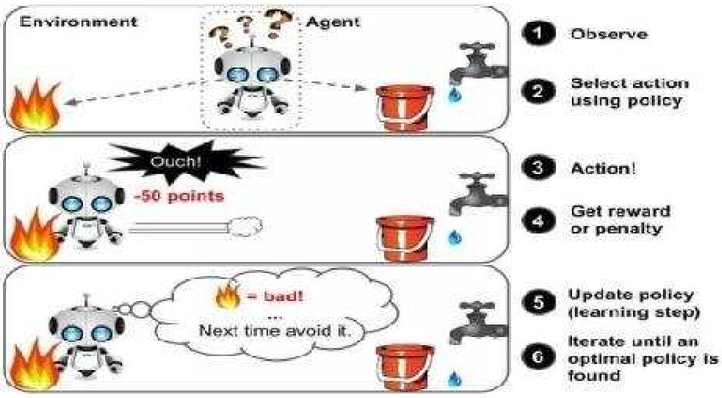


Figure. 2.5: Overview Of Reinforcement Learning [2].

5. Challenges Faced When Using Machine Learning

Like any other programming technique, machine learning has obstacles it has to overcome to get a working system that gives correct results, in this case the challenges concern the data, it's quality, quantity and how it is handled. We will address the issues in this part and suggest solutions to them.

5.1. Underfitting

This issue occurs when the model is too simple to learn, it doesn't have enough parameters to label the data correctly, when it comes to real life conditions, which are much more complex than the model in this case, it will give different results, and sometimes it might do that with the training data itself since it didn't learn correctly.

To fix this, we need to select a better model that has more parameters, also feed the best and most important features into the algorithms and reduce constraints on the model.

5.2. Overfitting

It is the opposite of underfitting, it happens when the model is too complex, it has too many parameters that overgeneralizes features, it may perform well with training data, but it in real conditions it will give incorrect results, since it is strictly made to give a very precise output for every data point, it will make an incorrect rule. The figure below explains overfitting, it makes a very complex curve to separate the different data types, it will give correct outputs with the training data since it's tailored to fit it, but it won't work in real-time conditions.

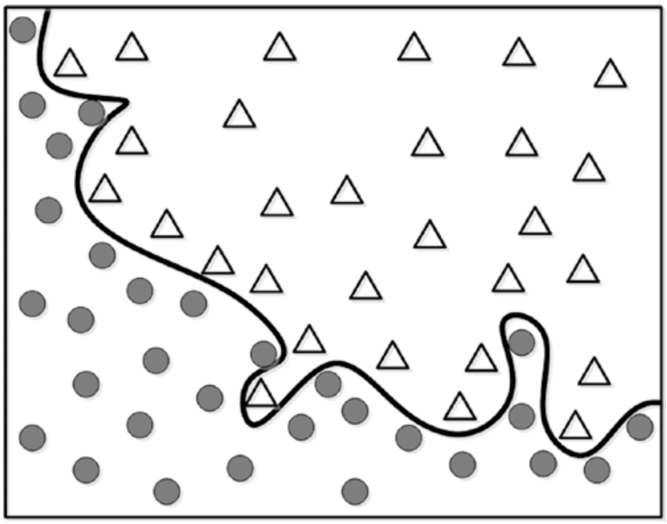


Figure. 2.6: An Example Of Overfitting [1].

To solve this problem, we can select a model with fewer parameters, gather more training data, and reduce the noise level in the data set.

5.3. Bad Quality And Quantity Of Data

Machine learning systems are not like humans, they can't just learn how to distinguish objects with very few examples, they require large amounts of data to work effectively, the data also has to be of good quality, it shouldn't have a lot of noises, errors and outliers as they make it harder for the system to detect patterns. Systems that work on speech recognition and image processing are highly sophisticated and so they would require a lot of good quality data sets to work properly, and so do other systems, and so we are required to provide a lot of training data, and also to spend a lot of time cleaning it up from any errors or noises.

6. Neural Networks

As stated before, neural networks are a widely used model in machine learning, they have a long history of development, especially lately as interest in Deep Learning has grown a lot, neural networks has grown important as well.

A neural network attempts to mimic the human brain, a computer uses memory to store data, the memory is divided into sections, each one has it's unique address, a brain however doesn't work like that, it instead alters the association of neurons, a neuron has no storage capability, it only transmits signals to other neurons, a human brain has approximately 86 billion neurons [8], that is a massive network, and the way these neurons are associated forms a specific information. Neural networks are constructed of nodes which imitate the brain's neurons, it also uses weight values that imitate the association of neurons, the figure below shows an example of a node.

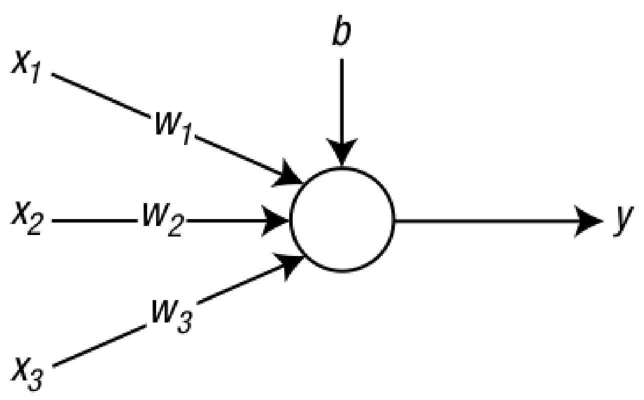


Figure. 2.7: A Neural Network With 3 Inputs and 1 Output Node[1].

The arrows denote the flow of the signals into and out of the node, x_1, $x_2$and x_3 are the input values while w_1, w_2 and w_3 are their respective weights, b is the bias value and finally y is the output of the node. Bias is a factor associated with the storage of information, in neural networks information is stored in forms of weights and bias. The output value is calculated as follows: first, each input value (x_i) get multiplied by it's weight (w_i), then the results are summed up at the node with the bias value, this gives us the weighted sum value v, which is finally put into the activation function $\varphi(v)$ which yields the output of the node. The following equations calculate the output for our 3 input node example.

$$v = (w_1 * x_1) + (w_2 * x_2) + (w_3 * x_3) + b \quad (2.1)$$

$$v = w*x + b \quad (2.2)$$

Where w and x are 1 by 3 and 3 by 1 matrices respectively

$$y = \varphi(v) = \varphi(w*x + b) \quad (2.3)$$

The activation function determines how the node behaves, there are multiple types of activation functions, one of the most used is the Sigmoid function illustrated in the following figure :

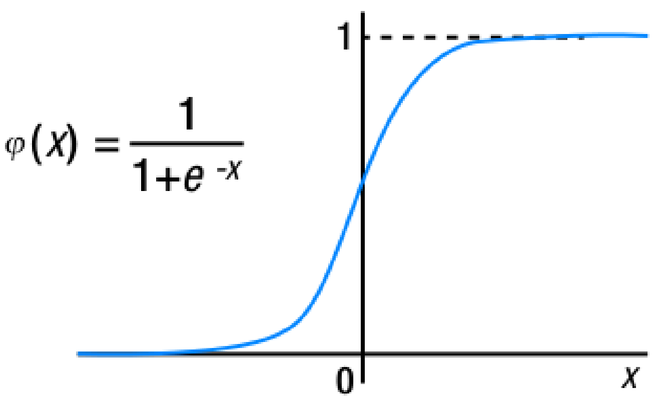


Figure. 2.8: The Sigmoid Function [1].

There are other types, we can also use a simple ramp for convenience in some examples. Just like the brain is a massive network of neurons, the neural network is network of many nodes, such networks are created depending how the nodes are connected, a commonly used type is the layered neural network, the figure coming next is an example :

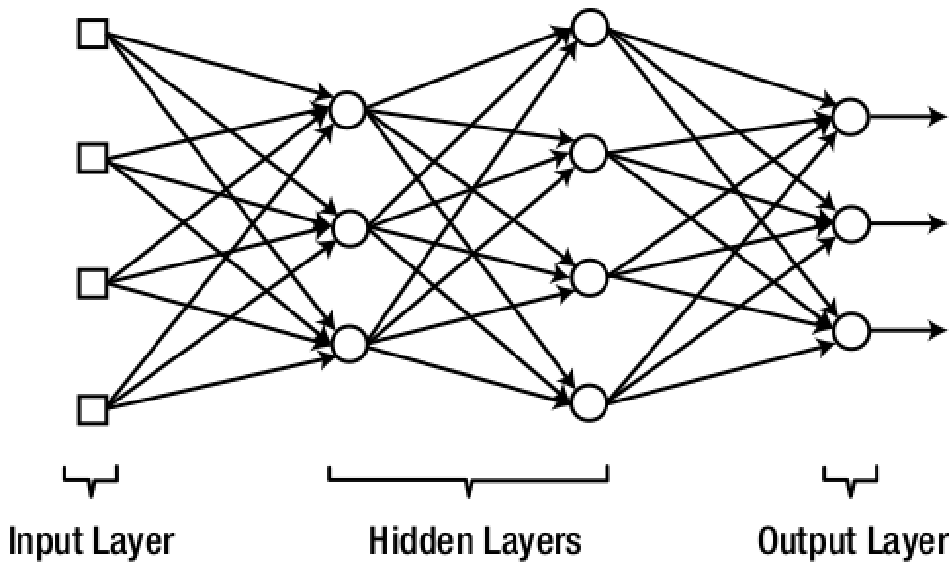


Figure. 2.9: A Multi-Layer Neural Network [1].

As it can be observed, this neural network has multiple layers of nodes, the nodes in the input layer transmit the input signal into the network, they do not have a weighted sum or an activation function, they simple send in the input, and so they are illustrated by squares, the nodes of the output layer give the final output of the network after it's been calculated throughout the network, the layers in between are called hidden layers since there is no way to access them as the neural network only gives the final output.

Neural networks have went through a long process of development to reach the state they are at today, it started with simple networks of only input and output layers, and have developed into deep networks with multiple hidden layers, we can distinguish 2 main types of neural networks: single layer neural networks which have only input and output layers, and then there are multi-layer neural networks, this type has 2 sub-types, shallow multi-layer neural networks, these have a single hidden layer of nodes, while the other sub-type is deep multi-layer neural networks which have multiple hidden layers, the example in the figure. 2.8 is a deep multi-layer neural network, the following figure shows the different types of layered networks:

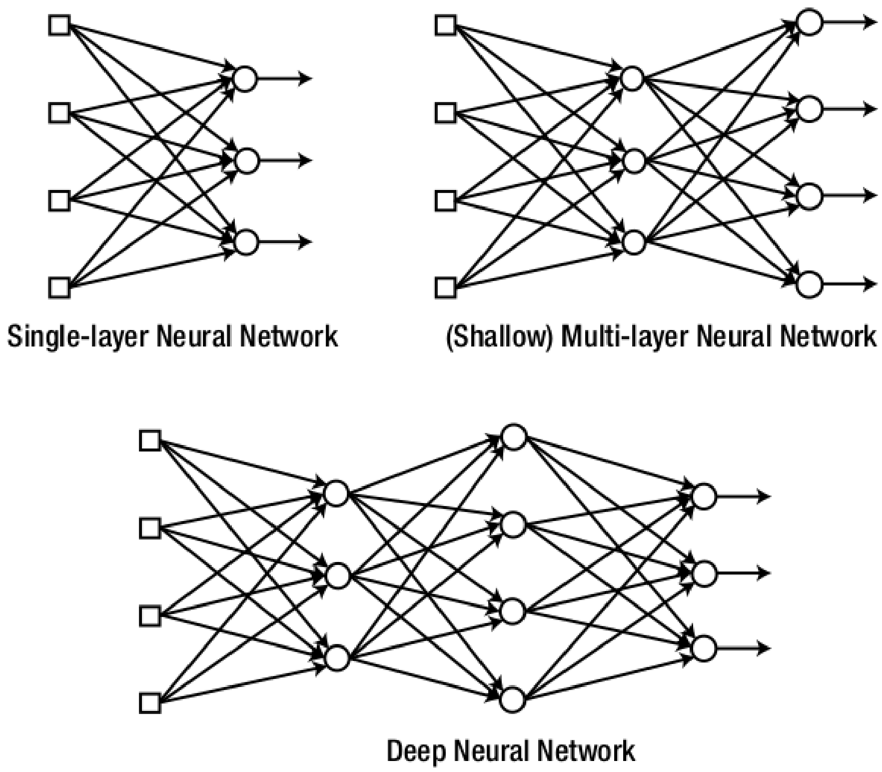


Figure. 2.10: Types Of Neural Networks [1].

In the next sub-sections, we will highlight the different types of layered neural networks, and we will also explain convolution neural networks.

6.1. Single Layer Neural Networks

As we previously explained, a single layer neural network consists of only input and output layers, and doesn't have any hidden layers. The only way to store information in a neural network is through learning, in the end, the information is stored in form of weights, any new information has to be learned by the network, the weights of course, should change accordingly, the learning rule for a neural network is the systematic approach to modifying the weights. In order to train a single layer neural network, we use the delta rule.

The delta rule is only capable of single layer training, but it is quite useful, and it helps grasp the core elements of learning rules for neural networks in general, how it works is pretty straight forward, as previously stated in supervised learning, the system receive the input in pairs (Input; Correct Output), what happens when using the delta rule is the network calculates it's output, then it calculates the error between the correct output and the calculated output, and then it adjusts the weights accordingly to minimize the error, hence why it is called the delta rule. The error and the updated weights are calculated by the following equations:

$$e_i = |y_i - d_i| \tag{2.4}$$

$$w_{ij} \leftarrow w_{ij} + \alpha * e_i * x_j \tag{2.5}$$

Where:

e_i is the error of the output node i.

y_i is the calculated output of the output node i.

d_i is the correct output of the output node i.

w_{ij} is the weight value between the input node j and the output node i.

α is the learning rate, it is valued between 0 and 1.

x_j is the output of the input node j.

The learning rate α determines how much the weights change per iteration, if it's the too high, the output won't converge to the correct value, while it would find it too slowly if the rate is too low, and so the learning rate value should be well determined.

In the end, the whole learning process using the delta rule can be summarized in the following steps:

1. Initialize the weights at adequate values.
2. Take the input from the training data of { input, correct output } and enter it to the neural network. Calculate the error of the output, y_i, from the correct output, d_i , to the input.
$$e_i = |y_i - d_i| \tag{2.6}$$
3. Calculate the weight updates according to the following delta rule
$$\Delta w_{ij} = \alpha * e_i * x_j \tag{2.7}$$
4. Adjust the weights as:
$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} \tag{2.8}$$
5. Perform Steps 2-4 for all training data.
6. Repeat Steps 2-5 until the error reaches an acceptable tolerance level [1].

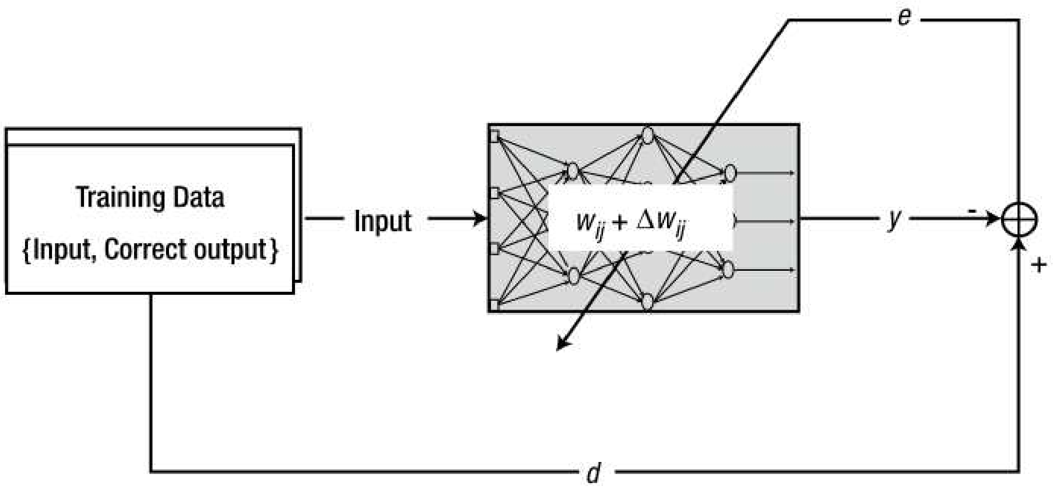


Figure. 2.11: Training of a single layer neural network [1].

The delta rule displayed so far is rather simplified, there is a more generalized form for it, one that can be used with any activation function, it is expressed as follows:

$$w_{ij} \leftarrow w_{ij} + \alpha * \delta_i * x_i \qquad (2.9)$$

Where δ_i stands for the error and is defined by the following equation:

$$\delta_i = \varphi'(v_i) * e_i \qquad (2.10)$$

Where φ' is the derivative of the activation function.

There are multiple schemes to calculate the weight update Δw_{ij}, three of them are available for the supervised learning of a neural network, and they are SGD, Batch and Mini Batch method.

6.1.1. Stochastic Gradient Descent (SGD)

The SGD calculates the error for every single data point and adjusts it's weight immediately, and so it will go on updating the weights point by point. This method adjusts the weights using the follow equation, and is it would imply, the previous delta rules are based on SGD:

$$\Delta w_{ij} = \alpha * \delta_i * x_j \qquad (2.11)$$

The following figure shows how the weight update of the stochastic gradient descent is related to the entire data:

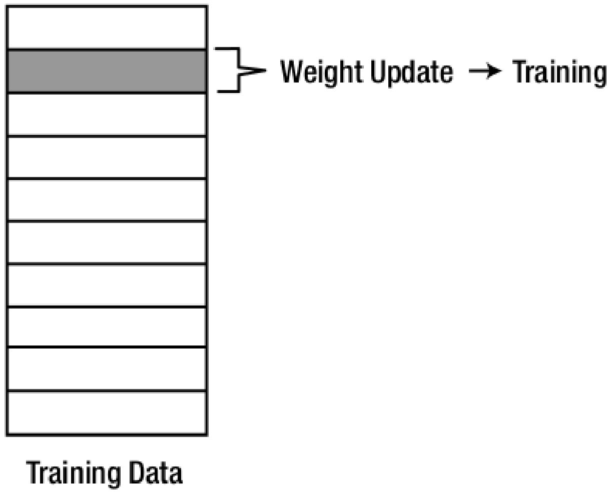


Figure. 2.12: Weight Updating In SGD [1].

6.1.2. Batch Method

Unlike the SGD method, the Batch method, calculates each weight update for all the errors of the training data, then the average of the weight updates is used to adjust the weights, and so this method uses all training data and updates the weights only once, however, because it calculates an average weight update to use for all data points, this method takes a lot of time for training. The following figure demonstrates how the method works :

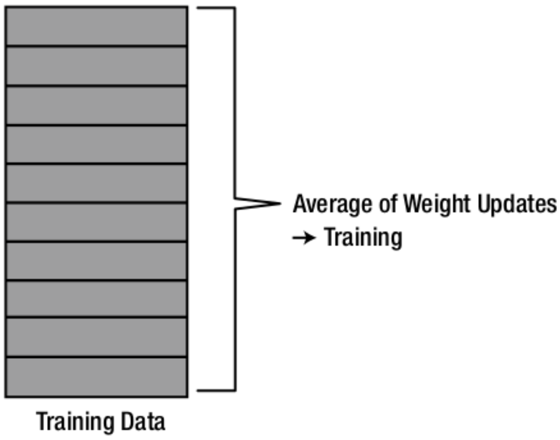


Figure. 2.13: Weight Updating In Batch Method [1].

The average weight update is calculated by the following equation:

$$\Delta w_{ij} = \frac{1}{N}\sum_{k=1}^{N} \Delta wij\,(k) \qquad (2.12)$$

Where $\Delta w_{ij}(k)$ is the weight update of the k-th training data point, and N is the total number of data points.

6.1.3. Mini Batch Method

Now this method comes in between the previous two, it is a mixture of both, it takes stability from the batch method, and speed from the SGD, how it works is simple, it selects a part of the training data and uses it for training in the batch method, so it calculates average weight updates but for a part of the data, so it takes less time to calculate, and it does the same each time until it has went through all of the data points, the following figure illustrates the mini batch method:

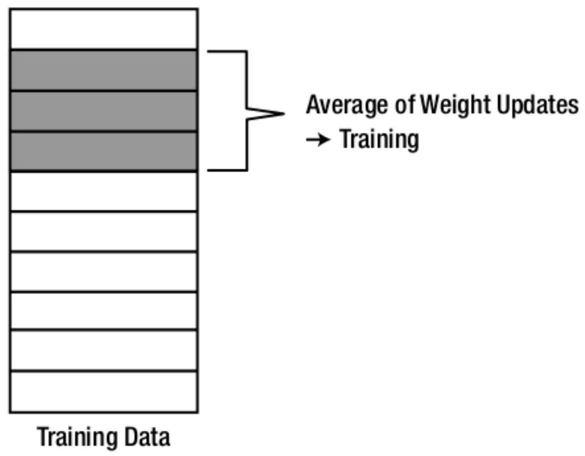


Figure. 2.14: Weight Updating In Mini-Batch Method [1].

Since this method is fast and stable, it is often used in deep learning since it uses a large amounts of data.

6.1.4. Comparison Between SGD And Batch Methods

When it comes to performance and response time, the Batch method makes for a stable network that performs pretty well in training, however it takes long periods of time in training because of all the calculations, while the stochastic gradient descent's approach is the opposite, trains quickly, but at the cost of some stability. As explained before, the SGD method cycles down with every data point while the Batch method uses the whole training data set at once, every time the neural network goes through all of the training data is called an epoch, the Batch method only has one training process in each epoch, while the SGD has N training processes per epoch, where N represents the number of data points.

As was elaborated before, the mini-Batch method is a mixture of both previous methods, it is faster than the Batch method and more stable than SGD, the number of training processes per epoch in this method is higher than 1 and lower than N. Below is a comparison between Batch and SGD methods done by Phil Kim in “MATLAB Deep Learning” [1], the program used in this example trains a neural network 1,000 times for each method, then shows the graph below, and as can be observed, the neural network learns faster with SGD.

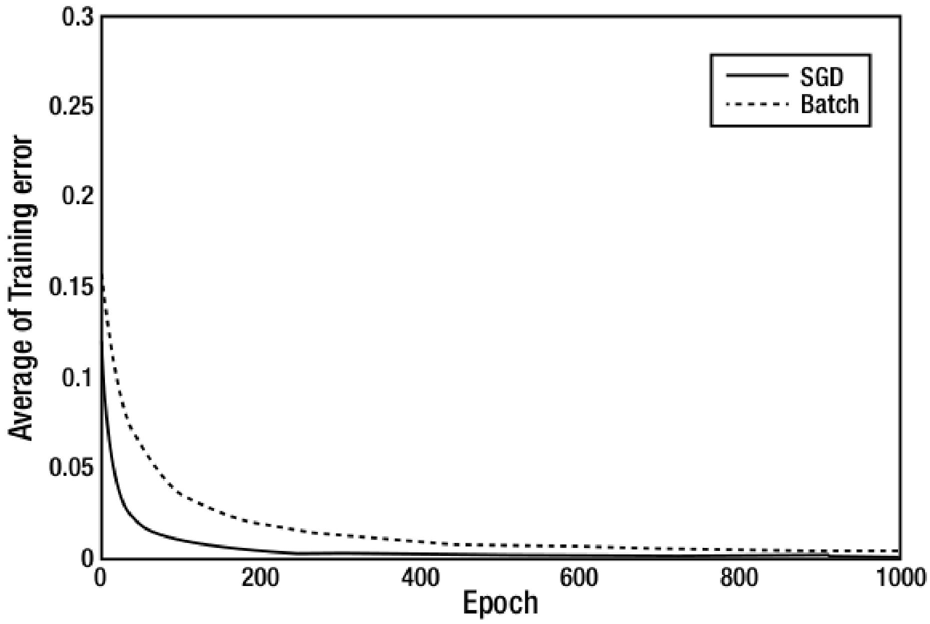


Figure. 2.15: Comparison Between Batch and SGD Methods [1].

6.1.5. Limitations Of Single Layer Neural Networks

As simple as it is to learn, a single layer neural network cannot be used to solve complicated problems, at best it is only capable of solving linear problems and it is unable to deal non-linear equations, which has lead in time to the creation of multi-layer neural networks.

Input/predicted output	Correct Output
(0, 0)/ 1	0
(0, 1)/1	1
(1, 0)/ 1	1
(1, 1)/ 1	0

Table. 2.1: Input and Output Of The XOR Problem [1].

The example below is taken from “MATLAB Deep Learning” [1], it shows how a single layer neural network cannot solve an XOR example, we use the sigmoid function as an activation function and we have the following data set:After the code has been run with this input, we get the following results, as they would suggest, the network has failed to give any correct results for this problem.

$$\begin{bmatrix} 0.5297 \\ 0.5000 \\ 0.4703 \\ 0.4409 \end{bmatrix} \quad \Leftrightarrow \quad D = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

The reason why the neural network was unable to get correct training results, is because the data points cannot be separated linearly, but they require a complex curve as the figure below shows, such non-linear equations require a multi-layer neural network.

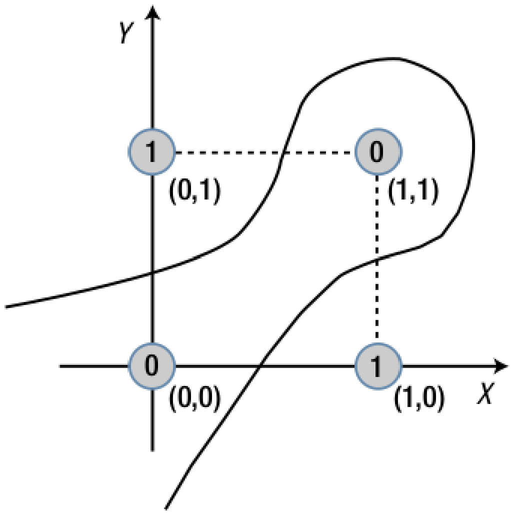


Figure. 2.16: The XOR Problem Requiring A Non-Linear Solution [1].

6.2. Multi Layer Neural Networks

To overcome the limitations of single layer neural networks, multi-layer neural networks where developed, however, it wasn't that simple, it took decades for the multi-layer architectures to finally evolve, mainly because of the learning rules, since neural networks can only store data through learning it, a neural network that cannot be trained was useless, and so we had to wait until a proper learning rule was found.

Single layer neural networks had the delta rule as their representative learning rule, it worked effectively with the simple architecture, however it was unusable for multi-layer neural networks, and that is because of the error, the delta rule calculates the error between the output of the neural network and the correct output, but it has no way of correcting the error in between the hidden layers and the output and input layers, and the training data doesn't provide correct outputs for the hidden layers too. In 1986, the back-propagation algorithm was introduced, it came as a solution to the previous problem, it provided a systematic method to calculate the error in the hidden layers, and after that error is determined, the delta rule could be applied on the neural network.

As the name would suggest, in the back-propagation algorithm, the error starts from the output layer and moves backward through the neural network until it reaches the hidden layer next to the input layer, to put it simply, in the opposite direction of the input, the following figure illustrates back-propagation:

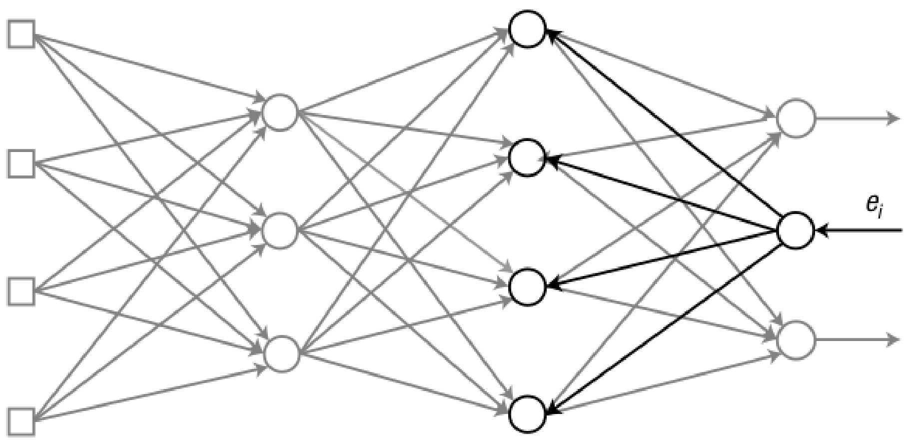


Figure. 2.17: Illustration Of The Back-Propagtion Algorithm [1].

6.2.1. Back-Propagation Algorithm

To train a multi-layer neural network, we first need to calculate the weighted sums, use the activation function to get the output of each node across the multiple layers of the network until we finally get the output of the output layer, after that is out of the way, we start using the back-propagation algorithm, first we calculate the delta $\boldsymbol{\delta_i}$of each node in the output layer, delta is defined by the following equations :

$$\boldsymbol{e_i = |y_i - d_i|} \tag{2.13}$$

$$\boldsymbol{\delta_i = \varphi'(v_i)*e_i} \tag{2.14}$$

As it can be observed, these are the same equations used for single layer neural networks. After $\boldsymbol{\delta}$ has been calculated, it is propagated backwards into the next hidden layer, it gets treated as the value of the node and it gets multiplied by the weight and summed up with the bias, the result of this calculation is the error of the hidden node, the following equation is an example, bias is ignored for convenience and the network contains two input nodes and one hidden layer:

$$e_1^{(1)} = w_{11}^{(2)} * \delta_1 + w_{21}^{(2)} * \delta_2 \quad (2.15)$$

$$\delta_1^{(1)} = \varphi'(v_1^{(1)}) * e_1^{(1)} \quad (2.16)$$

$$e_2^{(1)} = w_{12}^{(2)} * \delta_1 + w_{22}^{(2)} * \delta_2 \quad (2.17)$$

$$\delta_2^{(1)} = \varphi'(v_2^{(1)}) * e_2^{(1)} \quad (2.18)$$

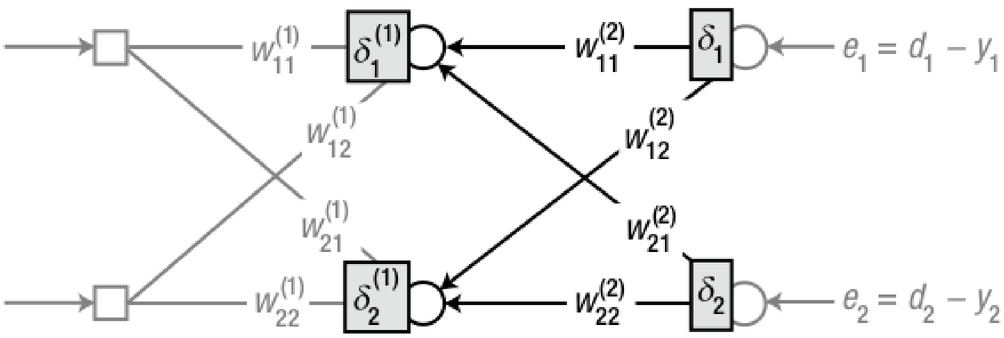


Figure. 2.18: Calculations Of $\delta_i^{(k)}$ [1].

The weight updates are calculated the same way they are in single layer neural networks, in summary, the back-propagation algorithm follows these steps [1] :

1. Initialize the weights with adequate values.
2. Enter the input from the training data { input, correct output } and obtain the neural network's output. Calculate the error of the output to the correct output and the delta, δ, of the output nodes.

$$e = |y - d|$$

$$\delta = \varphi'(v) * e$$

3. Propagate the output node delta, δ, backward, and calculate the deltas of the immediate next (left) nodes.

$$e^{(k)} = W^T * \delta$$

$$\delta^{(k)} = \varphi'(v^{(k)}) * e^{(k)}$$

4. Repeat Step 3 until it reaches the hidden layer that is on the immediate right of the input layer.
5. Adjust the weights according to the following learning rule.

$$\Delta w_{ij} = \alpha * \delta_i * x_j$$

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$$

6. Repeat Steps 2 to 5 for every training data point.
7. Repeat Steps 2 to 6 until the neural network is properly trained.

6.2.2. Momentum

When it came to weight adjustment and updating, what we previously shown was the simplest form of weight updating, still there are other forms of weight adjustment which help increase the speed and stability of the training process of a neural network, one of the forms used for weight adjustment is momentum.

The momentum, ***m*** is a term that gets added to the delta rule, it drives the weight adjustment in a certain direction, it is defined by the following equations:

$$\Delta w = \alpha * \delta * x \qquad (2.19)$$

$$m = \Delta w + \beta * m^{-} \qquad (2.20)$$

$$w \leftarrow w + m \qquad (2.21)$$

Where, m^{-} is the previous momentum and β is a constant $0<\beta<1$, the following formulas show how momentum grows over time:

$$m(0) = 0$$

$$m(1) = \Delta w(1) + \beta * m(0)$$

$$m(2) = \Delta w(2) + \beta * m(1)$$

$$\vdots$$

$$m(N) = \Delta w(N) + \beta * m(N-1) \qquad (2.22)$$

As it can be observed, the previous weight update gets added each time, however since $\beta<1$, the influence of older weight updates on the momentum dissipates over time, but since the old weight updates do exist in the equation, the newer weight updates would still be affected by them, and that ensure better stability in the learning process, also, the momentum increases with every iteration, which increases the value of the weight update, and that results in the learning rate increasing, and that speeds up the learning process.

6.3. Convolutional Neural Networks

Multi-layer neural networks are used to solve non-linear models, as was mentioned before, there are two types of multi-layer neural networks, shallow are neural networks with a single hidden layer, and deep, which have multiple hidden layers, this sub-type can solve complicated problems.

Convolutional neural network or ConvNet is a deep neural network that is employed in image recognition, this architecture imitates the brain's visual cortex and recognizes images, it differs in concept to other neural networks to a certain degree, but it is highly reliable and preforms well since it is specialized for image recognition.

Image recognition is basically classification, it's like recognizing handwritten letters from an image, a class in this example would a be letter such as “A” or “B”, and so it would recognize which letter is which. The recognition process works by extracting features, then matching to find the correct class for the given sample, feature extractors cost a lot of time and resources to design, and they don't yield the best results, of course these feature extractors do not depend on machine learning.

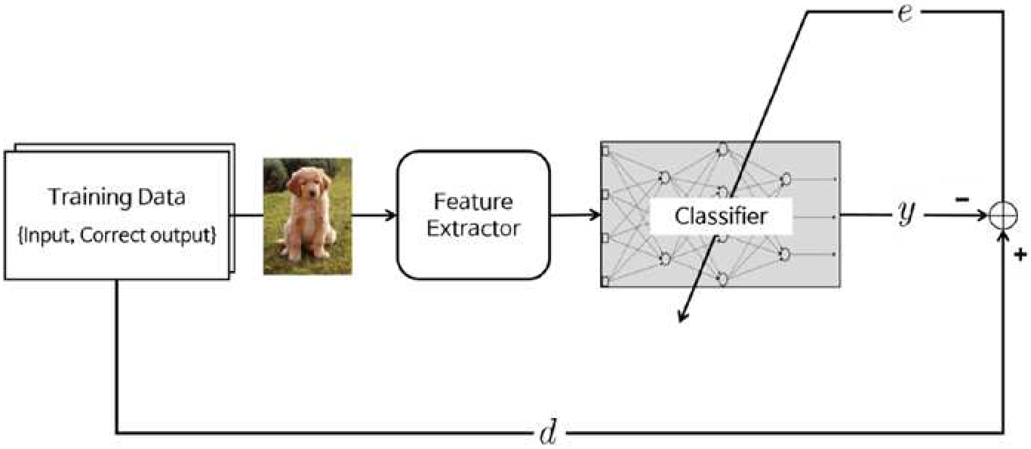


Figure. 2.19: Image Recognition Without ConvNet [1].

The ConvNet architecture includes the feature extractor, as it builds it during the learning process, which gives a great advantage since it won't be manually designed, that would reduce the cost in both time and resources. It would also yield much better results if it's feature extraction network is deeper.

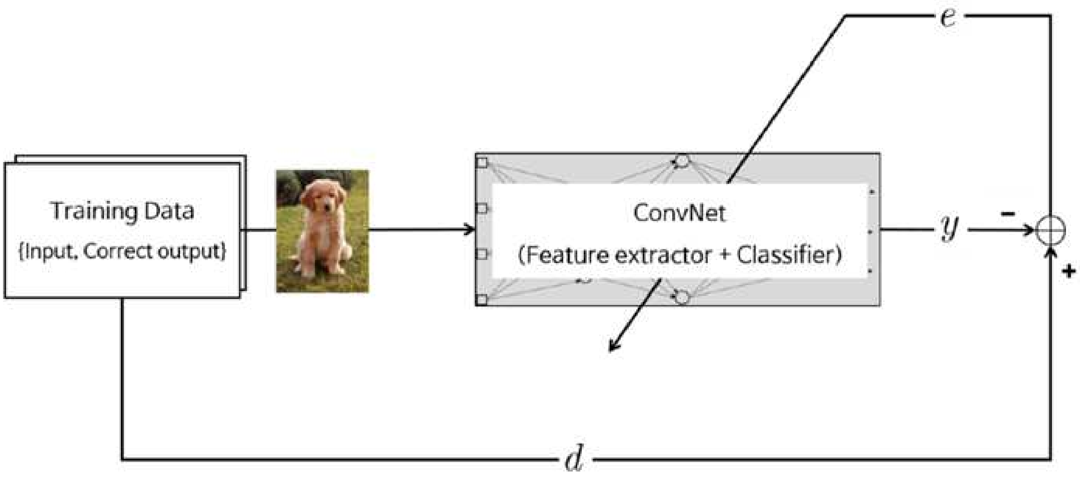


Figure. 2.20: Image Recognition With ConvNet [1].

When working on feature extraction, ConvNet uses two layers, convolutional layer and pooling layer, each layer has a specific function. The convolutional layer applies the convolution operation on the image in order to convert it, it can be considered as a set of digital filters, meanwhile, the pooling layer reduces the dimensions of the image by combining neighboring pixels into a single one; these two layers mark a point of difference between ConvNet and other neural networks.

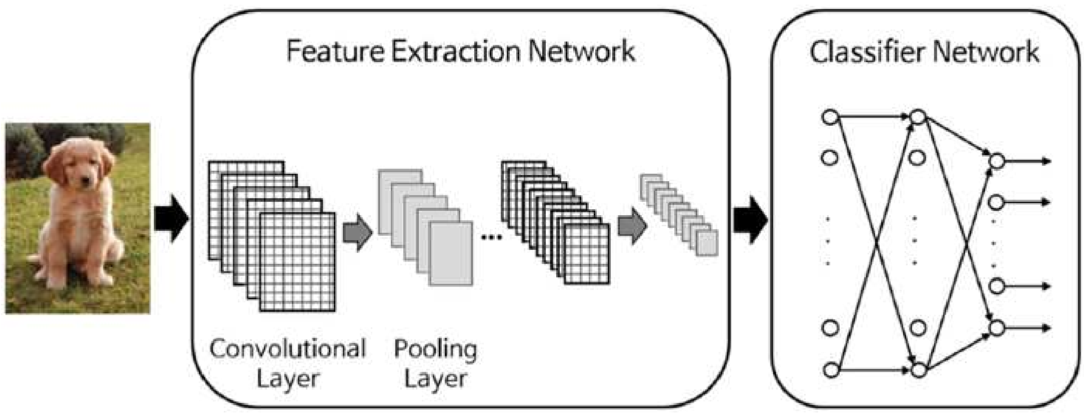


Figure. 2.21: Overview of the ConvNet Architecture [1].

There are other variants of convolution neural networks, such as AlexNet and ResNet which we will described in the next sub-sections.

6.3.1. AlexNet

AlexNet is a pre-trained convolutional neural network, it has 8 layers, and it can classify images into 1000 object categories including desk objects (pens, keyboards ...etc) and multiple animals, it has an input size of 227x227.

Deep Neural Networks which results in improved accuracy and performance. The intuition behind adding more layers is that these layers progressively learn This network has been on over 1 million images, so it is rich when it comes to feature recognition. AlexNet uses images as an input and gives a label for the object in the image as an output, it has multiple versions for the pre-trained network, a pre-trained network can be used in transfer learning.

6.3.2. ResNet

It is short for Residual Network, which is a deep convolutional neural network, it has been successful since its introduction in 2015. Usually, stacking extra layers on a deep neural network helps improve its accuracy as it allows it to learn more features, that’s especially useful in object recognition since each layer would work on detecting a certain feature, however, a limit to the amount of extra layers has been found. At certain point, adding more layers would only increase the error, the figure below shows the error for two convolutional neural networks, one has 20 layers and the other 56 layers.

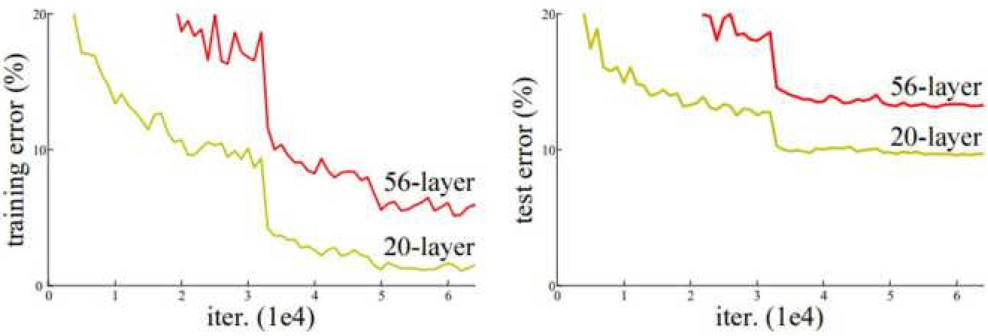


Figure. 2.22: Error % When Adding Extra Layers Without ResNet [4]

As is observable, the error is much higher for the 56 layer network in both training and testing, adding too many layers has degraded the performance. ResNet helps with this issue, as it makes deeper CNNs perform better without any limit to the extra layers, ResNets are made of residual blocks as the one shown in the following figure:

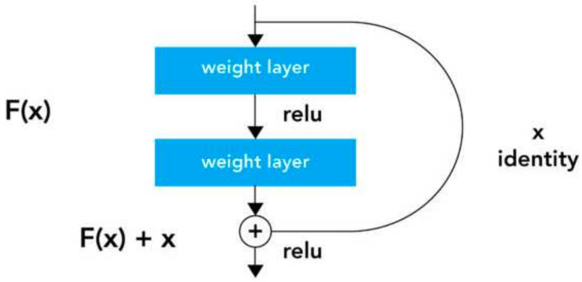


Figure. 2.23: Residual Block [4].

The residual block has a connection that allows the network to skip certain layers in between, this skip connection changes the output, the following equations show the output without then with the skip connection:

$$y = \varphi(v) = \varphi(w*x + b) \quad \text{(Without skip connection)} \quad (2.23)$$

$$y = \varphi(x) + x \quad \text{(With skip connection)} \quad (2.24)$$

The skip connections solve the problem of vanishing/exploding gradients as they allow shortcuts for the gradient to flow, they also allow the model to learn identity functions which in turn ensure that the higher layer would perform as good or better than lower layer and not worse, and highly increases the levels of performance, the following figure shows how ResNet improves performance when it comes the error, the networks are 18 and 34 layers deep.

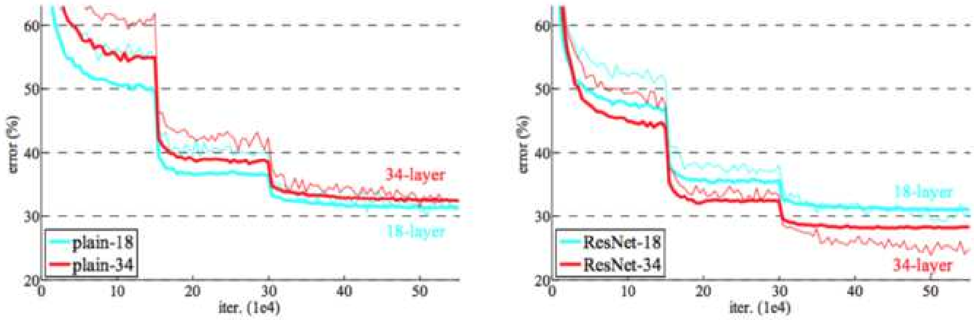


Figure. 2.24: Error % With And Without ResNet [4].

The architecture of ResNet is a 34 layer residual network, it is based off a 34 layer plain network in which skip connections are added, it has been inspired by VGG-19 architecture, the figure next illustrates the architecture.

7. Transfer Learning

When building a system that cannot learn incrementally, in other words, it doesn't update itself on change in data, we could use transfer learning. Basically, we need to give the system all the data it needs for training and testing. For example, we are building a fingerprint recognition system, we get a full database of fingerprints, all these fingerprints are already classed, we use this database to train our system on fingerprint recognition, once the training process it's done, we launch the system with a new database. Using large amounts of data during the training process would take very long periods of time, so this type of training is always done offline, so we are simply teaching the machine through giving it corrected examples, it's like learning how to solve math problems by looking at corrected exercises.

8. Palmprint Recognition using Deep Learning

There are multiple ways to build a palmprint recognition system, deep learning of course, offers the most reliable methods; possible one of the best is using convolutional neural networks to build the system.

The training process of a palmprint recognition system is usually done using transfer learning, there are multiple public palmprint databases available which can be used to apply this type of training such as:

- CASIA Palmprint database.

A good choice of convolutional neural network to use in the process is AlexNet, the pre-trained neural network has 1000 classes, so it can be used to classify a good number palmprints, for example the CASIA database has over 5,500 palmprint images that belong to 620 individuals, each individual is treated as a class in the neural network. We will further explain the protocols used to train our palmprint recognition system in the next chapter.

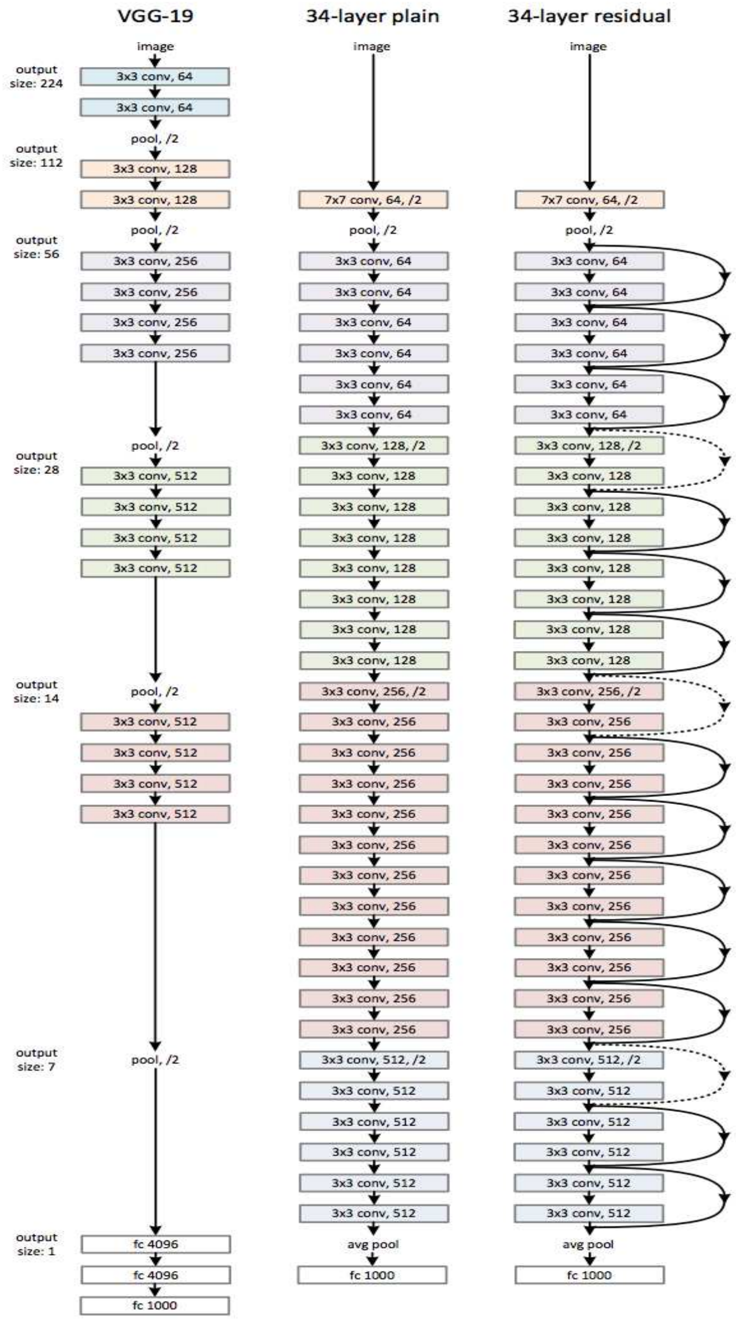


Figure. 2.25: ResNet Architecture [4].

9. Conclusion

Throughout this chapter, we explained the basic notions of Machine Learning and it's types, then we went through Deep Learning and thoroughly explained it's main algorithms, while focusing on the supervised learning of neural networks with their different types, we also displayed important convolutional neural networks AlexNet and ResNet, then we briefly explained transfer learning, and finally we reviewed palmprint recognition with the use of deep learning, highlighting the importance of this programming technique.

References

[1] MATLAB Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence by Phil Kim.

[2] Machine Learning Step-by-Step Guide to Implement Machine Learning Algorithms with Python by Rudolph Russell.

[3] Mathworks website page on AlexNet: https://www.mathworks.com/help/deeplearning/ref/alexnet.html;jsessionid=e89ccc2ed7a15aa3eed2f15e9a09

[4] K. He, X. Zhan et al., "Deep Residual Learning for Image Recognition" Microsoft Research, (2015) 1-7.

[5] W. Gong, X. Zhang et al., " Palmprint Recognition Based on Convolutional Neural Network-Alexnet", Proceedings of the FedCSIS. Leipzig, 18 (2019) 313-316.

[6] P. Cunningham, S. J. Delany . "k-Nearest neighbour classifiers", Mult Classif Syst. Springer; (2007) 34: 1–17.

[7] T. Evgeniou and M. Pontil, " Support Vector Machines: Theory and Applications", Advanced Course on Artificial Intelligence (ACAI 99) Chania, Greece; *Workshop on Support Vector Machines: Theory and Applications*; (2009).

[8] F. A. C. Azevedo, L. R. B. Carvalho et al., "Equal Numbers of Neuronal and Nonneuronal Cells Make the Human Brain an Isometrically Scaled-Up Primate Brain", The Journal of Comparative Neurology 513:532–541 (2009) 533-535.

1. Introduction

There are multiple ways to build a palmprint recognition system, each method has it's pros and cons, from its speed and accuracy of response, to its cost effectiveness. Using machine learning to build such a system is a very robust one, as it offers a lot of choices to ensure high performance in image recognition due to its advanced algorithms.

In this chapter, we are going to use deep learning to build a palmprint recognition system, we would employ transfer learning, and a pre-trained version of AlexNet in the process while training on two different data-sets.

2. Palmprint Databases

There are multiple public palmprint databases that are usable for transfer learning, each one offers a certain type of image from low to high resolution, and from 2D to multi-spectral images, along with the image sizes and the amount of samples, examples of these databases are the CASIA palmprint database along with PolyU 2D+3D and PolyU multi-spectral databases.

3. Evaluation Protocol

The protocol used in this research bases on the use of a pre-trained version of AlexNet. We basically use the same architecture but with a different number of classes according to the database in use; AlexNet has 1000 classes while CASIA and PolyU databases have 620 and 400 classes respectively. Our new network is named EarNet.

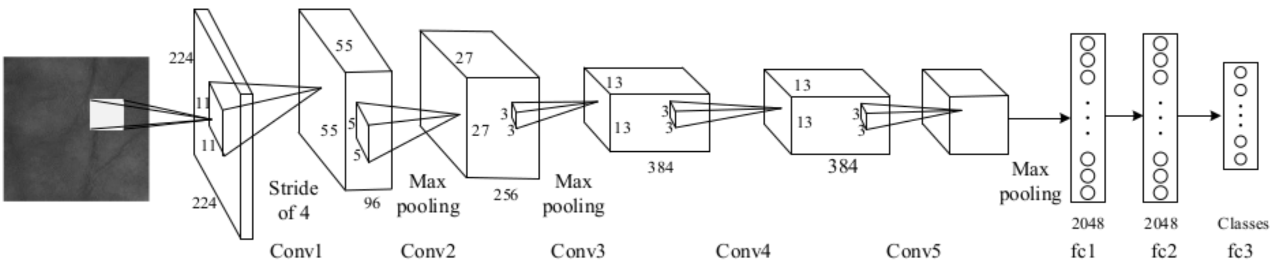


Figure. 3.1: The architecture of Alexnet [2].

The input images provided by the two databases were adjusted to fit the network as well, so the size of all inputs was changed to 227x227x3. As for the amount of training data we followed two approaches :

- With the CASIA database we divided our data set into two, a part for training and the other for testing; we did multiple experiments, changing the division ratio each time.
- With the PolyU2D database, we used it's first session to train the network, then we used the second session of the database to test it.

The following section provides more information about the databases, from the amount of samples to the type of the images, to the results of the different experiments done on both. The machine used in these experiments has an i3-7100 3.90GHz processor with 16GB of RAM, while using MATLAB 2019b on Windows 10 64bit.

3.1. CASIA Database

This database offers 5,501 palmprint images divided into 620 classes, the images has been taken from both left and right hands of 312 individuals, all the image are 8bit gray-level 128x128 JPEG files, these palmprint images are used to train a contactless palmprint recognition system.

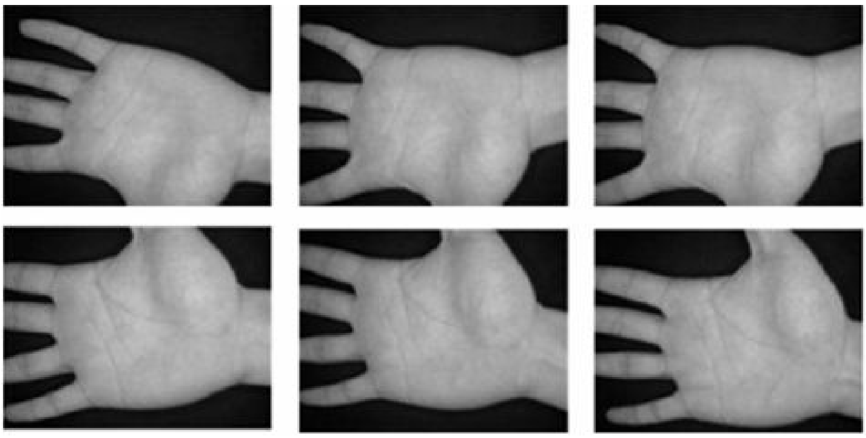


Figure. 3.1: Six Images From The Database [1].

3.1.1. Results And Discussion

In this training process we used an i3-7100 3.90GHz with 16GB of RAM, Windows 10 64 bits, while using MATLAB 2019b, we also used a training ratio of 8:2 (80% of the data for training and 20% for testing), according to the results we made the following graph;the final identification rate of the network is***97.44%***.

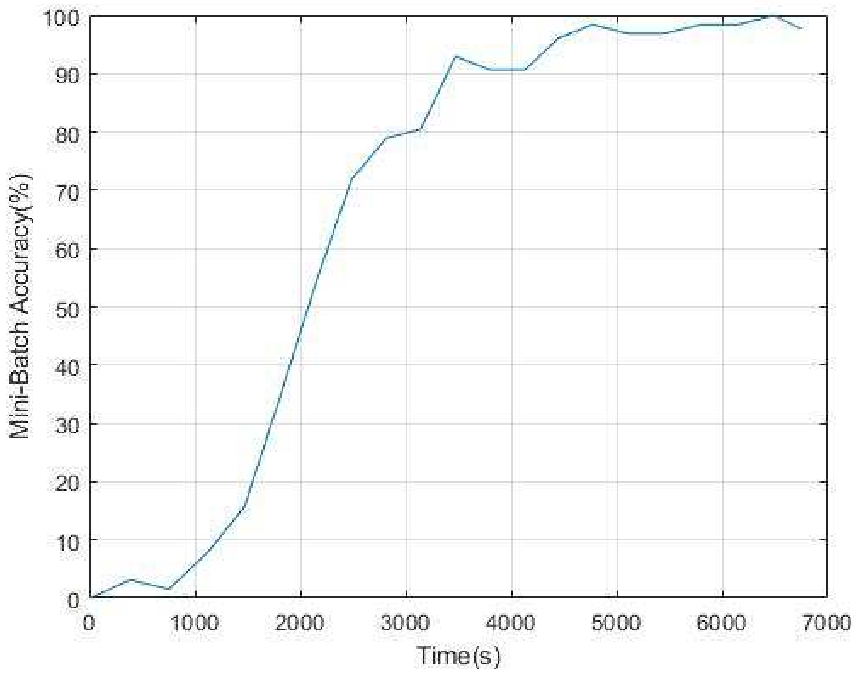


Figure. 3.2: Training Accuracy (%) With 8:2 Ratio On CASIA.

As was expected, the system took nearly 2 hours to train and test, which is a fairly long amount of time, it converged slowly but it reached a ***100%*** accuracy at the 29th epoch, and it's mini-batch loss was pretty low at ***0.0494***.

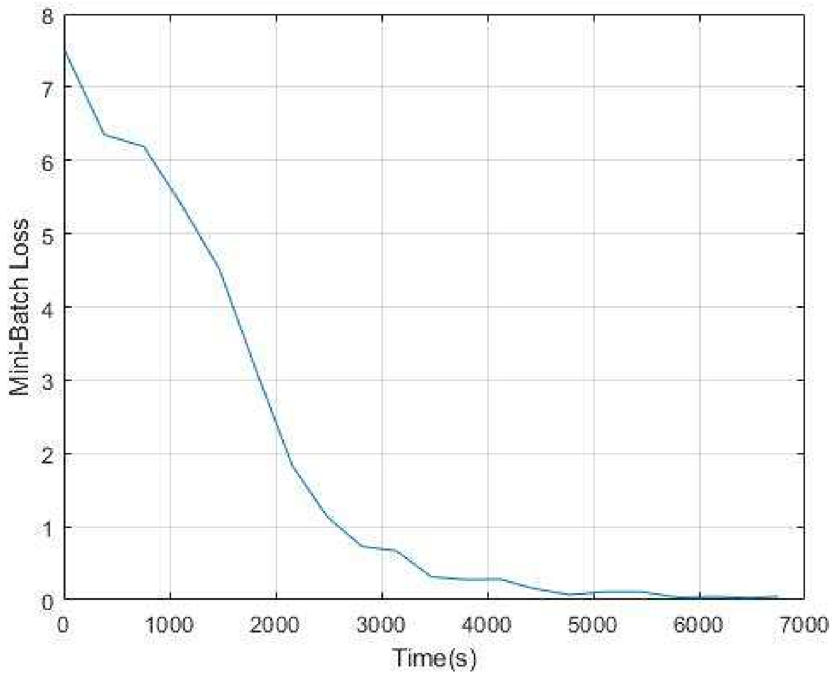


Figure. 3.3: Training loss with 8:2 Ratio on CASIA database.

Using the same machine as before, we did two more experiments, we changed the training ratio from 8:2 used in the first one, to 7:3 and 5:5 respectively, increasing the amount of the data used for testing and reducing the training amount, the following graphs displays the results, also the second experiment returned an identification rate of ***97.87%*** while the third returned ***96.09%***.

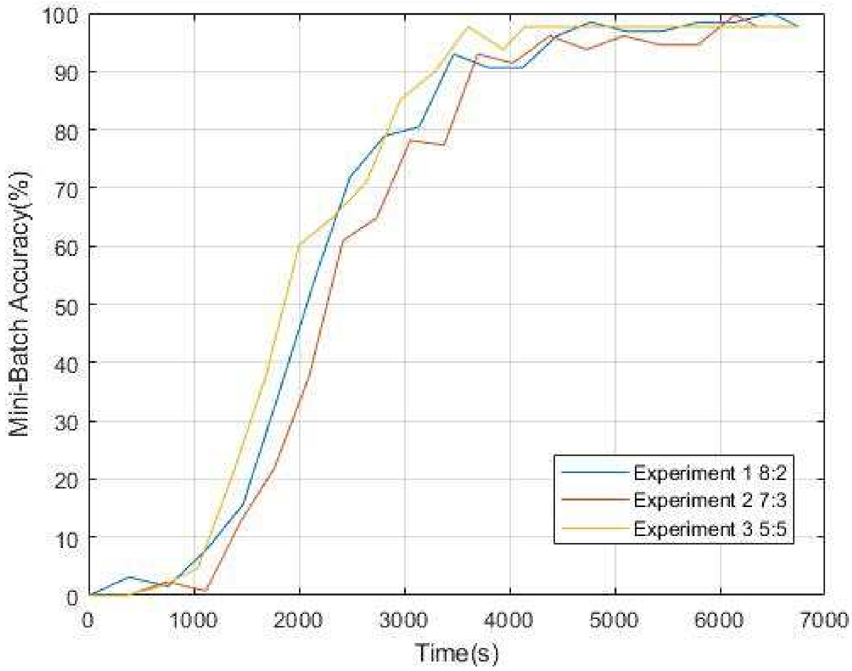


Figure. 3.4: Training Accuracy For All Three Experiments On CASIA.

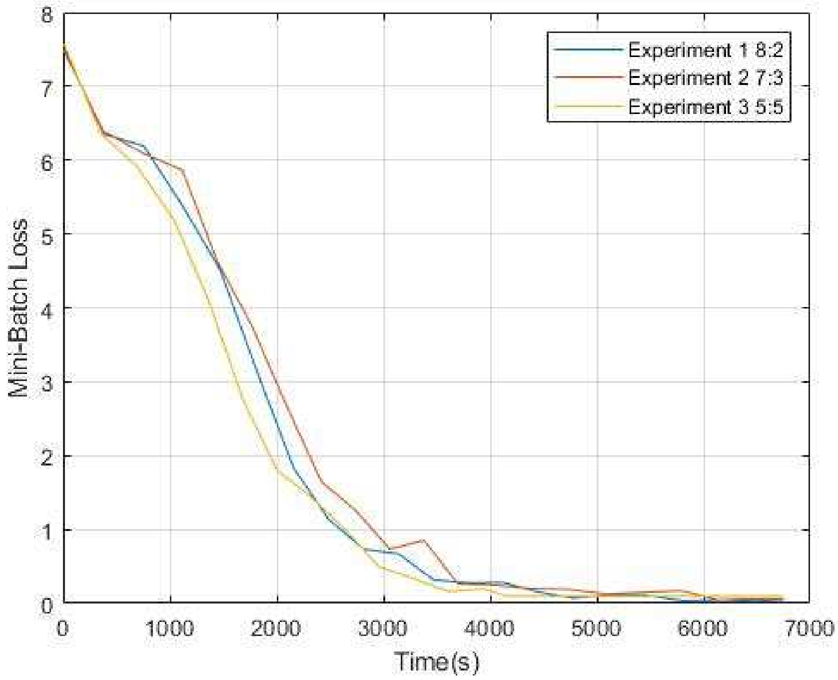


Figure. 3.5: Training Loss For All Three Experiments On CASIA.

Both experiments took a shorter period of time to finish training than the first one did, they also did less iterations, the final accuracy result is the same for all experiments at ***97.66%***, however the mini-batch loss is different, turns out the first one had the lowest loss rate at ***0.0494*** while the third experiment, which is the one with the lowest amount of training data had the highest loss rate at ***0.0974***, so in the end, reducing the amount of data set for training would reduce the training time but would degrade the performance of the system as a whole.

3.1.2. Comparative Study

Benjoudi et al. [4] used the CASIA database with a different protocol, the results of their comparative study are in the following table.

Methods	Nbr classes	Training	Test	Descriptors	IR(%)
[5]	310	First 5 images	Remaining	Texture pattern	96.52
[6]	620	First 4 images		LLDP_GABOR ResNet50	93.00 95.03
[7]	620	First 5 images		Principal lines Texture patterns Intramodal	92.98 96.00 98.00
[4]	620	First 4 images		BSIF_26	98.18
[4]	620	First 4 images		BSIF_26	99.34

Table. 3.1: Comparative results showing recognition rate of the proposed schemes and recently proposed methods on CASIA database [4].

Our experiments with the CASIA database has returned good results as well, the best identification rate acquired from our study was ***97.87%***, it was acquired when using the 7:3 training ratio, when put along with the rest of the methods in table. 3.1 it ranks third as the highest identification rate.

3.2. PolyU2DDatabase

This database contains nearly8000 palmprint images taken from 386 different palms divided into two sessions, each session has 3889 images, the images are 8bit gray-level images of 135x135. This database is used to train palmprint recognition systems that scan using contact.

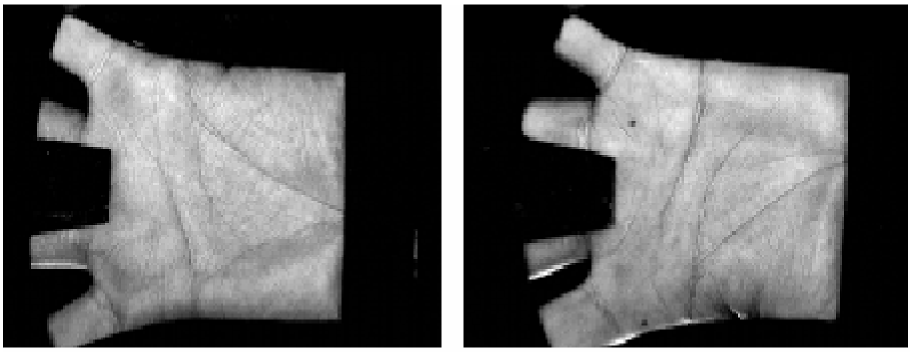


Figure. 3.7: Images From PolyU2D Database [3].

3.2.1. Results And Discussion

We used the first session of the database as a training set, and the training process took more than an hour (01:14:57); after the training process was done, loaded the second session and used it to test EarNet, and the result of the testing was a identification rate of ***99.97%*** which is much higher than the rates acquired when using the CASIA database. The following graphs show the development of the mini-batch accuracy and loss throughout the training process.

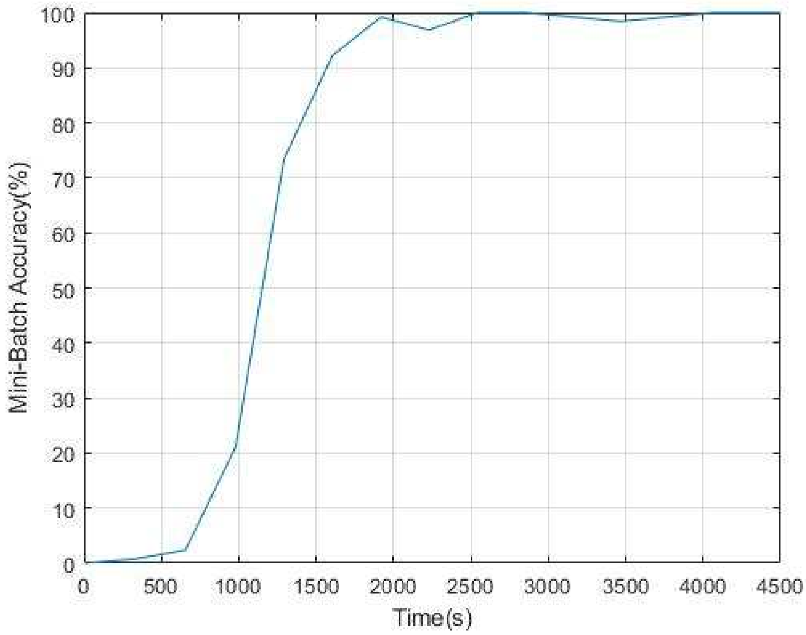


Figure. 3.6: Training Accuracy (%) For PolyU2D database.

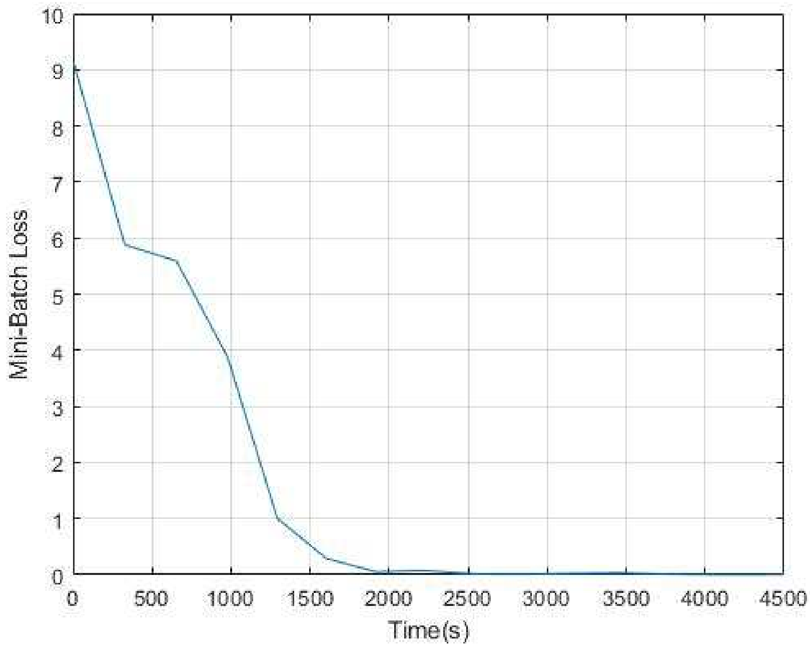


Figure. 3.7: Training Loss For PolyU2D database.

From this experiment we can see that our network gave great results with the PolyU2D database, the mini-batch loss was very low at only***0.0088***.

3.2.2. Comparative Study

Benjoudi et al. [4] also used the PolyU2D database, the following table shows the results of their comparative study.

Methods	Nbr classes	Protocol	N or K	Descriptors	IR(%)
[5]	386	F	N=5	Texture pattern+ principle lines shape	96.99
[8]	386	G	K=2	RDF	99.69
[9]	374	C	N=3	DOC ($n\theta$=12)	99.74
[10]	386	B	N=3	$LLDP_G$/9216 $LLDP_M$/9216	100 100
[11]	386	E	N=4	LDDBP	99.96
[4]	386	C B	N=3	BSIF_26/L=5 BSIF_26/L=5	100 100
[4]	386	C B	N=3	BSIF_26/L=5 BSIF_26/L=5	100 100

Table. 3.2: Comparative results showing identification rate of the proposed schemes and recently proposed methods on PolyU2D database [4].

The results of our experiment come out good enough compared to the other results, the identification rate we got of ***99.97%*** proves the quality of our study.

4. Conclusion

In this chapter, we employed the transfer learning technique using the pre-trained convolutional neural network AlexNet to build a system capable of recognizing palmprints. We also made use of two different databases, CASIA and PolyU, and we did comparative studies for each database.

References

[1] http://www.cbsr.ia.ac.cn/english/Palmprint%20Databases.asp

[2]W. Gong, X. Zhang et al., " Palmprint Recognition Based on Convolutional Neural Network-Alexnet", *Proceedings of the FedCSIS. Leipzig*, 18 (2019) 313-316.

[3] S. K. Panigrahy, D. Jena et al., "A Rotational-and Translational-Invariant Palmprint Recognition System",*First International Conference on Data Engineering and Management, Tiruchirapalli*, (2008) 383.

[4] S. Benjoudi, H. Bourouba et al., "Palmprint identification performance improvement via patch-based binarized statistical image features", Journal of Electronic Imagining 28(5), 053009 (2019).

[5] R. Mokni, H. Drira, and M. Kherallah, "Combining shape analysis and texture pattern for palmprint identification," *Multimed. Tools Appl*. 76, 23981–24008 (2017).

[6] L. Fei et al., "Feature extraction methods for palmprint recognition: a survey and evaluation," *IEEE Trans. Syst. Man Cybern.* 49, 346–363 (2019).

[7] M. Hammami, S. B. Jemaa, and H. Ben-Abdallah, "Selection of discriminative sub-regions for palmprint recognition," *Multimedia Tools Appl.* 68(3), 1023–1050 (2014).

[8] D. Tamrakar and P. Khanna, "Noise and rotation invariant RDF descriptor for palmprint identification," *Multimedia Tools Appl.* 75, 5777–5794 (2016).

[9] L. Fei et al., "Double-orientation code and nonlinear matching scheme for palmprint recognition," Pattern Recognit. 49, 89–101 (2016).

[10] Y. Luo et al., "Local line directional pattern for palmprint recognition," Pattern Recognit. 50(2), 26–44 (2016).

[11] L. Fei et al., "Local discriminant direction binary pattern for palmprint representation and recognition," *IEEE Trans. Circuits Syst. Video Technol*.

Throughout this research, we went around the different aspects of biometry, identity verification, machine learning and deep learning to build a biometric system that works using palmprint recognition. Biometry has proven to be quite the important science, it has a lot of applications in multiple fields, it has developed exponentially over the last decade, and it can only grow more in the future. Palmprint recognition is very useful way to verify identities, the accessibility of the modality and its overall robustness makes biometric systems based on it greatly reliable. The state of development that neural networks has reached since their first emergence during the late 20^{th} century has provided solutions to many complicated mathematical problems, and has allowed us to build a robust palmprint recognition system, the application of transfer learning using the pre-trained CNN AlexNet has led to a system that accurately recognizes palmprints. Overall though, the system can still be developed to get better results.