T.C

ALTINBAS UNIVERSITY

Electrical and Computer Engineering

FINGERPRINT PATTERN RECOGNITION SYSTEM BASED ON MODIFY MULTI-CONNECT ARCHITECTURE (MMCA)

Layth Kamil Adday AL-Majmaie

Doctor of Philosophy

Supervisor

Prof. Dr. Osman Nuri UCAN

FINGERPRINT PATTERN RECOGNITION SYSTEM BASED ON MODIFY MULTI-CONNECT ARCHITECTURE (MMCA)

by

Layth Kamil Adday AL-Majmaie

Electrical and Computer Engineering

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

ALTINBAŞ UNIVERSITY

2019

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Osman Nuri Ucan

Supervisor

Examining Committee Members (first name belongs to the chairperson of the jury and the second name belongs to supervisor)

Assoc. Prof.Dr. Oguz Bayat	School of Engineering and Natural Science, Altinbas University	___________________
Asst. Prof.Dr. Çağatay AYDIN	School of Engineering and Natural Science, Altinbas University	___________________
Asst. Prof.Dr. Doğu Çağdaş ATİLLA	School of Engineering and Natural Science, Altinbas University	___________________
Prof.Dr. Hasan Huseyin BALIK	Air Force Academy, National Defense University	___________________
Asst. Prof.Dr. Adil Deniz Duru	Faculty of Sport Sciences, Marmara University	___________________

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Asst. Prof.Dr. Cagdas Dogu ATILLA

Head of Department

Prof. Dr. Oguz BAYAT

Director

Approval Date of Graduate School of

Science and Engineering: ____/____/____

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.

Layth Kamil Al-Majmaie

DEDICATION

To the spirit of my martyr brother.

ACKNOWLEDGEMENTS

In the name of Allah, the Beneficent and the Merciful. Praise and Gratitude be to Allah for giving me strength and guidance, so that this thesis can be finished accordingly.

I would like to thank my supervisor: Prof. Dr. Osman Nuri Ucan and Prof. Dr. Oguz Bayat. Please let me express my deep sense of gratitude and appreciation to both of you for the knowledge, guidance and unconditional support you have given me. I wish you all the best and further success and achievements in your life.

My deepest gratitude goes to my dearest parents, for their immense patience and unconditional support and encouragement throughout my life. My Wife, children, brothers, sisters and their daughters and sons: thank you very much for your prayers and encouragements. My friends and colleagues: thank you very much for what you have done for me. I thank you all for the companionship that has made this journey much easier. In fact, I do not need to list your names because I am sure that you know who you are.

I would like to thank all the Altinbas University, college of engineering, department of the computer.

Finally, I also thank the Iraqi Ministry of Higher Education and Scientific Research, the Iraqi Cultural Attaché in Ankara, University Of Technology for supporting me during my study abroad.

ABSTRACT

FINGERPRINT PATTERN RECOGNITION SYSTEM BASED ON MODIFY MULTI-CONNECT ARCHITECTURE (MMCA)

Layth Kamil Adday AL-Majmaie

PhD , Electrical and Computer Engineering , Altinbas University

Supervisor: Prof. Dr. Osman Nuri Ucan

Date: September , 2019

Pages: 84

Fingerprinting is the most widely used and recognised biometric technology for human authentication. Fingerprint authentication has a proven record as highly secure and convenient as compared to passwords. Hence, fingerprint sensing has come to be recognized as a common and product-differentiating feature in smart phones, tablets and PCs. This thesis proposes to develop a Fingerprint recognition system for authentication of persons by using a new technique, termed as ‘associative memory with modify multi-connect architecture’. This, in turn, may pave the way to develop more efficient Fingerprint systems having accuracy and lesser processing time. Further, with application of additional tranches of associative memory, such systems in the future will acquire potential to perform highly complex operations and save memory. In this thesis, three databases viz., FVC (2004) database, internal database and International NIST database 4 are used. FVC (2004) database contains 640 fingerprint patterns, while internal

database contains 2500 different fingerprint patterns; and the International NIST database 4 consists of 2000 pairs of fingerprint patterns. The proposed fingerprint recognition system has an average accuracy of 99.56% and a pattern recognition processing time of approximately 30s.

Keywords: Associative memory, Biometric and Fingerprint, modify multi-connect architecture.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

LIST OF ABBREVIATIONS

AM	:	Associative Memory
ANN	:	Artificial Neural Network
CP	:	Converged Pattern
DHGN	:	Distributed Hierarchical Graph Neuron
HNN	:	Hopfield Neural Network
MD	:	Majority Description
MMCA	:	Modify Multi-Connect Architecture
MCA	:	Multi-Connect Architecture
NN	:	Neural Network
RNN	:	Recurrent Neural Network
RGB	:	Red, Green, Blue
RMD	:	Result Majority Description
SMD	:	Save Majority Description
SVW	:	Save Vector Weights
TH	:	Threshold
TIF	:	Tagged Image File
TMD	:	Test Majority Description
TVW	:	Test Vector Weights

1. INTRODUCTION

1.1 INTRODUCTION

Biometric Identification Systems are generally utilized for novel distinguishing proof of people primarily for confirmation and ID. Biometrics is utilized as a type of personality get to the board and access control. Subsequently, utilization of biometrics in understudy participation the board framework is a protected methodology. There are numerous sorts of biometric frameworks like unique mark acknowledgment, face acknowledgment, voice acknowledgment, iris acknowledgment, palm acknowledgment and so on [1]

A fingerprint is the portrayal of the epidermis of a finger : it comprises of an example of interleaved edges and valleys. Fingertip edges advanced throughout the years to enable people to handle and grasp objects. Like everything in the human body, fingerprint edges structure through a blend of hereditary and natural elements. Truth be told, fingerprint development is like the development of vessels and veins in angiogenesis. [1, 2]

Skin on human fingerprints contains ridges and valleys which together forms distinctive patterns as shown in Figure 1.1. These patterns are fully developed under pregnancy and are permanent throughout whole lifetime. Prints of those patterns are called fingerprints. Injuries like cuts, burns and bruises can temporarily damage quality of fingerprints but when fully healed patterns will be restored. Through various studies it has been observed that not two persons have the same fingerprints, hence they are unique for every individual. [1, 6]

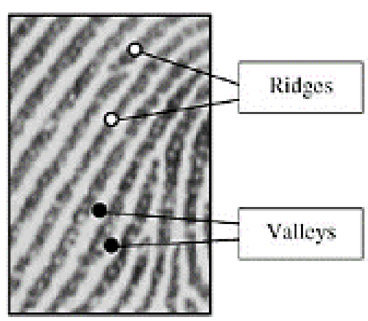


Figure 1.1: Ridges and valleys on a fingerprint image.

A unique mark is the example of edges and valleys on the outside of a fingertip. The endpoints and intersection purposes of edges are called details. It is a broadly acknowledged presumption that the particulars example of each finger is one of a kind and does not change amid one's life.

Edge endings are the focuses where the edge bend ends, and bifurcations are the place an edge parts from a solitary way to two ways at a Y-intersection. [1]

Unique mark biometric idea: Fingerprint biometric is the normally utilized most seasoned and exclusively strategy universally acknowledged as lawful technique to recognize an individual. Unique mark is the impressions of the moment edge (called as dermal) of the finger. Unique finger impression edges and valleys are exceptional and unalterable. Unique mark biometric is utilized in various applications that incorporate regular citizen and business applications like military, law requirement, drug, training, common administration, crime scene investigation, driver permit enlistment, mobile phone get to. [31, 61]

Unique mark ID is the most generally utilized biometrics advancements and is utilized in criminal examinations, business applications, etc. With such a wide assortment of employments for the innovation, the socioeconomics and condition conditions that it is utilized in are similarly as differing. Be that as it may, the recognizable proof exhibition of such framework is exceptionally touchy to the nature of the caught unique mark image. Unique mark image quality examination or appraisal is helpful in improving the presentation of unique finger impression recognizable proof frameworks. [43, 62]

In numerous frameworks it is desirable over substitute low quality images for better ones. Subsequently, image quality examination takes a significant part in image handling. The evaluation of image quality permits to tune a framework and to assess estimation precision of a given information image. [62]

1.2 DATABASE

Testing and evaluation is critical for the effective functioning of a fingerprint system. In order to evaluate the effectiveness of our proposed system, three different datasets have been used in our research work as described below:

- The performance of our proposed technique is tested on FVC (2004) database. The database contains 80 different fingers and eight impressions of each finger (80x8=640 fingerprints). The images in DB1, DB2, DB3 and DB4 are of 8 bit gray level with sizes of 640x480, 328x364, 300x480, and 288x384, respectively; and each having a resolution of 500 dpi. [35]

- The internal database contains 2500 different fingers and ten impressions of each finger. The image in internal database is of 8 bit gray level with a size of 300*300.
- The International NIST database 4 is a specific database for fingerprint recognition, which consists of 2000 pairs of fingerprint images in the format 512×512 with an 8-bit grayscale. [36]

All patterns in a database are in TIF format (Tagged Image File). TIF is immune from any degradation or loss, which offers the highest quality format for commercial work. Moreover, it is the most versatile and does not require compression

1.3 DIGITAL IMAGE [5]

It is exhibit of pixels, the estimation of every pixel is proportion of splendor of pixel of the image, these pixels are comprise advanced images, and computerized image takes of just whole number qualities, these qualities are discrete.

There are three kinds of computerized images:

- Binary image: this type of image is just two potential qualities for every pixels, which is simply dark (0) or white (1).
- Greyscale image: every pixels in image spoke to by means of 8 bits (i.e., one byte), is extending from 0 to 255.
- Color image (RGB): The shading image distinguished by using the measure of blue, green and red, these segments has a range from 0 to 255. The quantity for these bits for every pixel is 24.

1.4 GRAYSCALE IMAGE [11]

Convert an RGB images to greyscale images by using luminosity method by taking values for each pixel (i.e., Red, Green and Blue) and calculate a single value as output, which is a more sophisticated version of the average method. It also averages the values, but it forms a weighted average to account for human perception. It is more sensitive to green than other colors (i.e., green weighted most heavily). Compute gray value for each pixel by using Equation (1.1).

$$\text{Greyscale Value} = 0.2989 * R + 0.5870 * G + 0.1140 * B \tag{1.1}$$

1.5 INTEGRAL IMAGE

Can be characterized as an instrument which are uses each time a capacity from pixels force to numbers are exist, and the computation of the entirety of this capacity over a rectangular district is required. Entirety of these capacities should be determined on straight time of every square shape if vital image isn't utilized. While in necessary image, a steady time of entirety over numerous covering rectangular windows is accomplished. Each new pixel (x , y) in fundamental image (I) may be determined by using summation of every pixels to one side or more (x, y) pixel in first image (f). Equation (1.2) is utilized to figure necessary image for every pixel. [58]

$$I(x,y) = f(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1) \quad (1.2)$$

The Figure1.2 shows the calculation of a necessary image. Have indispensable image, whole of capacity for any square shape with upper left corner (x1, y1), and with lower right corner (x2, y2) can be processed in steady time utilizing the Equation (1.3). [51]

$$\sum_{x=x1}^{x2}\sum_{y=y1}^{y2} f \equiv I(x2,y2) - I(x2,y1-1) - I(x1-1,y2) + I(x1-1,y1-1) \quad (1.3)$$

4	1	2	2
0	4	1	3
3	1	0	4
2	1	3	2

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

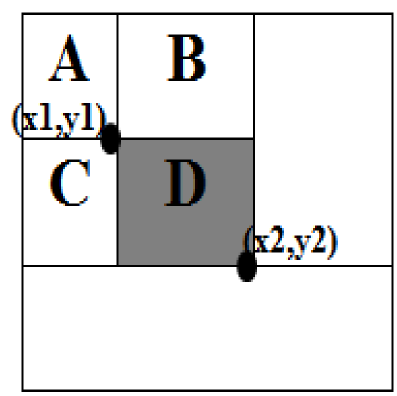


Figure 1.2: Compute of fundamental image**.**

- Simple contribution of image esteems.
- Registered vital image.
- By using the vital image to figure the entirety over square shape D.

Therefore instance: in light of straightforward image in Figure 1.2, discover entirety square shapes for (x1, y1) and (x2, y2) where x1=y1=1 and x2=y2=3 using Equation (1.4)

$$\sum_{x=1}^{3}\sum_{y=1}^{3} f\ (1,1) \equiv I(3,3) - I(3,0) - I(0,3) + (0,0) \tag{1.4}$$

The result is:

$$33 - 9 - 9 + 4 = 19$$

1.6 IMAGE BINARIZATION

Binarization alludes to the change of a Grayscale picture to paired, it is characterized into two classes: neighborhood and worldwide. The neighborhood techniques process an edge for each pixel by assessing locale around the pixel while Global threshold is characterized as strategy utilizing a special limit an incentive for all pixels in the picture by utilizing Equation (1.5)., the outcomes from this technique is inadmissible when distinctive light or shadows happen in the picture. [72]

$$Output\ \ Pixel = \begin{cases} white\ its\ grey\ level > T \\ black\ its\ grey\ level \leq T \end{cases} \tag{1.5}$$

Bradley technique is nearby threshold strategy for continuous application that is uses essential picture procedure. It fixes the unsuitable outcomes, which show up in Global threshold techniques, just as, Bradley technique strategy that is the straightforwardness execution.

That means utilized in Bradley technique are [13]:

- Necessary picture should be registered toward the start of first picture.
- Depend on necessary image; normal of square shape is registered for each pixel in the consistent times.
- When the pixel esteem is (t) percent not exactly average then estimation of pixel is equal to dark.
- When the pixel esteem is (t) percent more than or equivalent normal then estimation of pixel is equal to white.

1.7 FINGERPRINT

Skin on human fingertips contains edges and valleys which together structures particular examples. These examples are completely created under pregnancy and are changeless all through entire lifetime. Prints of those examples are called fingerprints. Wounds like cuts, consumes and wounds can briefly harm nature of fingerprints however when completely mended examples will be re-established. Through different examinations it has been seen that not two people have the equivalent fingerprints, consequently they are one of a kind for each person. [1]

Previously mentioned properties that human fingerprints have, made them prevalent as biometrics estimations. Particularly in law implementation where they have been utilized over a hundred years to help tackle wrongdoing. Tragically fingerprint coordinating is an unpredictable example acknowledgment issue. Manual fingerprint coordinating isn't just tedious however instruction and preparing of specialists takes quite a while. Hence since 1960s there have been done a part of effort on advancement of programmed fingerprint acknowledgment frameworks. Today there exist numerous deferent programmed frameworks with different calculations that take care of this example acknowledgment issue with excellent outcomes.

In any case, this doesn't implies that programmed fingerprint acknowledgment is a completely tackled issue. Structuring calculations that gives hearty answers for this specific issue is as yet a difficult and testing task.

Automatization of the fingerprint acknowledgment procedure ended up being accomplishment in criminological applications. Accomplishments made in legal zone extended the use of the programmed fingerprint acknowledgment into the regular citizen applications. Fingerprints have exceptional permanency and independence over the time. Likewise the numerous long periods of fulfilling knowledge in law requirement, utilizing the fingerprints as identification marks. The perceptions demonstrated that the fingerprints offer more secure and dependable individual identification than keys, passwords or id-cards can give. With diminishing expense of fingerprint per users and shabby expanding PC control, programmed fingerprint acknowledgment gives an efficient and reasonable option in contrast to conventional arrangements in person identification. [1, 62]

Precedents, for example, cell phones and PCs outfitted with fingerprint detecting gadgets for fingerprint based secret word assurance are being created to supplant conventional secret key insurance techniques. Those are just a small amount of regular citizen applications where fingerprints can be utilized.

There are numerous fingerprint coordinating strategies that can be connected for fingerprint acknowledgment, lamentably the constrained time don't permit examination of all. Another impediment is that this proposition takes just to thought alleged balanced coordinating strategy, generally likewise called verification. This implies the examination is performed just once and that is between format (the pre-put away fingerprint(s) on a sheltered spot in the framework) and fingerprint of the individual to confirm that his character is equivalent to the layouts. The strategy that is chosen for fingerprint coordinating was first found by Sir Francis Galton. In 1888 he saw that fingerprints are wealthy in subtleties likewise called details in type of discontinuities in edges. He additionally seen that situation of those particulars doesn't change over the time. Along these lines particulars coordinating are a decent method to set up if two fingerprints are from a similar individual or not. Most basic particulars that are found in fingerprints can be found in Figure 1.3 (on the left). [1, 52]

To show event of those in a genuine fingerprint investigate Figure 1.3 (on the right). The dim (dark) lines are the edges and light (white) lines are the valleys. After a closer examination there can be seen that the two most significant particulars are end and bifurcation, rest of the details are just mixes of

Those two types:

Thus, the fingerprint verification problem can be divided in two main tasks:

1. Minutiae extraction from fingerprints.
2. Minutiae matching.

Figure 1.3: Left: diff erent types of minutiae; right: real fingerprint.

1.8 BIOLOGICAL NEURON

A human mind comprises of roughly 1011 registering components called (neurons). They impart through an association system of axons and neurotransmitters having a thickness of around 104 neural connections for each neuron. Our theory in regards to the demonstrating of the common sensory system is that neurons speak with one another by methods for electrical motivations [16]. Figure 1.4 Illustration of an organic neuron. [54]

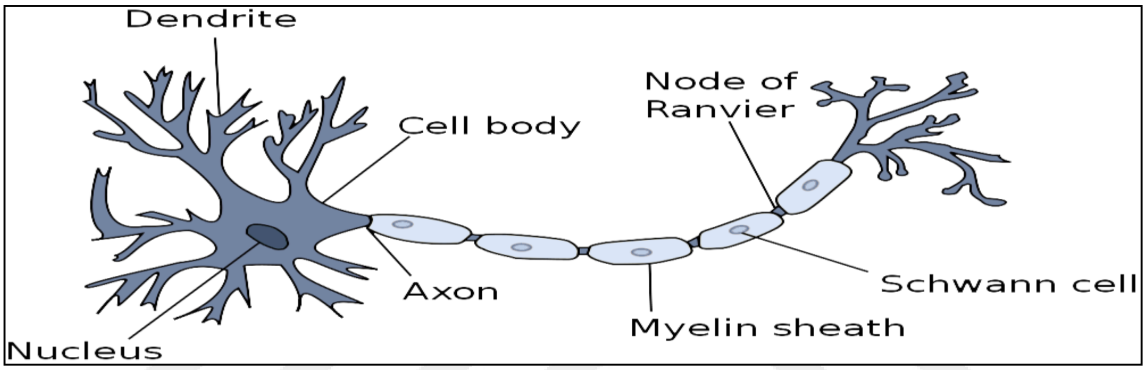


Figure 1.4: Illustration of a biological neuron.

1.9 ARTIFICIAL NEURAL NETWORK (ANN)

An Artificial Neural Network (ANN) is a scientific model that attempts to reenact the structure and functionalities of natural neural systems. Fundamental structure square of each counterfeit neural system is fake neuron, that is, a straightforward scientific model (work). Such a model has three straightforward arrangements of standards: increase, summation and enactment. At the passage of fake neuron the sources of info are weighted what implies that each information esteem is increased with individual weight. In the center area of counterfeit neuron is total Function that aggregates every weighted information and predisposition. At the exit of fake neuron the entirety of recently weighted sources of info and predisposition is going through enactment work that is additionally called exchange work as appeared in Figure 1.5. [53]

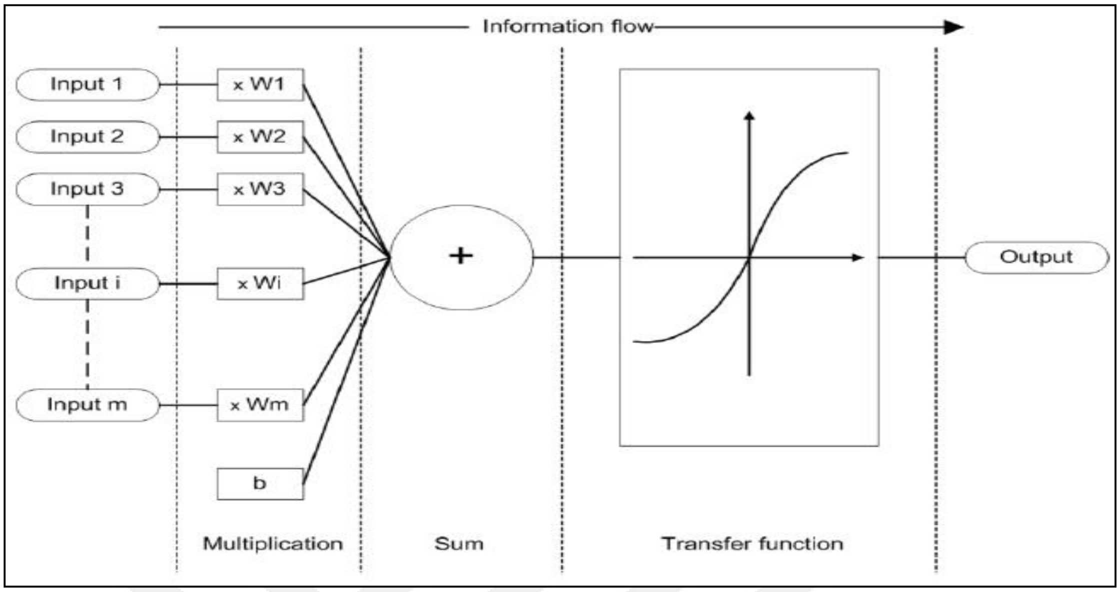


Figure 1.5: Working principle of an artificial neuron.

Similarities in design and functionalities can be seen in Figure 1.6. Where the left side of a figure represents a biological neuron with its soma, dendrites and axon and where the right side of a figure represents an artificial neuron with its inputs, weights, transfer function, bias and outputs. [53]

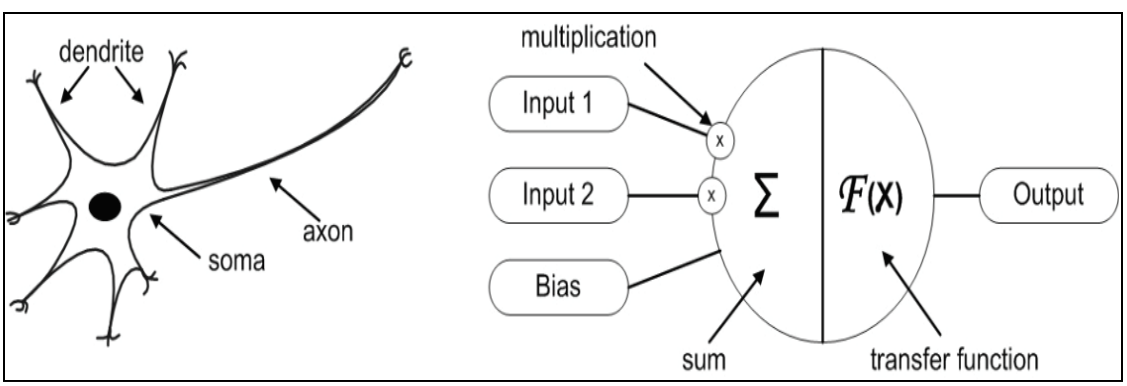


Figure 1.6: Biological and artificial neuron design.

1.10 ADAPTATION OR LEARNING RULE

The most striking contrast among NNs and conventional writing computer programs is that neural systems are in truth not customized by any stretch of the imagination, however can "learn" by model, along these lines extricating and summing up highlights of an introduced preparing set and relating them to the ideal yield. After a specific preparing period, the net ought to have the option to deliver the correct yield likewise for neither new information esteems, which are not part of the preparation set.

The preparation stage regularly continues as pursues: arbitrary qualities are at first relegated for the loads of the neurons. Examples from a preparation information record are then displayed to the system and the weightings are adjusted based on the learning principle and preparing design until an intermingling rule, for example a characterized blunder edge, is achieved. A test eliminate is then conveyed, in which obscure test examples are displayed to the system to build up the degree to which the system has gain proficiency with the assignment close by [43, 53]. The learning procedures can be partitioned in to:

(1) Supervised Learning

On account of administered learning, notwithstanding the info designs, the ideal comparing yield examples are additionally exhibited to the system in the preparation stage. The system figures a present yield from the info example, and this present yield is contrasted and the ideal yield. A mistake sign is gotten from the distinction between the produced and the required yield. This sign is then utilized to alter the loads as per the present learning rule, because of which the blunder sign is diminished. The best-known and most ordinarily utilized system model here is the multilayer perceptron with back proliferation learning rule.

Speculation, Supervised learning in NNs for the most part includes three informational indexes:

- A Training Set. Info examples utilized amid preparing.
- A Test Set. Info examples used to test the system in the wake of preparing is finished.
- A Validation Set. Info examples used to decide when to quit preparing to keep the system from ending up excessively explicit to the preparation information and accordingly not summing up well. This is accomplished by testing the system on the approval set each n cycles of preparing on the preparation information. Preparing is halted when the approval set blunder is at its minim. Figure 1.7 demonstrates the learning managed [53]

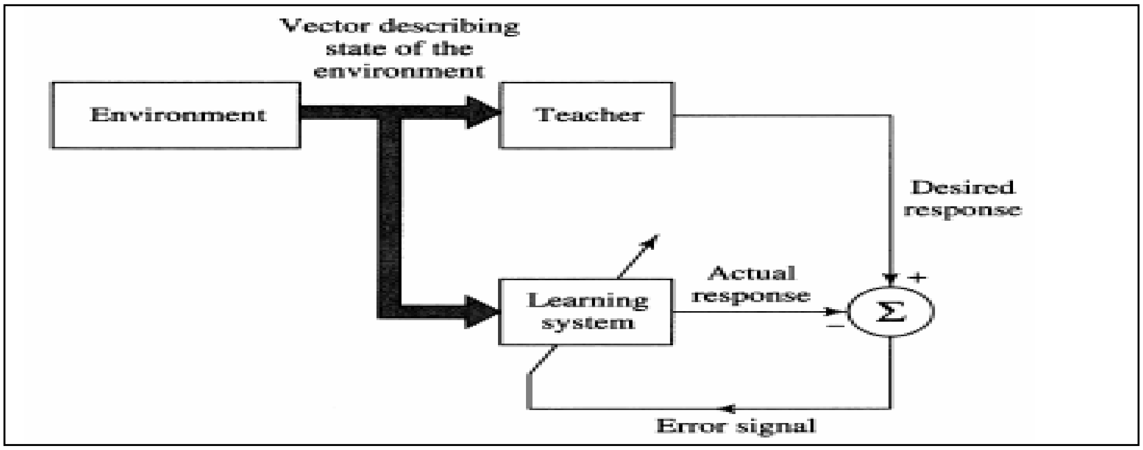


Figure 1.7: Supervised learning.

(2)Unsupervised (self-organizing)

On account of unsupervised learning, the system is required to discover grouping criteria for the information designs freely. The system endeavors to find normal highlights among the introduced information designs by means of a "closeness correlation", and to adjust its weight structure in like manner. The neurons hence from free example classes and become design finders. This technique is comparative basically to bunching calculations or vector evaluation strategies. Kohonen's gives a case of this procedure (self-sorting out component maps), which compose themselves with the point of changing over sign likeness in to vicinity between energized neurons.

Figure 1.8: demonstrate the learning unsupervised (self-arranging) [20, 11].

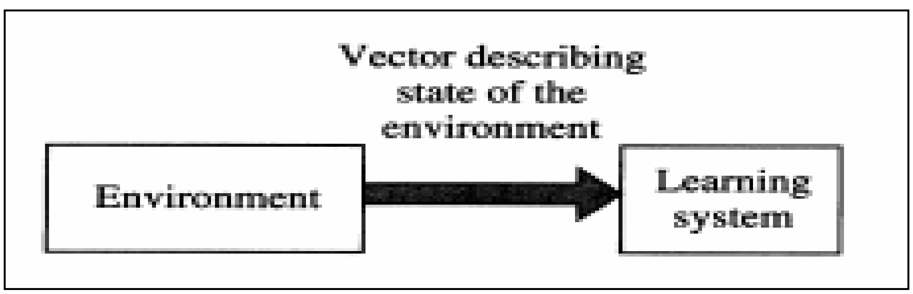
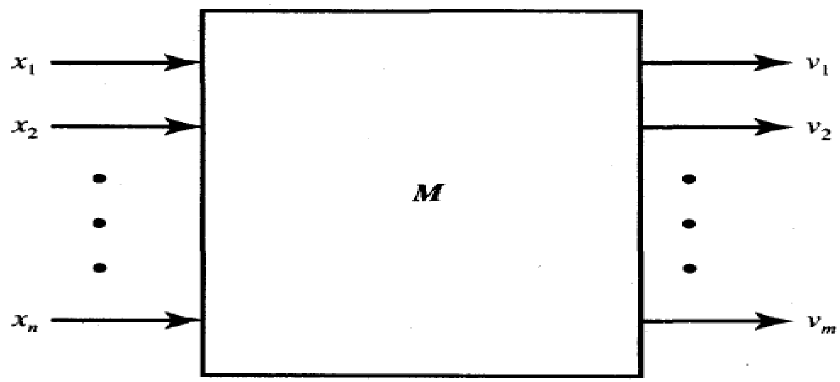


Figure 1.8: Unsupervised learning.

1.11 ASSOCIATIVE MEMORY

That is trusted the human memory is put away as intricate interconnections between different neurons. In counterfeit neural systems that assume the job of cooperative memory, information is by and large put away as a memory or weight lattice, which is utilized to produce the yield that ought to relate to a given information. The way toward building up the weight grid is alluded to as learning or putting away the ideal examples, while recovery or review alludes to the age of a yield design when an information example is displayed to the system [53].

Figure 1.9 illustrated general block diagram of associative memory performing associative mapping of vector xas the input to vector vas the output (See Equation (1.6)).



$$\mathbf{x}=[x_1\ x_2\ \dots\ x_n]^t$$
$$\mathbf{v}=[v_1\ v_2\ \dots\ v_m]^t$$

Figure 1.9: Block diagram for AM.

$$V=M[X] \qquad (1.6)$$

The framework shows map vectors xinto vectors v**,** in the example space and yield space, individually, by change administrator M signifies general nonlinear grid type administrator, and it has distinctive importance for every one of the memory models. Its structure, truth be told, characterizes a particular model that should be painstakingly laid out for every sort of memory. Structure of M mirrors particular neural memory worldview. For dynamic recollections, in additionally includes time variable. In this way, v is accessible at memory yield sometime in the not too distant future than the information has been connected. [54]

Accept that a lot of examples can be put away in the system. Afterward, if the system is given an example like an individual from the put away set, it might connect the contribution with the nearest put away example. The procedure is called auto affiliation. Normally, a debased information example fills in as signal for recovery of its unique structure. This is outlined schematically in Figure (1.10 (a)). The figure demonstrates a mutilated square reviewing the square encoded. Relationship of information examples can likewise be put away in hetero-association variation. In hetero-associative preparing, the relationship between sets of examples is put away. This is schematically appeared in Figure (1.10 (b)). Square information example exhibited at the info results in the rhomboid at the yield. It very well may be surmised that the rhomboid and square comprise one sets of put away examples. A mutilated info example may likewise cause right hetero-association in yield as appeared dashed line [53].

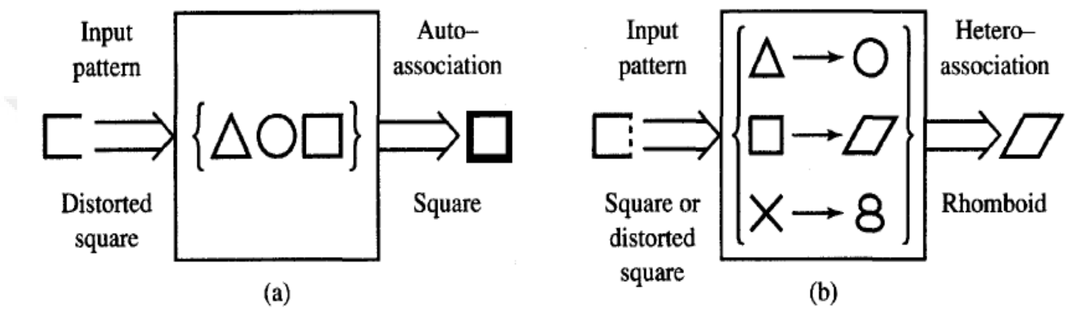


Figure 1.10: Association response: (a) auto association and (b) hetero association.

1.12 HOPFIELD NEURAL NETWORK

Physicist of the Hopfield depicted, Hopfield net it was chipping away at attractive conduct of solids (glass of turn) [58]. Around the same time, he expressed that I sing-turn, property for attractive particles, are depicted by two states (1 and – 1). Intriguing angle that is attractive common trade between iotas can be portrayed by numerical equation that at long last prompts Hopfield arrange [37, 38].

1.12.1 Architecture of Hopfield Neural Network

Hopfield net is an auto-acquainted framework, in that hub esteems are intuitively refreshed dependent on nearby calculation guideline. A new condition of every node based just on its net weight contribution at given time. [29, 57, 69]

As appeared in Figure 1.11, the Hopfield neural system has n nodes. Every node is associated with another node, yet not to itself. Qualities or loads of association are symmetric in which weight from node I to node jare equivalent to that of the node j to node I. That is, wij=wji and wii=0 for all I and j. also, edges are altogether thought to be zero. It is striking that progression of data in that kind of system isn't in solitary bearing, since it is workable for sign to spill out of a node back to itself by means of different nodes. Along these lines, one may state that there is criticism or repeat in the system since nodes might be utilized over and over to process data. Like system can be stood out from feed forward systems which have been utilized only to date Information (x1,x2,… , xn) and the yield (y1,y2,… ,yn) of this neural system take on esteem 1 and -1. [29, 30, 58]

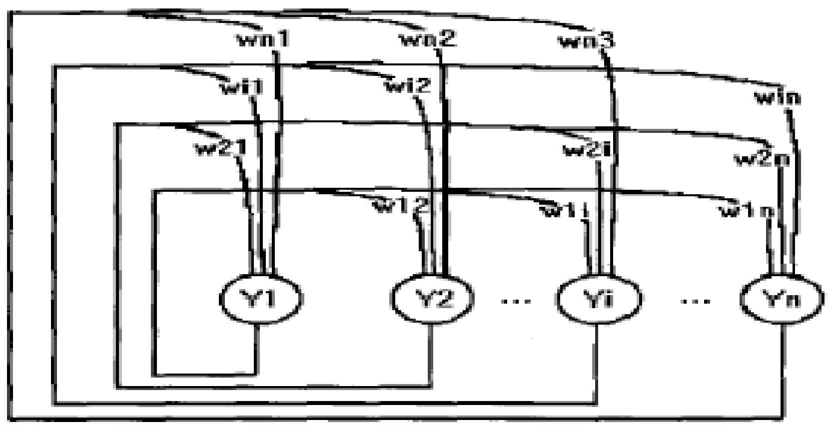


Figure 1.11:architecture Hopfield net Single layer n-node.

1.12.2 Algorithm of Hopfield Network

In two phases Hopfield network algorithm works i.e., learning and convergence phases, as illustrates in Algorithm (1.1:Algorithm of HNN). Create symmetric weights ($t_{ij} = t_{ji}$) for training patterns in learning phase, that is not change. Create associative weights for every training patterns, if there is more than one training pattern ($s_1, s_2, ..., s_n$), in additional operation to every symmetric weight of every training patterns ($t^1, t^2, ..., t^n$) will be performed. [29]

In the wake of introducing the Hopfield connect with an obscure info design, the assembly stage begins, and such an activity is rehashed until there is no adjustment in the Hopfield system yield all through the progressive cycles. At that point, the procedure is ceased, and the recuperated example is coordinated with the put away examples, and is allocated a class [57, 65].

Hopfield Neural Network
Input: Pattern x with length n. Output: Converge pattern μ.
Step 1:Assign connection weights: $$t_{ij} = \begin{cases} \sum_{s=0}^{M-1} x_i^s x_j^s & i \neq j \\ 0 & i = j, 0 \leq i, j \leq n-1 \end{cases}$$ where: M: is the number of patterns. n: is the number of element in the input vector. t_{ij} : is the connection weight from node i to node j. x_i : is an element i of a pattern s, which can be +1 or -1. s: is the training pattern (the pattern that it will be saved). Step 2: Initializes the unknown n input patterns: $$\mu(0) = x_i, \quad 0 \leq i \leq n-1$$ where: $\mu(t)$: is the output of node i at time t. Step 3: Iterates until convergence: $$\mu(t+1) = f_k\left[\sum_{i=0}^{n-1} t_{ij}\mu(t)\right] \quad 0 \leq i \leq n-1$$ where: f_k : is the hard limiting. Step 4: Repeats (for another unknown pattern) by going to step 2.

1.12.3 Applications of Hopfield Network

Hopfield net is typically utilized with parallel information sources. That net is more suitable when accurate double portrayals for examples is conceivable, e.g., highly contrasting pictures where the info components is pixel esteems, or ASCII content where information esteem are 8-bit ASCII portrayal of every characters. Therefore, that net is less suitable when info esteems are constant. That is on the grounds that a principal portrayal issue must be routed to change over the analogical amounts to twofold qualities. [50, 57]

An acquainted memory may be utilized, as substance addressable memory in various applications. At the point while few pieces of information are lost, it will finish contribution by finding lost part. As appeared in Algorithm (1.1), 120 node system was prepared utilizing 8 models appeared (A). For example digit "3" was undermined using haphazardly turning around every piece with likelihood of 0.25. At that point, that example was connected to system at time 0. Yields at time 0 and after the initial 7 cycles are appeared (B). [57, 65]

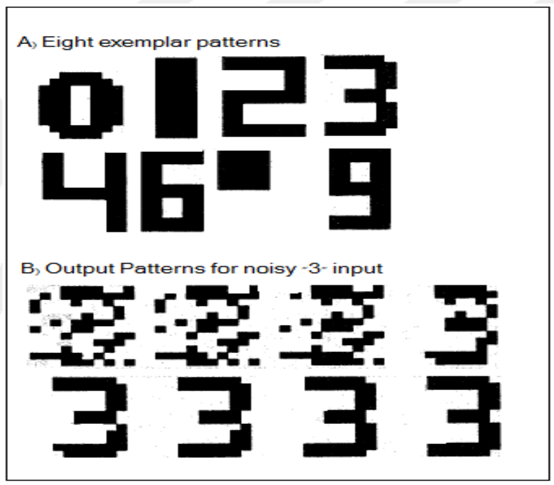


Figure 1.12: example of behaviour of a Hopfield net by using content-addressable memory.

In addition, the Hopfield net has been utilized to decide if information vector is "known" or is "unknown" vectors. The system perceives a "known" vector by delivering an example of enactments on the nodes of system that is like those put away by vector in system. On off chance that information "unknown" vector, the actuation vector delivered as system emphasizes will merge to enactment vector that isn't one of put away examples. Such an example is alluded to false steady state. [29]

In numerous applications, for example, mechanical vision where very quick acknowledgment of muddled pictures is required e.g., exceedingly twisted pictures must be recognized precisely, the cooperative memory neural systems are valuable for example acknowledgment and for partner the perceived examples in such undertakings. [65]

1.13 LYAPUNOV FUNCTION (ENERGY FUNCTION)

To demonstrate that iterated assembly procedures are steady, one ought to consider utilization of vitality work. [22, 50]. The last assumes an indispensable job in the credibility of the entire procedure.

The presence of a vitality capacity empowers one to demonstrate that the system will meet to a steady arrangement of initiations, instead of, continue swaying. This capacity diminishes as the conditions of the framework change. Such a capacity should be found and looked as the system task proceeds starting with one cycle then onto the next. The least mean square blunder is a case of such a capacity. Vitality work use guarantees the soundness of the framework that can't happen without combination. It is helpful to have one estimation of the vitality capacity to determine the framework conduct. In the Hopfield organize, a vitality capacity is the capacity that should be determined at whatever point the condition of any unit changes, i.e., this capacity dependably diminishes bit by bit to achieve a base and after that stops when the system is steady. As expressed in Hopfield [38], the vitality capacity is:

$$E = \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{i=1}^{n} v_i v_j w_{ij} + \sum_{k=1}^{n} \delta_k v_k \right) \tag{1.7}$$

Where

E = energy artificial network.

n = elements number.

Wij = weight from output of node it0 input of node j.

δ = limiting value, that equals to 0 in Hopfield net.

Vitality capacity is consequence of entirety of yield results of various nodes with association weight between them. Sets of nodes yields are increased in each term. The vitality capacity can be determined for each info vector made in system. In event that one figures vitality work for all the conceivable information vectors, one gets a vitality scene with greatest and least focuses. As indicated by [56], the base purposes of the vitality capacity are the examples of a Hopfield organize.

1.14 MCA ASSOCIATIVE MEMORY

MCA can describe as associative memory, and it is single-layer neural network. It modified from Hopfield network through modifying the architecture of the network, learning phase and convergence phase. MCA used to improve the efficiency of the Hopfield by avoiding most limitations of the Hopfield. [18, 20]

The MCA associative memory carried out in two phases: first phase is learning and the second phase is a convergence. In additional, it is dependent on two principles: that is, use smallest size of the network (fixed size of the network) rather than based on the whole size of the pattern through splitting the pattern into vectors with length equal to 3. While other principle, the MCA will avoid learning the same part of the pattern (which represents the vectors) many times. However, the bipolar representation used to represent the elements of the vectors; will be either 1 or -1. By using the bipolar representation, the data reduced and the important information in the image will kept. [18, 20]

Nevertheless, just like Hopfield networks, each neuron is connected to all neurons but not too itself. The weight for each vector represented by connections between neurons. Size of these vectors is three. Thus, there are no more than eight states for these vectors as 000, 001,…, and 111. Therefore, the learning process will create four weights matrices as a maximum (the size of each weight matrix will be 3*3) based on fact that every pair of orthogonal vectors have same weight. These weights matrices are symmetric, therefore that weight from node j into node i is same as that from node i into node j; that is, $w_{ji}= w_{ij}$ and $w_{ii}= 0$. Suppose the threshold for this network structure is zero [19, 20]. (See Figure 1.13).

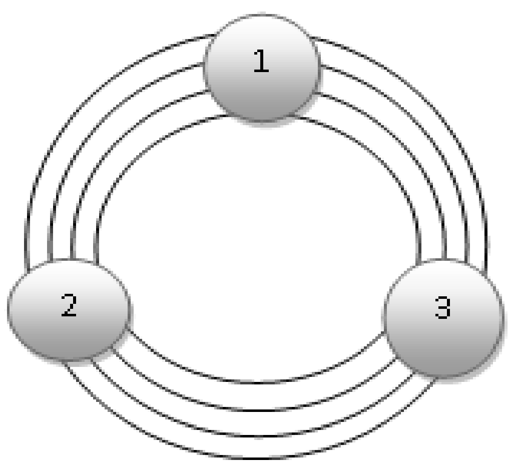


Figure 1.13: The Multi Connect Architecture (MCA).

1.14.1 Learning Phase

This section explained the MCA learning phase algorithm, which illustrated in Algorithm (1.2:The MCA learning algorithm). The whole required bipolar vectors of the pattern would be recognize during MCA associative memory by implantation the learning phase. Thus, input for learning phase is series for training patterns for the MCA. While the output results for learning phase after division, the training patterns to vectors with length three will be, save in a lookup table. These results consist two parts: that are, a set of weights that indicate to the training patterns vectors, these weights are Save Vector Weights (svw). And the next part is majority description (md) for every training pattern vectors, which are Save Majority Description (smd). [17, 20].

Input: training patterns p.
Output: lookup table for all n corresponding stored patterns.

Step_1: Initialize the four connection weights matrices.

$$w_0 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad w_1 = \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}, \quad w_2 = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}, \quad w_3 = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}$$

Step_2: Repeat step 2.1 to the end of training pattern p:

Step_2.1: Divide the training pattern p to n vectors v with length three.

Step_2.2: For each vector v, repeat steps 2.2.1, 2.2.2 and 2.2.3:

Step_2.2.1: Assign the vector majority description smd as follows:

$$md(v) = \sum_{i=1}^{3} v_i$$

$$smd = \text{hard} \quad \text{limiter}(md(v)) \begin{cases} 1 & md(v) \geq 1 \\ -1 & md(v) < 1 \end{cases}$$

Where:
md: majority description
smd: save majority description

Step_2.2.2: Assign the stored vector's weight svw as follow:

$$svw = f(Dcode(v)) \begin{cases} 0 \ \ or \ \ 7 & 0 \quad \{means\, w0 \\ 1 \ \ or \ \ 6 & 1 \quad \{means\, w1 \\ 2 \ \ or \ \ 5 & 2 \quad \{means\, w2 \\ 3 \ \ or \ \ 4 & 3 \quad \{means\, w3 \end{cases}$$

Where: Decode is a function to convert the binary number to decimal number.

Step_2.2.3: Save smd and svw for this vector in the lookup table.

Step_3: End.

The algorithm of learning phase illustrated that input for learning phase is serial of patterns to consider as training patterns for MCA. Depend on new architecture foe MCA, algorithm of learning begin with pattern, which split to n vectors, called vas illustrated in Figure 1.14. [17]

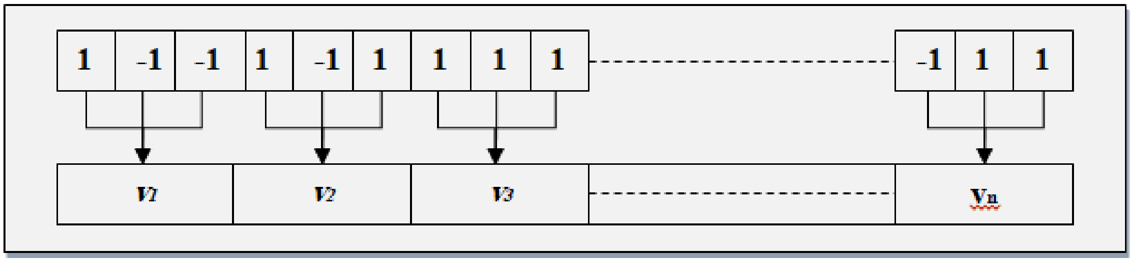


Figure 1.14: A training pattern split to n vectors with 3 elements.

Figure 1.14 illustrated how training pattern splitting in n vectors, each vector with length 3. Because that the elements of these vectors are bipolar, thus for these vectors no more than eight possibilities as illustrated in Table 1.1. [17]

Table 1.1: States of possible vector.

State No.	Bipolar vector state		
0	-1	-1	-1
1	-1	-1	1
2	-1	1	-1
3	-1	1	1
4	1	-1	-1
5	1	-1	1
6	1	1	-1
7	1	1	1

The 8 possible vectors with bipolar representation and with corresponding state number as illustrated in Table 1.1. The process of splitting training pattern to set vectors with length 3 is to accommodate MCA AM structure and these vectors have three elements, and these three elements represent a neuron in MCA structure .The weights matrices for the vectors represented by the connection between neurons based on fact that total numbers of vectors are 8 and bipolar representation of half of these vectors are orthogonal to other half, then numbers of weight matrices will be 4 because same weight matrix for each 2 orthogonal vectors. [17, 19]

Table 1.2 illustrated all possible vectors with bipolar representation and with corresponding state number. These weights computing using Equation (1.8). [20]

$$t_{ij} = \begin{cases} \sum_{s=0}^{M-1} x_i^s x_j^s & i \neq j \end{cases} \tag{1.8}$$

Where

M: number of patterns.

t_{ij} : Connection weights.

x_i : Element i of pattern s.

S: training pattern.

Table 1.2: Bossible vectors with corresponding weight matrices.

Orthogonal vectors

Binary representation of the corresponding bipolar vectors	Bipolar vector state	Wight
0 0 0	-1 -1 -1	$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$
0 0 1	-1 -1 1	$\begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}$
0 1 0	-1 1 -1	$\begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}$
0 1 1	-1 1 1	$\begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}$
1 0 0	1 -1 -1	$\begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}$
1 0 1	1 -1 1	$\begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}$
1 1 0	1 1 -1	$\begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}$
1 1 1	1 1 1	$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

In addition, the same weights for each orthogonal pair vectors, as illustrated in Table 1.3. The phenomenon of orthogonal leads to 4 weights matrices as illustrated Table 1.4. [17]

Table 1.3: Bipolar vectors with corresponding number of decimal and with corresponding svw.

Bipolar representation for vectors v	State No. for the vectors v	Corresponding weight for the vectors v
-1 -1- 1 or 1 1 1	0 or 7	W0
-1 -1 1 or 1 1 -1	1 or 6	W1
-1 1- 1 or 1- 1 1	2 or 5	W2
-1 1 1 or 1 -1 -1	3 or 4	W3

Table 1.4 Weight matrices for MCA AM

$$w_0=\begin{bmatrix}0 & 1 & 1\\ 1 & 0 & 1\\ 1 & 1 & 0\end{bmatrix},\quad w_1=\begin{bmatrix}0 & -1 & -1\\ -1 & 0 & 1\\ -1 & 1 & 0\end{bmatrix},\quad w_2=\begin{bmatrix}0 & -1 & 1\\ -1 & 0 & -1\\ 1 & -1 & 0\end{bmatrix},\quad w_3=\begin{bmatrix}0 & 1 & -1\\ 1 & 0 & -1\\ -1 & -1 & 0\end{bmatrix}$$

In Table 1.4, significance of MCA originates from requirement for more than one association with speak to the loads of these vectors. In this manner, number of nodes is credited to number of components (i.e., 3) of every vector though the 4 associations are ascribed to number of loads comparing to these vectors (i.e., 4). These weights are constant and don't require any calculation to be determined during learning phase. Accordingly, this calculation registering by instating every one of the 4 loads frameworks, for example (w_(0,) w_1,w_2,w_3). [17].

In spite of the fact that the misuse of the symmetrical wonder has been valuable in lessening number of loads, it is important to locate an inventive component to recognize each symmetrical pair. The lion's share portrayal of every vector is an extra change of MCA that used to recognize the symmetrical sets. Equation(1.9) and Equation(1.10) used to check Majority Description (md) for vectors [17]:

$$md(v) = \sum_{i}^{3} v_i \tag{1.9}$$

$$smd(v) = \text{hard} \quad \text{limiter}(\ md(v)) = \begin{cases} 1 & md(v) \geq 1 \\ -1 & md(v) < 1 \end{cases} \tag{1.10}$$

For example, binary vectors of [1 1 -1] and [-1 -1 1] have the same weight because are orthogonal to each other, while the md for these vectors are different. Therefore, after executed Equations(1.9)and (1.10), md for vector [1 1 -1] is 1 while the md for vector [-1 -1 1] is -1 as illustrated in Table 1.5. [17]

Table 1.5: Vectors different there corresponding md.

Bipolar vector state			Vector Majority Description
-1	-1	-1	-1
-1	-1	1	-1
-1	1	-1	-1
-1	1	1	1
1	-1	-1	-1
1	-1	1	1
1	1	-1	1
1	1	1	1

1.14.2 Convergence Phase

Algorithm (1.3:Convergence algorithm for MCA) shows that the output of convergence phase is based on results of learning phase. [17]

Input: n of unknown patterns p.
Output: Convergence pattern CP

Step_1: Initialize the four connection weights matrices.

$$w_0 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad w_1 = \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}, \quad w2 = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}, \quad w_3 = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}$$

Step_2: Initialize the energy function matrix e:

$$e = \begin{bmatrix} -3 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & -3 \end{bmatrix}$$

Step_3: Repeat steps 3.1, 3.2 and 3.3 until the unknown pattern p is ended:

Step_3.1: Divide the unknown pattern p to n vectors v with length three.

Step_3.2: Assign the test vector's weight tvw for all vectors v of the test pattern as follows:

$$tvw = f(Dcode(v)) \begin{cases} 0 \;\; or \;\; 7 & 0 \quad \{means\, w0 \\ 1 \;\; or \;\; 6 & 1 \quad \{means\, w1 \\ 2 \;\; or \;\; 5 & 2 \quad \{means\, w2 \\ 3 \;\; or \;\; 4 & 3 \quad \{means\, w3 \end{cases}$$

Step_3.3: Sum up the energy function for all n vectors in the unknown pattern each with its corresponding vector in the stored patterns:

$$ep = \sum_{i=1}^{n} e[svw_i, tvw_i] \;\; * \;\; (\;\; smd \;\; * \;\; tmd \;\;)$$

Step_4: Determine the stored pattern number minp with the minimum energy function to converge the unknown pattern towards it:

$$minp = \min(ep)$$

Where the min function is to determine the minimum energy function in ep array.

Step_5: Repeat steps 5.1, 5.2 and 5.3 to build the final converge pattern cp:

Step_5.1: Assign tempcv for each n vector in the test unknown pattern p:

$$tempcv_i = v_i \times svw_{minp}$$

Step_5.2: Assign the result majority description rmd to each n tempcv vector.

$$md(tempcv^i) = \sum_{j=1}^{n} tempcv^i_j$$

$$rmd = \text{hard limiter}(md(tempcv_i)) \begin{cases} 1 & md(tempcv_i) \geq 1 \\ -1 & md(tempcv_i) \leq 0 \end{cases}$$

Step_5.3: Create the converge vector cv in the converge pattern cp:

$$cv_i = (smd_{minp} \times rmd) tempcv_i$$

Step_6: End.

Like the learning stage, every one of the four weights net (i.e., (w_0 w_1,w_2,w_3) are introduced, and the unknown pattern separated to n vectors with every three components. The adjustment procedure implants the expanding job of energy function in combination procedure. Energy function is function whereby at whatever point the condition of any unit changes, the function dependably diminishes continuously to achieve the base. At that point, it stops after every one of the vectors with three components in the unknown pattern are joined towards the put away pattern with minimum energy function, that is lead as far as possible of the convergence process. [17, 20].

Energy function in MCA used to connect unknown pattern and patterns which saved its information (for example smd and svw) in lookup table thought learning phase. The correlation process achieved by computing summation of considerable number of estimations of energy function between each pair of vectors in both unknown and saved patterns. As illustrated Equation (1.11). [17, 19]

$$E = -\tfrac{1}{2}\left(\sum_{i=1}^{n} \sum_{i=1}^{n} v_i v_j w_{ij} + \sum_{k=1}^{n} \delta_k v_k \right) \tag{1.11}$$

Where

E = Energy artificial net.

n = elements number.

Wij = weight from node I as output to node j as input

δ = limiting value, that equals 0.

Value of energy function based on vectors and its identical weights. Therefore, these vectors and weights are constant; it is possible to computing every values of energy function and assign them in 2-dimensional matrix as shown in Table 1.6. [17]

Table 1.6: Values of energy function identical to every vector.

State no.	7 or 0	6 or 1	5 or 2	4 or 3
7 or 0	-3	1	1	1
6 or 1	1	-3	1	1
5 or 2	1	1	-3	1

Energy function value are reaches minimum value, that is -3, as illustrated in Table 1.6, when complete matching between test vector and saved vector; else, is 1. Identical energy functions in matrix of energy function is determined based on svw of saved Pattern in lookup table and on weight of test vector tvw from unknown pattern, that uses Equation (1.12)**.** [17]

$$\text{tvw} = f\ (\text{Dcode}\ (\text{v})\ ) \left| \begin{array}{llllll} 0 & or & 7 & 0 & \{means & w0 \\ 1 & or & 6 & 1 & \{means & w1 \\ 2 & or & 5 & 2 & \{means & w2 \\ 3 & or & 4 & 3 & \{means & w3 \end{array} \right. \tag{1.12}$$

by using Equation (1.13) are summed all n vectors of energy functions. [19]

$$ep = \sum_{i=1}^{n} e[svw_i, tvw_i] * (smd * tmd) \tag{1.13}$$

Then, by using converge algorithm assign minimum value of energy function summation to all saved pattern by chose closer to unknown pattern, this operations implemented by using Equation (1.14). [17]

$$cv_i = (smd_{minp} \times rmd) tempcv_i \tag{1.14}$$

It is to note that the Equation (1.14) used different type of the MD, called smd and rmd for saved and result vectors sequentially, that is computing by Equations (1.9) and (1.10). Each smd and rmd plays important role to retrieve the right vector. [17]

2. RELATED WORKS

This chapter provides a survey of the literature related to Fingerprint recognition system. That was developed to improve Fingerprint recognition system.

Hassan Abdel Qader et al. [31] proposed a fingerprint-matching approach that localises the matching regions in fingerprint images. The region of interest was determined by using core-points-related information and a novel feature vector extracted by Zernike moment invariant from each fingerprint image. Zernike moments are used as feature extractors due to their robustness to the noise, geometrical invariant properties and orthogonal properties in images.

Ruxin Wang et al. [61] propose a novel method using two classification probabilities for fuzzy classification which can effectively enhance the accuracy of classification. By only adjusting the probability threshold, we get the accuracy of classification is 96.1% (setting threshold is 0.85), 97.2% (setting threshold is 0.90) and 98.0% (setting threshold is 0.95). Using the fuzzy method, we obtain higher accuracy than other methods.

Sozan Abdulla Mahmood [66] The main objective in this paper, the geometrical shapes are used to extract features based on minutiae point. The features have been used as a set of descriptors for the fingerprint data and it is too complex to reconstruct the original fingerprint image using the extracted data. Back propagation Neural Networks and Support Vector Machine were used for Classification and recognition in the purposed system. It is found that the process of preprocessing steps necessary for accurate minutiae extraction. Also the method of constructing geometric shapes has great effects on producing good results in the recognition rate. The recognition rate of the Neural Network is more accurate than Support Vector Machine.

S. R. Patil (Waghjale) and S. R. Suralkar [62] proposed a new fingerprint classification approach based on artificial neural networks (ANN) and individual features, such as singular points. This approach is robust, reliable and able to address problems in rotation and translation. The classification efficiency of this approach was improved by using the back propagation algorithm because an input fingerprint image does not need to be compared with all registered fingerprint images.

Purneet Kaur and Jaspreet Kaur [52] enhanced their image classification results by combining the genetic algorithm with the neural network, which are currently the best available techniques for image classification. Specifically, they used the genetic algorithm to extract minutiae and used the neural network to recognise fingerprints.

Stephane Kouamo [64] proposed a method that uses the neural network to authenticate people who want to use an automated fingerprint system for e-learning. This method applies the back propagation algorithm on a multilayer perceptron during the training step. To its advantages, this technique uses a hidden layer, which allows the network to make comparisons and calculate probabilities on a template that is invariant to translation and rotation.

Einas Almarghni Azzoubi et al. [15] proposed a fingerprint-matching algorithm based on the Euclidean distance (ED) between two finger codes. This algorithm can be personalised according to the values of ED and threshold (TH). If the ED is less than the TH or equal to zero, then the two fingerprint images are deemed to have come from the same person.

Gu J, Zhou J, Yang C, et al. [28] proposes a novel portrayal for fingerprints which incorporates both particulars and model-based direction fields. At that point, unique mark coordinating should be possible by joining the choices of the matchers dependent on the worldwide structure (direction field) and the nearby sign (particulars). We have directed a lot of analyses on enormous databases and made exhaustive examinations with the condition of expressions of the human experience frameworks. Broad exploratory outcomes demonstrate that joining such neighborhood and worldwide discriminative data can to a great extent improve execution. The proposed framework is more powerful and exact than customary details based techniques, other than being superior to the past works which certainly fuse the direction data.

Engelsma J J, Arora S S, Jain A K, et al. [21] presented the design and manufacturing of high fidelity universal 3D fingerprint targets, which can be imaged on a variety of fingerprint sensing technologies, namely capacitive, contact-optical, and contactless-optical. In the first place, the universal 3D fingerprint targets enable not only a repeatable and controlled evaluation of fingerprint readers, but also enhance their ability to conduct fingerprint reader interoperability studies. Fingerprint reader interoperability refers to how robust the fingerprint recognition systems are to variations in the images acquired by different types of fingerprint readers.

Wang X F, Huang D S. [69] proposed another thickness put together bunching system based with respect to the presumption that the group focuses in information space are the objective items in a image space. In the first place, the dimension set advancement is received to discover a guess of group focuses by utilizing another underlying limit arrangement plot. Likewise, three kinds of introductory limits are characterized with the goal that every single one of them can advance to approach the group focuses in various ways. To stay away from the long cycle time of level set advancement in information space, an effective end model is introduced to stop the development procedure in situations having no more group focuses. At that point, another viable thickness portrayal, called level set thickness (LSD) is developed from the advancement results. At long last, the valley looking for bunching is utilized to assemble information focuses into relating groups dependent on the LSD. Examinations on some engineered and genuine informational collections have exhibited the productivity and adequacy of the proposed bunching system.

Abdelwahed Motwakel and Adnan Shaout [3], In this paper, present a novel procedure to examination unique mark image quality utilizing fluffy rationale. The nature of unique mark image extraordinarily influences the presentation of particulars extraction and the way toward coordinating in unique mark recognizable proof framework. The framework utilizes the separated four highlights from a unique finger impression image which are the Local Clarity Score (LCS), Global Clarity Score (GCS), Ridge_Valley Thickness Ratio (RVTR), and the complexity.

GajananPandurangKhetriand et al,. [26], In this paper, Fingerprint recognition is one of the accurate biometric recognition system and identification of Human fingerprint patterns using neural network. We optimize, feed forward back propagation neural network using the two different database like CASIA Fingerprint database and FVC2002 database for fingerprint pattern recognition.

Eduardo Zurek and et al., [14] presents an implementation of Artificial Neural Networks (ANN) for acoustic fingerprints recognition ,applied to the identification of marine vessels. In many cases, the vessel recognition process from an audible signal is performed by human operators, which could lead to failures in the identification process. Before entering the ANN classification process, the signal is filtered and its spectral characteristics are extracted. A comparison of the classification process between three types of neural networks is presented.

3. FUNDAMETAL OFMODFIY MULTI-CONNECT ARCHITYUCTURE ASSOCATIVE MEMORY

3.1 INTRODUCTION

In this chapter, we will explain theoretical fundamentals of algorithms that are used in this thesis.

3.2 MMCA AM NETWORK [59, 60]

In general, the operations of the AM (learning and convergence phases) in fact are less complex than other NN(neural network). Therefore, the associative memory network illustrated in this thesis, in particular, the MCA. Choosing MCA associative memory because of its learning ability; this ability comes from association among a pair of patterns using of energy function.

Nevertheless, there are three reasons make the MCA effective for real-time pattern recognition. The first reason, the architecture of MCA that is constant with any size of pattern. While the next reason depends on its ability to obviate most of AM limitations for learning phase. This phase creates 4-weight matrices with size 3*3, in addition to its capacity. The third reason is a convergence phase that is able to deal with problems for auto-associative memory, like an inverse value of the pattern, local minimum and boundary of allowable percentage of noise ratio.

Therefore, MCA AM has the ability to deal with the top limitations and increment capability for noise strong as well as that learning phase accelerate and convergence processes adjust associative memory depends on MCA AM named Modify Multi Connect Architecture (MMCA). In next subsections, this modification will be explain, which are include architecture of MCA network, in addition, to learning phase and convergence phase.

3.3 MMCA ARCHITECTURE [59, 60]

The MMCA associative memory comprise of small architecture by using smallest size of network, a phases that may be done by implementing of small number by MMCA of nodes which makes the process of calculation for MMCA very fast and possible process in the real time. According to this basics, size of the MMCA net is yet fixed (but with 2 node) with a multiple connections between these 2 nodes. The number of connections between nodes fixed at 2; this feature help to dealing with small size of network regardless of all size of patterns as illustrated in Figure 3.1.

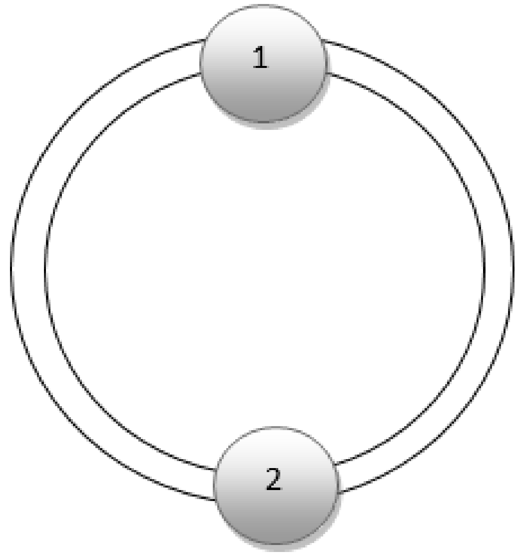


Figure 3.1: The MMCA AM.

Each neuron connected to the other neuron but not to itself as illustrated in Figure 3.1. These connections between the neuron represents the corresponding weight for each vectors. The possible number of vectors is 2^2, which that need to the two only of the weights. This is because the bipolar representation for half of these vectors will be orthogonal to other half and these orthogonal vectors have the same weights.

3.4 ALGORITHMS OF MMCA [59]

Therefore, MMCA used same Architecture of MCA, but number of node will be decreased to 2 instead of 3. The next subsections explain the learning phase and convergence phase algorithms.

3.4.1 Learning Phase

This phase is a significant processes that affecting on the performance of MMCA. Thus, in this thesis, propose modification for learning phase depends on MCA associative memory architecture and the learning phase will be create a symmetric weights ($t_{ji} = t_{ij}$) for all training pattern but without zero diagonal because the zero diagonal leads to oscillations. Algorithm (3.1:Learning phase of MMCA) illustrated learning phase.

Input: training patterns p.
Output: lookup table for all n corresponding stored patterns.

Step_1: Initialize the two connection weights matrices.

$$W_0 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad W_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Step_2: Repeat steps 2.1 and 2.2 to the end of training pattern p:

Step_2.1: **Divide the training pattern p to n vectors with length two.**

Step_2.2: For each vector v, repeat steps 2.2.1, 2.2.2 and 2.2.3:

Step_2.2.1: Assign the save majority description smd as follows:

$$md(v) = \begin{cases} v[1] & v[1] <> v[2] \\ \sum_{i=1}^{2} vi & otherwise \end{cases}$$

$$smd(v) = \text{hard} \quad \text{limiter}(md(v)) = \begin{cases} 1 & md(v) \geq 0 \\ -1 & md(v) < 0 \end{cases}$$

Step_2.2.2: Assign the save vector's weight svw as follow:

$$svw = f(Decode(v)) \begin{cases} 0 \quad or \quad 3 & 0 \ \{means \quad w0 \\ 1 \quad or \quad 2 & 1 \ \{means \quad w1 \end{cases}$$

Where: Decode is a function to convert the binary number to decimal number.

Step_2.2.3: Save smd and svw for this vector in the lookup table.

En

Algorithm (3.1) shows that inputs for the learning phase is a series of training patterns. The learning algorithm for the MMCA begins with pattern to splitting into n vectors called v, as illustrated in Figure 3.2.

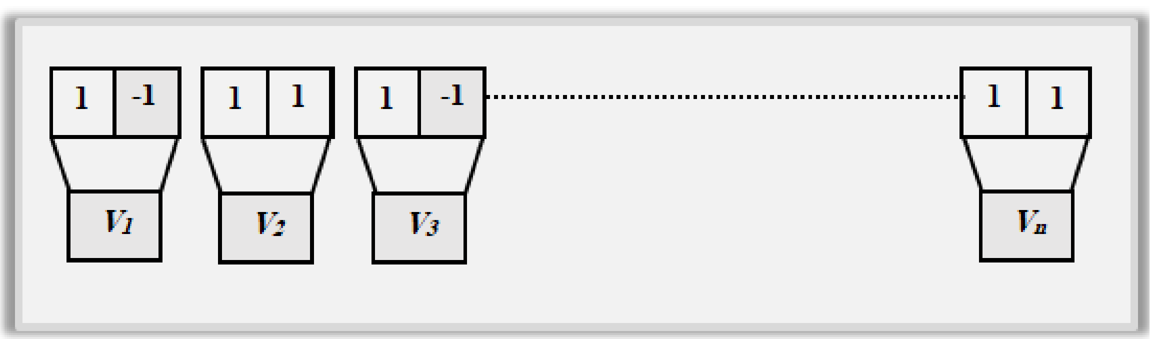


Figure 3.2: Splitting training pattern to n vectors.

In Figure 3.2 illustrates how to split training pattern to n vectors, each vector with length 2. There are only four possibilities for these vectors and represent these vectors with a bipolar values, as shown in Table 3.1.

Table 3.1: Four possible vectors.

State No.	Bipolar Vector State	
0	-1	-1
1	-1	1
2	1	-1
3	1	1

The 4 possible vectors with bipolar representation and with corresponding state number as illustrated in Table 3.1. Process of the splitting training pattern to many vectors with length 2 is to accommodate MMCA AM structure and these vectors have two elements, and these two elements represent a neuron in the MMCA structure. The weights matrices for the vectors represented by the connection between neurons. Based on fact that total numbers of all vectors are 4 and bipolar representation of semi of these vectors are orthogonal to other semi, then numbers of weight matrices will be only 2 because same weight matrix for each 2 orthogonal vectors.

4 possible vectors with theirs weight matrices as illustrated in Table 3.2.

Furthermore, the orthogonal phenomenon lead to just two weights matrices and these weight matrices are fixed and do not need any calculation to be computed through the learning phase. Therefore, the learning phase algorithm begins by initializing all 2 weights matrices (i.e., w_0 and w_1) as shown in Table 3.3 and in Table 3.4 shows the same weight for each orthogonal pair of vectors.

Table 3.2: 4 Bossible vectors with theirs corresponding weights matrices.

Binary representation of the corresponding bipolar vectors	**Bipolar vector state**	**Weights**
00	-1-1	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$
01	-1 1	$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$
10	1-1	$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$
11	11	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

Table 3.3: Weight matrices for MMCA associative memory.

$$w_0 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad w_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Table 3.4: 4 Bossible vectors with bipolar representation with theirs decimal number and corresponding weight (svw).

Bipolar representation for vectors v	State No. for the vectors v	Corresponding weight for the vectors v
-1 -1 OR 1 1	0 OR 3	W_0
-1 1 OR 1 -1	1 OR 2	W_1

The phenomenon of the orthogonal is useful in order to decrease number of weights. The majority description (MD) is additional modification for MCA AM. It was found to recognize between each orthogonal pair. This additional modification used in MMCA associative memory.

By using Equations (3.1) and (3.2) to calculate save majority description (svw) for MMCA.

$$md(v) = \begin{cases} v[1] & v[1] <> v[2] \\ \sum_{i=1}^{2} vi & otherwise \end{cases} \tag{3.1}$$

$$smd(v) = \text{hard} \quad \text{limiter}(md(v)) = \begin{cases} 1 & md(v) \geq 0 \\ -1 & md(v) < 0 \end{cases} \tag{3.2}$$

For example, the vectors v_1 = [1 -1] and v_2 = [-1 1] with bipolar representation, they have same weight because they are orthogonal for each other, but smd for each is various. Therefore, by using Equations (3.1) and(3.2) smd for v1 [1,-1] is 1 and v2 [-1,1] is a -1, as shown in Table 3.5.

Tablea 3.5: The vectors with their corresponding md.

Bipolar Vector State		Save Majority Description
-1	-1	-1
-1	1	-1
1	-1	1
1	1	1

Learning phase results will be saving in lookup table as shown in Figure 3.3.These results consist of a set of weights that indicate to the training patterns vectors these weights are svw, and the md for each the training pattern vectors that is smd, as shown in Figure 3.4. By using Equations (3.3) to calculate the svw. At the end, the accessing process for these results is a direct access, as will be explained in the convergence phase.

$$svw = f(Decode(v)) \begin{cases} 0 \quad or \quad 3 \quad 0 \ \{means \quad w0 \\ 1 \quad or \quad 2 \quad 1 \ \{means \quad w1 \end{cases} \tag{3.3}$$

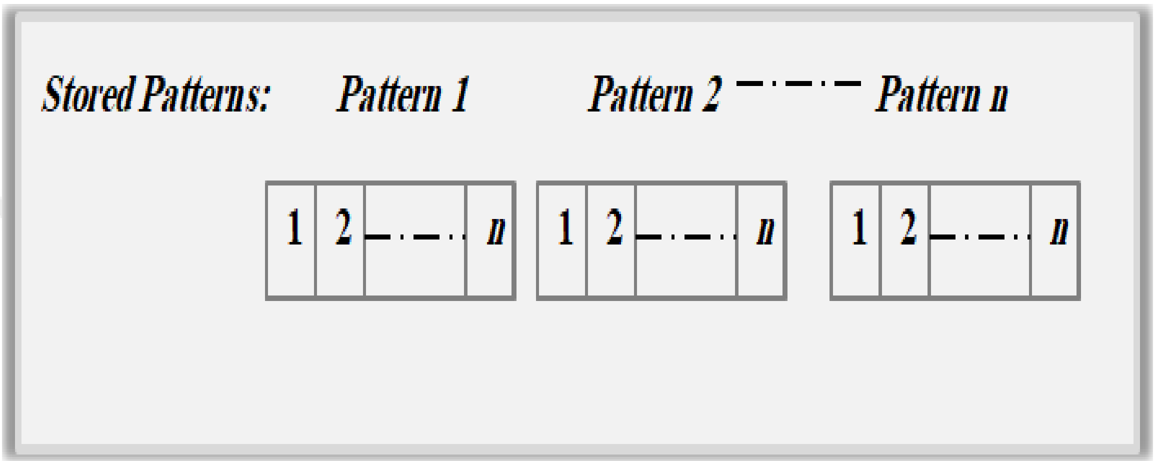


Figure 3.3: Stored patterns in lookup table**.**

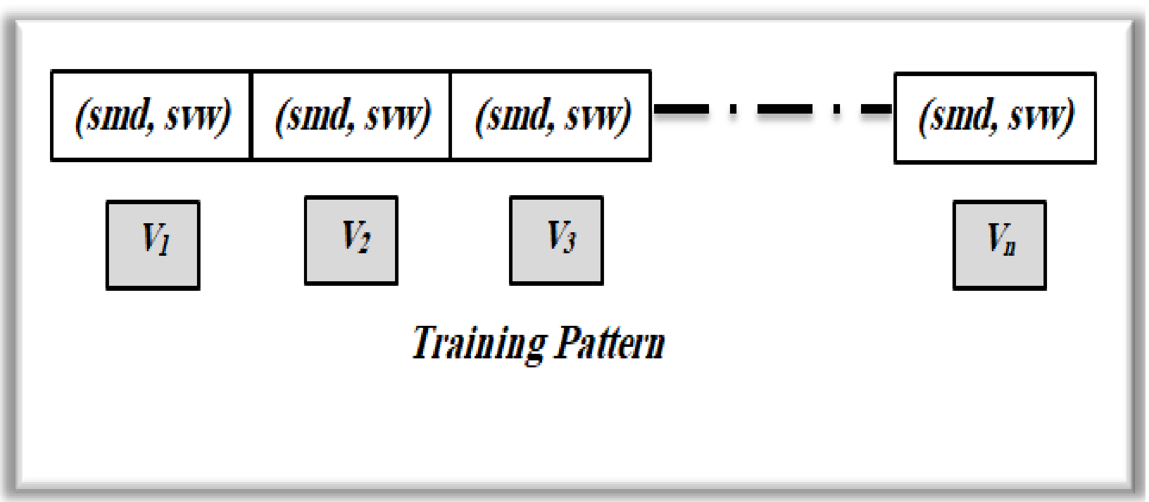


Figure 3.4: Lookup table of the training pattern with their corresponding array.

3.4.2 Convergence Phase

Algorithm (3.2:Algorithm for convergence phase MMCA) shows that the output of convergence phase is based on results of learning phase.

Input: n of unknown patterns p.
Output: Convergence pattern CP

Step_1: Initialize the two connection weights matrices.

$$w_0 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad w_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Step_2: Initialize the energy function matrix e:

$$e = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix}$$

Step_3: Repeat steps 3.1, 3.2 and 3.3 until the unknown pattern p is ended:

Step_3.1: Divide the unknown pattern p to n vectors with length two.

Step 3.2: For each vector v, repeat steps 3.2.1 and 3.2.2:

Step_3.2.1: Assign the test majority description tmd as follows:

$$md(v) = \begin{cases} v[1] & v[1] <> v[2] \\ \sum_{i=1}^{2} vi & otherwise \end{cases}$$

$$tmd = \text{hard} \quad \text{limiter}(md(v)) \begin{cases} 1 & md(v) \geq 0 \\ -1 & otherwise \end{cases}$$

Step_3.2.2: Assign the test vector's weight tvw as follow:

$$tvw = f(Decode(v)) \begin{cases} 0 \quad or \quad 3 \quad 0 & \{means \quad w0 \\ 1 \quad or \quad 2 \quad 1 & \{means \quad w1 \end{cases}$$

Step_3.3: Sum up the energy function for all n vectors in the unknown pattern each with its corresponding vector in the stored patterns:

$$ep = \sum_{i=1}^{n} e[svw_i, tvw_i] * (smd \quad * \quad tmd)$$

Step_4: Determine the stored pattern number minp with the minimum energy function to converge the unknown pattern towards it:

$$minp = \min(ep)$$

Where the min function is to determine the minimum energy function in ep array.

Step_5: Repeat steps 5.1, 5.2 and 5.3 to build the final converge pattern cp:

Step_5.1: Assign tempcv for each n vector in the test unknown pattern p:

$$tempcv_i = v_i \times svw_{minp}$$

Step_5.2: Assign the result majority description (rmd) to each n tempcv vector.

$$md(tempcv_i) = \begin{cases} v[1] & v[1] <> v[2] \\ \sum_{j=1}^{n} tempcv^i & otherwise \end{cases}$$

$$rmd = \text{hard} \quad \text{limiter}(md(tempcv_i)) \begin{cases} 1 & md \quad (tempcv_i) \geq 0 \\ -1 & otherwise \end{cases}$$

Step_5.3: Create the converge vector cv in the converge pattern cp:

$$cv_i = (smd_{minp} \times rmd) tempcv_i$$

End

As like to learning phase initialize both weight (i.e., w0 and w1), then splitting unknown patterns to n vectors with two elements.

These modifications ensure for the convergence process increments role energy function. The energy function computation by applying Equation (1.15). The output of energy function is based on vector and weight. These vectors and their corresponding weights are constant, it is potential compute all the possible values for energy function, as shown in Table 3.6. These values indexed in 2-dimensional matrix shape.

Table 3.6: Energy function values.

State no.	0 or 3	1 or 2
3 or 0	-2	0
2 or 1	0	-2

As illustrated in Table3.6, energy function values achieved minimum value is -2 when all vectors in the stored pattern are fully match with vectors in the unknown pattern, otherwise is 0.thevalues of matrix energy functions of energy function are chosen based on smd& svw of stored pattern with tmd and tvw of unknown pattern. By using Equations (3.1), (3.2) and (3.3) to calculate tmd and tvw.

In MMCA, the unknown pattern is correlate by using the energy function, and through learning phase, the data of patterns (i.e., smd and svw) are save in the lookup table. This Process achieved by computing the summation for all the energy function values among every pair of vectors in each of stored and unknown's patterns using Equation (1.17).

After compute the energy function, the convergence phase algorithm assigns minim value of energy function from all stored patterns which is latter nearest to unknown pattern and by using Equation (1.18) will be applied the convergence phase.

Two types of md, which used the Equation (1.18), the first called smd for stored vector and second called rmd for the result vector, computed by Equations (3.1) and (3.2). The value of smd and rmd play important role to restore right vector. At end, convergence phase of MMCA is not iterative process like convergence phase of MCA.

3.5 COMPLEXITY OF MMCA [16, 59]

O (n) is useful scale for the time of execution or memory to segment the algorithm. Figure 3.5 illustrates work of algorithm of MMCA.

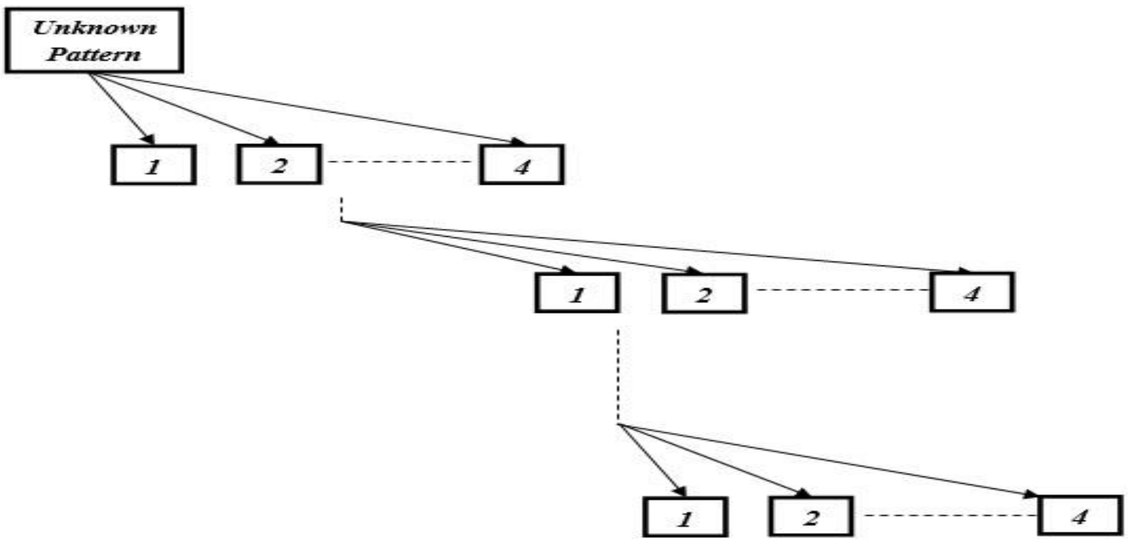


Figure 3.5: MMCA convergence of unknown' pattern towards ones off stored pattern.

For each level, compute the energy function among vectors of stored pattern and vectors of the unknown pattern. Like a computation dependent on both values of energy function until current level and energy function on this level. MMCA associative memory determines the convergence trend at the following level depending on the energy function value. The maximum numbers of branching is four because there are only four possible vectors, as illustrated in Table 3.3. According to Table 3.5, maximum length path of pattern with n elements is k, that is equal to n/2, (n/2 is equal to numbers of the vectors with 2 elements on pattern with n elements). Thus, the O (n) of this process is O(4× (n/2)). While, O(n) for MCA is O(8× (n/3)). Accordingly, it is clear that the complexity of MMCA is lower than that of MCA.

3.6 MMCA EXAMPLES [59]

This section presents two examples of the MMCA. All steps of learning process and convergence process of MMCA are illustrated in next examples.

Ex_1: Learning Two Patterns: Pattern1 and Pattern2

There are 2 patterns with 10 elements shown in Figure 3.6 must be convert into vectors by taken one by one from rows of patterns, begin of first row and end with last row. Therefore, these patterns are as follows:

Pattern 1 = [1,-1,-1,1,-1,1,1,-1,-1,-1,-1,-1,-1,1,1,-1,-1,1,1,-1];

Pattern 2 = [-1,-1,-1,-1,-1,1,1,1,-1,-1,-1,-1,-1,1,1,1,-1,-1,-1,-1].

Figure 3.6: Pattern1 and Pattern2.

Both patterns are learning by learning phase MMCA AM, which described by:

St_1: Two weights matrices are initialized (i.e. w_1 and w_2).

St_2: Repeat St_2, 1 and St_2, 2 for each training patterns.

St_2.1: divined training patterns into n vectors with length 2.

10 vectors for each pattern. The vectors for Pattern1 are [1,-1],[-1,1],[-1,1],[1,-1],[-1,-1],[-1,-1],[-1,1],[1,-1],[-1,1],[1,-1] while vectors for Pattern2 are: [-1,-1],[-1,-1],[-1,1],[1,1],[-1,-1],[-1, -1],[-1,1],[1,1],[-1,-1],[-1,-1]. Where the length of these vectors is twenty,

St_2, 2: For all vectors, repeat steps St_2,2,1 .2,2,2 and 2,2,3.

St_2,2,1: For each vectors, determine the smd. Compute the smd for Pattern 1 and Pattern 2 as shown in Tables (3.7) and (3.8).

Table 3.7: Computed smd for pattern1.

Pat1		
Vectors	***Majority Description***	***Save Majority Description***
Vec_1	***md [1 -1] = 1***	***Smd = 1***
Vec_2	***md [-1 1] = -1***	***Smd = -1***
Vec_3	***md [-1 1] = -1***	***Smd = -1***
Vec_4	***md [1 -1] = 1***	***Smd = 1***
Vec_5	***md [-1 -1] =-2***	***Smd = -1***
Vec_6	***md [-1 -1] =-2***	***Smd = -1***
Vec_7	***md [-1 1] = -1***	***Smd = -1***
Vec_8	***md [1 -1] = 1***	***Smd = 1***
Vec_9	***md [-1 1] = -1***	***Smd = -1***
Vec_10	***md [1 -1]***	***Smd = 1***

Table 3.8: Computed smd for pattern2.

Pat2		
Vectors	***Majority Description***	***Save Majority Description***
Vec_1	***md [-1 -1] =-2***	***Smd = -1***
Vec_2	***md [-1 -1] =-2***	***Smd = -1***
Vec_3	***md [-1 1] = -1***	***Smd = -1***
Vec_4	***md [1 1] = 2***	***Smd = 1***
Vec_5	***md [-1 -1] =-2***	***Smd = -1***
Vec_6	***md [-1 -1] =-2***	***Smd = -1***
Vec_7	***md [-1 1] = -1***	***Smd = -1***
Vec_8	***md [1 1] = 2***	***Smd = 1***
Vec_9	***md [-1 -1] =-2***	***Smd = -1***
Vec_10	***md [-1 -1] =-2***	***Smd = -1***

St_2,2,2: assign svw for each vectors, as shown in Tables (3.9) and (3.10) compute svw for Pattern 1 and Pattern 2.

Table 3.9: Computed svw for pattern1.

Pat1		
Vectors	***Decode Function***	***Save Vector Weight***
Vec_1, [1 -1]	***Decimal number = 2***	***svw = w1***
Vec_2, [-1 1]	***Decimal number = 1***	***svw = w1***
Vec_3, [-1 1]	***Decimal number = 1***	***svw = w1***
Vec_4, [1 -1]	***Decimal number = 2***	***svw = w1***
Vec_5, [-1 -1]	***Decimal number = 0***	***svw = w0***
Vec_6, [-1 -1]	***Decimal number = 0***	***svw = w0***
Vec_7, [-1 1]	***Decimal number = 1***	***svw = w1***
Vec_8, [1 -1]	***Decimal number = 2***	***svw = w1***
Vec_9, [-1 1]	***Decimal number = 1***	***svw = w1***
Vec_10, [1 -1]	***Decimal number = 2***	***svw = w1***

Table 3.10: Computed svw for pattern2.

Pat2		
Vectors	***Decode Function***	***Save Vector Weight***
Vec_1, [-1 -1]	***Decimal number = 0***	***svw = w0***
Vec_2, [-1 -1]	***Decimal number = 0***	***svw = w0***
Vec_3, [-1 1]	***Decimal number = 1***	***svw = w1***
Vec_4, [1 1]	***Decimal number = 3***	***svw = w0***
Vec_5, [-1 -1]	***Decimal number = 0***	***svw = w0***
Vec_6, [-1 -1]	***Decimal number = 0***	***svw = w0***
Vec_7, [-1 1]	***Decimal number = 1***	***svw = w1***
Vec_8, [1 1]	***Decimal number = 3***	***svw = w0***
Vec_9, [-1 -1]	***Decimal number = 3***	***svw = w0***
Vec_10, [-1 -1]	***Decimal number = 3***	***svw = w0***

St_2,2,3: For each vectors the value of smd and svw are store in the lookup table, as illustrated in Table 3.11.

Table 3.11: Store svw and smd for 2 patterns (Pattern1 and Pattern2) in lookup table of the MMCA.

Pattern Vectors	*Pat1*		*Pat2*	
	Smd	*Svw*	*Smd*	*Svw*
Vector 1	1	W_1	-1	W_0
Vector 2	-1	W_1	-1	W_0
Vector 3	-1	W_1	-1	W_1
Vector 4	1	W_1	1	W_0
Vector 5	-1	W_0	-1	W_0
Vector 6	-1	W_0	-1	W_0
Vector 7	-1	W_1	-1	W_1
Vector 8	1	W_1	1	W_0
Vector 9	-1	W_1	-1	W_0
Vector 10	1	W_1	-1	W_0

Store the value of smd&svw in lookup table for all vectors in the stored patterns as shown in Table (3.11). For all vectors, the smd value found by applying the Equations (3.1) and (3.2). Furthermore, for all vectors the svw found by applying St_2,2,2 in learning algorithm.

Ex_2: Unknown Pattern Convergence Phase

The patterns (Pattern1 and Pattern2) in the Ex_1 are MMCA stored patterns, and unknown pattern "Un-Pat" as illustrated in Figure 3.7.

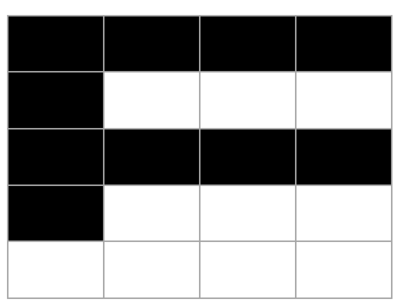


Figure 3.7: Un-Pat.

The unknown pattern illustrated in Figure3.7 should be converted to multiples vectors by taken one by one from rows of pattern, begin from beginning row to end row. Therefore, the unknown pattern will be:

Un-Pat = [-1, -1,- 1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1].

By using the convergence algorithm, the unknown pattern will be converged to one of the saved patterns, as follow:

St_1: Two weight matrices are initialized (such as, w_1, and w_2).

St_2:matrix of Energy function is initializing.

St_3: Repeat Steps St_3.1 and 3.2, for the Un-Pat.

St_3.1: split the Un-Pat into n vectors with length 2.

Unknown pattern(Un-Pat) has ten vectors, the ten vectors are [-1,-1],[-1,-1],[-1,1],[1,1],[-1,-1],[-1,-1],[-1,1],[1,1],[1,1],[1, 1].

St_3.2: For all vectors, repeat Steps St_3.2.1 and 3.2.2.

St_3.2.1: For each unknown vectors, designate the tmd value. Calculate value oftmd as shown in Table (3.12).

Table 3.12: Compute tmd for unknown pattern.

Unknown Pattern		
Vectors	***Majority Description***	***Test Majority Description***
Vec_1	***md [-1 -1] = -2***	***tmd = -1***
Vec_2	***md [-1 -1] = -2***	***tmd = -1***
Vec_3	***md [-1 1] = -1***	***tmd = -1***
Vec_4	***md [1 1] = 2***	***tmd = 1***
Vec_5	***md [-1 -1] = -2***	***tmd = -1***
Vec_6	***md [-1 -1] = -2***	***tmd = -1***
Vec_7	***md [-1 1] = -1***	***tmd = -1***
Vec_8	***md [1 1] = 2***	***tmd = 1***
Vec_9	***md [1 1] = 2***	***tmd = 1***
Vec_10	***md [1 1] = 2***	***tmd = 1***

St_3.2:2: For each vectors in Un-Pat, assign value of tvw, as shown in the Table (3.13).

Table 3.13: Computed tvw for Un-Pat.

Unknown Pattern		
Vectors	***Decode Function***	***Test Vector Weight***
Vec_1, [-1 -1]	***Decimal Number = 0***	***tvw = w0***
Vec_2, [-1 -1]:	***Decimal Number = 0***	***tvw = w0***
Vec_3, [-1 1]:	***Decimal Number = 1***	***tvw = w1***
Vec_4, [1 1]	***Decimal Number = 3***	***tvw = w0***
Vec_5, [-1 -1]	***Decimal Number = 0***	***tvw = w0***
Vec_6, [-1 -1]	***Decimal Number = 0***	***tvw = w0***
Vec_7, [-1 1]	***Decimal Number = 1***	***tvw = w1***
Vec_8, [1 1]	***Decimal Number = 3***	***tvw = w0***
Vec_9[1 1]	***Decimal Number = 3***	***tvw = w0***
Vec_10, [1 1]	***Decimal Number = 3***	***tvw = w0***

St_3.2.3: Summation energy function values for every vectors in stored patterns with identical vectors in the unknown pattern as shown in Table (3.14).

Tables 3.14: The energy function for each patterns (Pattern1 and Pattern2).

Vectors	*tvw for the test patterns vectors*	*svw for the stored patterns p1*	*svw for the stored patterns p2*	*Energy function for the test pattern with the stored pattern p1*	*Energy function for the test pattern with the stored pattern p2*
Vector 1	tvw=W_0=0	svw=W_1=1	svw=W_0=0	e[0,1]=0	e[0,0]=-2
Vector 2	tvw=W_0=0	svw=W_1=1	svw=W_0=0	e[0,1]=0	e[0,0]=-2
Vector 3	tvw=W_1=1	svw=W_1=1	svw=W_1=1	e[1,1]=-2	e[1,1]=-2
Vector 4	tvw=W_0=0	svw=W_1=1	svw=W_0=0	e[0,1]=0	e[0,0]=-2
Vector 5	tvw=W_0=0	svw=W_0=0	svw=W_0=0	e[0,0]=-2	e[0,0]=-2
Vector 6	tvw=W_0=0	svw=W_0=0	svw=W_0=0	e[0,0]=-2	e[0,0]=-2
Vector 7	tvw=W_1=1	svw=W_1=1	svw=W_1=1	e[1,1]=-2	e[1,1]=-2
Vector 8	tvw=$W_{0=}$0	svw=W_1=1	svw=W_0=0	e[0,1]=0	e[0,0]=-2
Vector 9	tvw=W_0=0	svw=W_1=1	svw=W_0=0	e[0,1]=0	e[0,0]=2
Vector 10	tvw=W_0=0	svw=W_1=1	svw=W_0=0	e[0,1]=0	e[0,0]=2
				Sum=-8	Sum=-12

Energy function value assigned for all vectors from matrix of the energy function based on values of smd, svw, tmd and tvw.

St_4: Select the minimum energy function to determine saved number pattern (minp) to converge unknown pattern. The unknown pattern is nearest to pattern2 than pattern1 as illustrated in Table (3.14). That means, the MMCA net has recognized correctly for pattern.

St_5: To build the last converged pattern (CP), should repeat the Steps St_5.1, 5.2 and 5.3

St_5.1: For each n vectors in Un-Pat, assign tempcv as shown in the Table (3.15).

Table 3.15: Assign tempcv for vectors in the unknown pattern Un-Pat.

Assign tempcv for each n vector in the unknown patt ι P		
Vectors	***tempcv***	***After applying the hard limiter***
Vec_1, [-1 -1]	$tempcv = \begin{bmatrix}-1\\-1\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} = \begin{bmatrix}-2 & -2\end{bmatrix}$	***[-1 -1]***
Vec_2, [-1 -1]	$tempcv = \begin{bmatrix}-1\\-1\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} = \begin{bmatrix}-2 & -2\end{bmatrix}$	***[-1 -1]***
Vec_3, [-1 1]	$tempcv = \begin{bmatrix}-1\\1\end{bmatrix}\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix} = \begin{bmatrix}-2 & 2\end{bmatrix}$	***[-1 -1]***
Vector4, [1 1]	$tempcv = \begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} = \begin{bmatrix}2 & 2\end{bmatrix}$	***[1 1]***
Vec_5, [-1 -1]	$tempcv = \begin{bmatrix}-1\\-1\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} = \begin{bmatrix}-2 & -2\end{bmatrix}$	***[-1 -1]***
Vec_6, [-1 -1]	$tempcv = \begin{bmatrix}-1\\-1\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} = \begin{bmatrix}-2 & -2\end{bmatrix}$	***[-1 -1]***
Vec_7, [-1 1]	$tempcv = \begin{bmatrix}-1\\1\end{bmatrix}\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix} = \begin{bmatrix}-2 & 2\end{bmatrix}$	***[-1 1]***
Vec_8, [1 1]	$tempcv = \begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} = \begin{bmatrix}2 & 2\end{bmatrix}$	***[1 1]***
Vec_9, [1 1]	$tempcv = \begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} = \begin{bmatrix}2 & 2\end{bmatrix}$	***[1 1]***
Vec_10, [1 1]	$tempcv = \begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} = \begin{bmatrix}2 & 2\end{bmatrix}$	***[1 1]***

St_5.2: For all tempcv vectors, designate value of rmd, as shown in Table (3.16).

Table 3.16: Compute rmd for each n tempcv vector.

Vectors	*Majority Description*	*Result Majority Description*
Vec_1	*md [-1 -1] =-2*	*rmd = -1*
Vec_2	*md [-1 -1] =-2*	*rmd = -1*
Vec_3	*md [-1 1] = -1*	*rmd =-1*
Vec_4	*md [1 1] = 2*	*rmd = 1*
Vec_5	*md [-1 -1] =-2*	*rmd = -1*
Vec_6	*md [-1 -1] =-2*	*rmd = -1*
Vec_7	*md [-1 1] = -1*	*rmd = -1*
Vec_8	*md [1 1] = 2*	*rmd = 1*
Vec_9	*md [1 1] = 2*	*rmd = 1*
Vec_10	*md [1 1] = 2*	*rmd = 1*

Step_5.3: Create converged pattern (CP) from converged vector (cv) as follows:

$$cv_1 = (-1 \times -1) \times [-1 \quad -1] = [-1 \quad -1]$$

$$cv_2 = (-1 \times -1) \times [-1 \quad -1] = [-1 \quad -1]$$

$$cv_3 = (-1 \times -1) \times [-1 \quad 1] = [-1 \quad 1]$$

$$cv_4 = (1 \times 1) \times [1 \quad 1] = [1 \quad 1]$$

$$cv_5 = (-1 \times -1) \times [-1 \quad -1] = [-1 \quad -1]$$

$$cv_6 = (-1 \times -1) \times [-1 \quad -1] = [-1 \quad -1]$$

$$cv_7 = (-1 \times -1) \times [-1 \quad 1] = [-1 \quad 1]$$

$$cv_8 = (1 \times 1) \times [1 \quad 1] = [1 \quad 1]$$

$$cv_9 = (-1 \times 1) \times [1 \quad 1] = [-1 \quad -1]$$

$$cv_{10} = (-1 \times 1) \times [1 \quad 1] = [-1 \quad -1]$$

Final Converged Pattern illustrated in Figure 3.8:

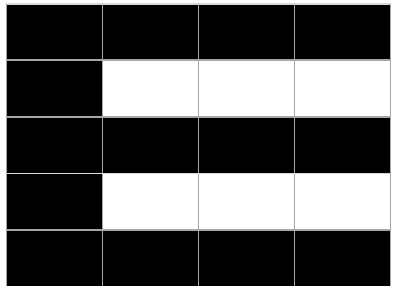


Figure 3.8: CP (Converged pattern).

The MMCA net converge to the pattern2 correctly because it is closest to Un-Pat than to the pattern1 as showed in example above. For clarity, the entire process illustrated in Figure 3.9.

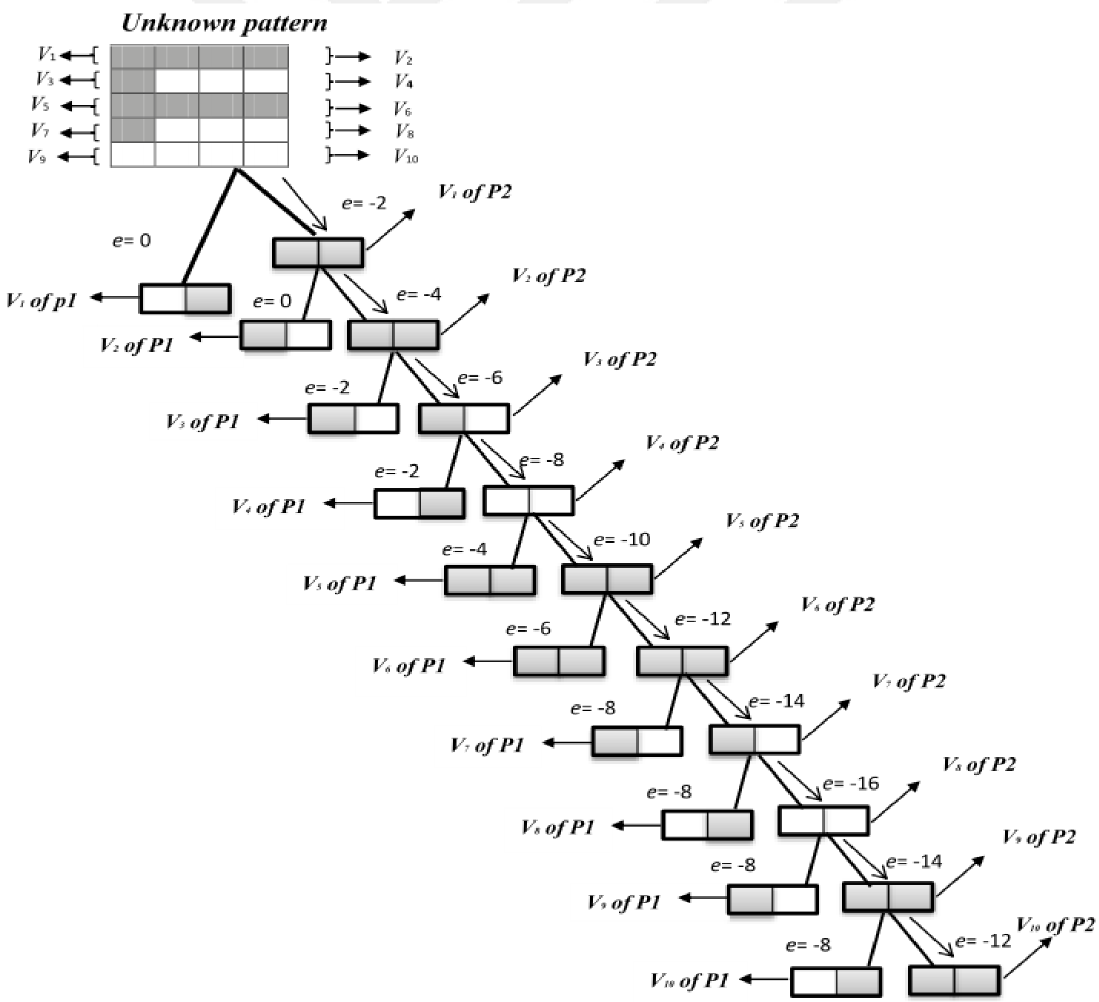


Figure 3.9: The convergence phase for Ex_2.

Figure 3.9 shows, that all levels are dedicated to all vectors; for example, level_1 is dedicated for vector_1, level_2 dedicated to vector_2,………, and level 10 dedicated to vector 10. Energy function for all levels between vectors of unknown pattern and vectors of the stored pattern is computing the next energy function value until current level. The MMCA network determines the convergence trend in the following level depending on the energy function value.

4. PROPOSED ALGORITHMS

4.1 INTRODUCTION

The methodology that have been use to improve the efficiency of Fingerprint system is explain in this chapter, which is use for recognize fingerprint. The fingerprint patterns are given as input to the computer after captured with different methods of capture. Technically, proposed fingerprint system has two components: it is preprocessing step and MMCA network, which work with two phases the training and the analysis phases. All details are shown with the next section and illustrated in Figure 4.1.

4.2 DESIGN SYSTEM PROPOSED

This work focus on a many of steps during procedure of planning the system proposed in order to achieve most of techniques and methods that contribute to resolve the problem of thesis. These steps are content: preprocessing step and recognition step (use MMCA network which work with 2 phases training and analysis phases).

Proposed system is based on following steps as it is explain in Figure 4.1:

- **The Preprocessing step:** several processes are performed to improve the quality of the fingerprint patterns**.**
- **The Recognition step:** MMCA AM authenticates these fingerprint patterns.

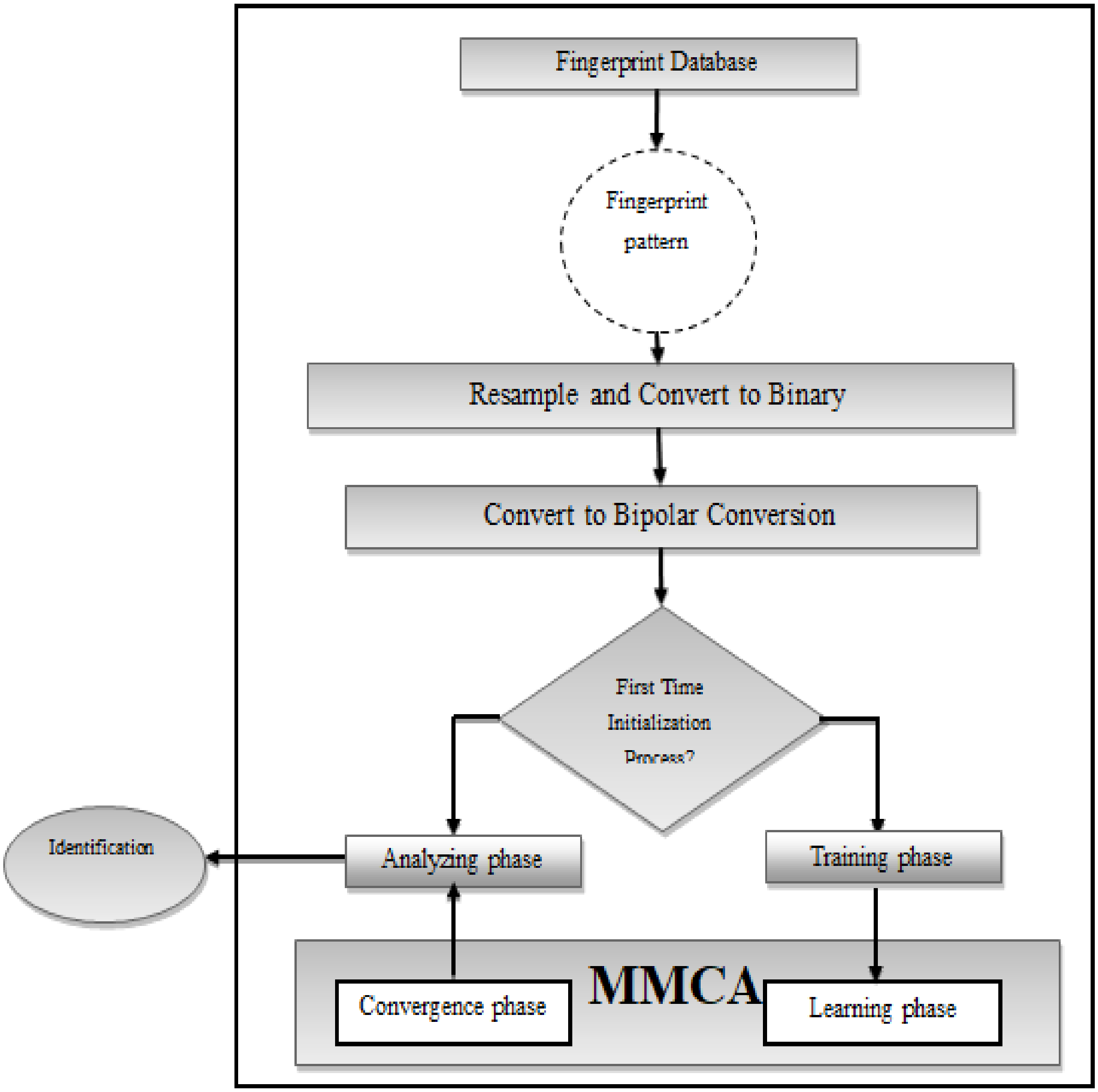


Figure 4.1: Framework of the Proposed Fingerprint recognition System.

4.2.1 Pre-processing Step

The preprocessing step consists set of operations to prepare the fingerprint patterns for next step. Preprocessing step includes steps as shown in Figure 4.2.

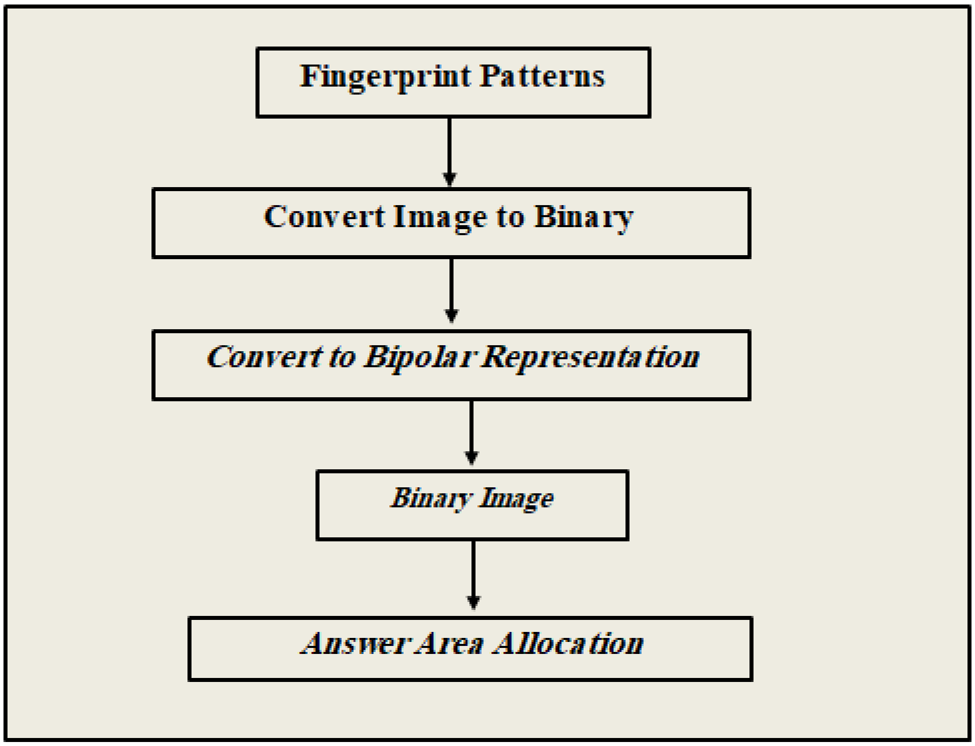


Figure 4.2: Pattern preprocessing step.

Fingerprint image pre-processing operations are image Segmentation and image binarization. The binary pattern is initially computed from the input grey scale fingerprint pattern by applying the Bradly technique. This technique ensures a better binarization and a shorter execution time compared with the constant TH. Then, the speed of the system can be increased by cutting and cropping the image region containing the fingerprint feature (ROI) as well as having to deal with only 50% from the finger pattern, not all. Also, it may be noted that the speed of the system increases without any reduction in the accuracy of the system.

Afterwards, the binary patterns are converted into a bipolar representation of the pixel values (either 1 or −1). This conversion reduces number of information to be processes and conserves important image-related data.

4.2.1.1 Pattern resampling

The pattern Fingerprint in the system proposed are different dimension (W×H) and high resolution, which mean a large numbers of pixels will be processed with a more time.

Fingerprint system is work in real time. Therefore, must be necessary to decrease the dimensions of pattern while saved quality, which is important to increase accuracy FP system. Method of Bilinear interpolation is used to reduce the dimensions (W×H) pixels for pattern , this method will be save quality . That method is showed in Algorithm (4.1).

Input
    OriginalImg (F)
Output
    ResamplingImg ($\hat{F}$)

Begin
Step1: //Compute scale factors (S_x, S_y)
    s_x ← WidthOrignalImg / WidthResamplingImg
    s_y ← HeightOrignalImg / HeightResamplingImg
Step2: For x← 0 to WidthResamplingImg loop
        For y← 0 to HeightResamplingImg loop
        //Compute (x′,y′) by applying Equation (2.6):
        x′← $S_x * x$
        y′← $S_y * y$
        Find(x1,y1,x2,y2)
        FindValues (F(x1,y1), F(x2,y1), F(x1,y2), F(x2,y2),a, b, x′,y′,x,y)
        //Compute estimated pixel value in ResamplingImg by applying Equation(2.5):
$$\hat{F}(x,y)=(1-b)\left[(1-a)F(x1,y1)+aF(x2,y1)\right]+b\left[(1-a)F(x1,y2)+aF(x2,y2)\right]$$
        End Loop
    End Loop
End

v

4.2.1.2 Pattern converting binariztion

Shadow and brightness during experiments and testing of Fingerprint pattern are a factor that affects conversion processes. By using a constant threshold value that lead to poor binarization, while by using Bradley method (local threshold)that lead to good binarization and reduce the execution time. Algorithm (4.2) illustrated integral image.

```
Input
    ResampledLP (f)
Output
    IntegralLPImage (I)
Begin
    For y=0 to image height
        For x=0 to image width
            //Compute integral image (I) by applying Equation (2.11)
            I(x,y) = f(x,y) + I(x − 1,y) + I(x,y − 1) − I(x − 1, y − 1)
        End Loop
    End Loop
End
```

Algorithm (4.3) is illustrated steps to convert Fingerprint pattern to binary depend on Bradley method.

```
Input
    ResampledLP (f)
Output
    Binary image of LP (BI)
Begin
Step1 : Call Generate Integral Image( f, I)
Step2:
    RectangleSize ← 31
    t ← 0.9
Step3:
    For y←0 to HeightImage loop
        Compute rectangle's Y coordinates (y1,y2, RectangleSize)
        For x←0 to WidthImage loop
            Compute rectangle's X coordinates (x1,x2, RectangleSize)
            //Compute average of rectangle by applied Equation (2.12):
            SumRect←I(x2,y2) − I(x2,y1 − 1) − I(x1 − 1, y2) + I(x1 − 1, y1 −
    1)
            AvgValue← (SumRect/((x2 − x1) ∗ (y2 −y1) )) ∗ t
            Get GSvalue from f(y,x)
            If GSvalue < AvgValue then
                BI(y,x)←0
            Else
                BI(y,x)←255
        End Loop
    End Loop
End
```

4.2.1.3 Bipolar conversion

Fingerprint proposed system used MMCA AM network, the fingerprint pattern must be converted binary pixel of pattern to bipolar representation of value (such as, each pixel in binary pattern must be either 1 or -1). With bipolar representation will be reduce amount of the information to be processed and conserve important data of pattern.

4.2.2 Recognition Step

Modify Multi-Connect Architecture (MMCA) is solve for recognition. In MMCA, training phase is not complicated but is limited to one sample for each Fingerprint Pattern. MMCA work in two phases is viz., training phase and analysis phase. In the training phase, each fingerprint pattern that is obtained during the learning phase will be processed by MMCA and stored in a lookup table for the next phase. Meanwhile, in the analysis phase, an analyst fingerprint recognition system is used to authenticate an unknown fingerprint, while the MMCA uses the lookup table that was built in the previous phase. Figure 4.3 and Algorithm (4.4:Figerprint Recognition System) illustrates these two phases.

Input: Fingerprint patterns. **Output:** Fingerprint patterns identification.
Step_1: Training Phase **Step_1.1**: Initialize Fingerprint Database. **Step_1.2**: Pre-processing for Fingerprint pattern. **Step_1.3**:Fingerprint will be processed by MMCA learning phase (See Algorithm 1). **Step_1.4**:Stored result in a lookup table. **Step_2:** Analysis Phase **Step_1.1**: Unknown Fingerprint Pattern. **Step_1.2**: Pre-processing for unknown Fingerprint pattern. **Step_1.3**:Fingerprint will be processed by MMCA Convergence phase (See Algorithm 2). **Step_3**: Compute Energy Function. **Step_4**: Select Minimum Value. **Step_5**: Fingerprint identification.

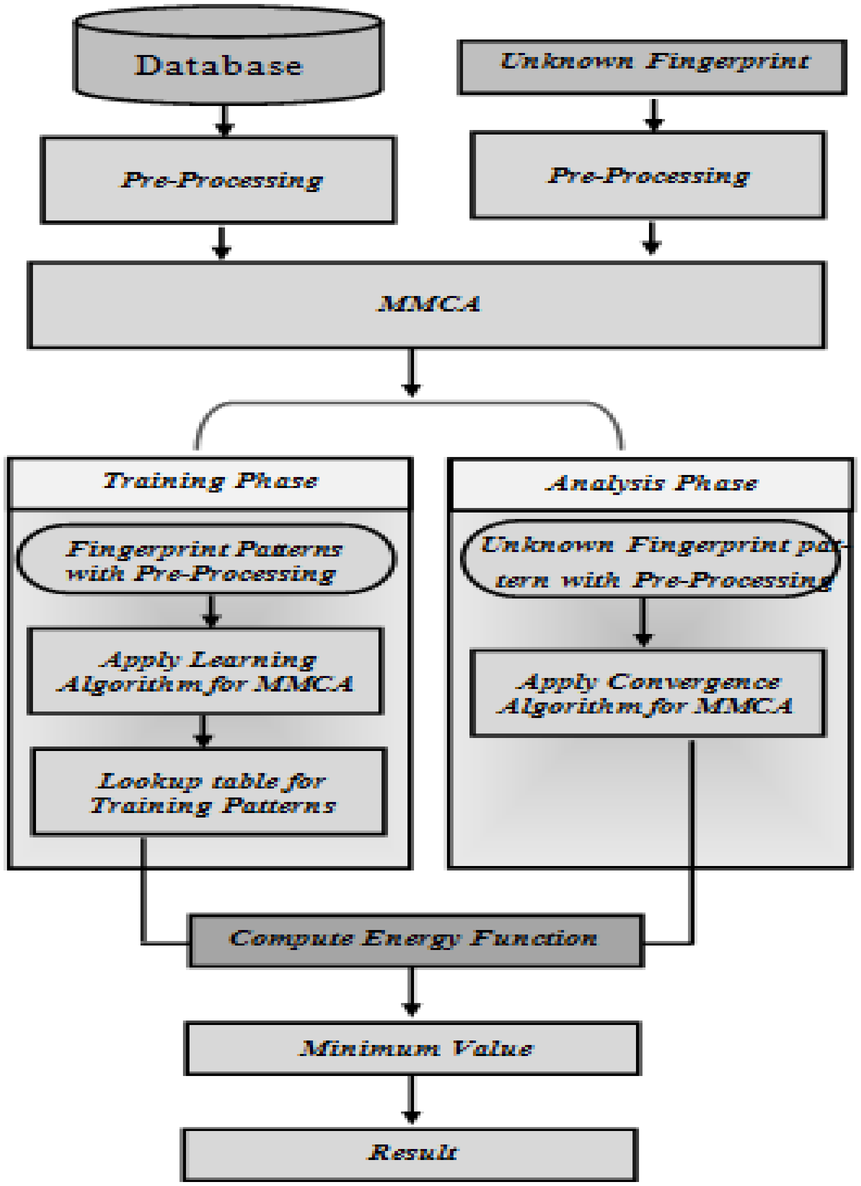


Figure 4.3: Training and Analysis Phases for implementing MMCA.

5. RESULTS AND DISCUSSION

5.1 INTRODUCTION

This chapter presents a complementary evaluation via a series of experiments. The results of these experiments have been analyze and discusses. Fingerprint system evaluation evaluation in order of its accuracy and average speed.

5.2 THE HARDWARE AND SOFTWARE REQUIREMENTS

The next subsections explain the environment (hardware and software specifications) of the Fingerprint system:

- **HARDWARE REQUIREMENTS**

1. Processor is Intel ® Core i7-4500U CPU @ 1.80GHz.
2. Memory is 8.00 GB

- **SOFTWARE REQUIREMENTS**

1. Operating System OS: Windows 10 Pro (64-bit).
2. Compiler: MATLAB & SIMULINK R2016a.

5.3 FINGERPRINT SYSTEM EVALUATIONS

To evaluate performance and stability of the proposed Fingerprint system, the databases that use for evaluate Fingerprint system are explain in chapter one. The Fingerprint patterns are transfer to computer and stored with TIF formats. The samples details of databases are shown in Table 5.1.

Table 5.1: Database.

Database	No. of Samples
FVC (2004)	640
Internal Database	2500
NIST	2000

5.4 TESTING FINGERPRINT SYSTEM WITH MMCA

In this experiment, percentage of identification was measured, three types of databases were tested as showed in Table 5.1. The accuracy results of each database are shown in Table 5.2.

Table 5.2: Accuracy of the Proposed System.

Database	No. of Samples	Recognition Rate	Recognised Patterns	Unrecognised Patterns
FVC (2004)	640	100%	640	0 Patterns
Internal Database	2500	99.48%	2487 patterns	13 patterns
NIST	2000	99.2%	1984	16 Patterns

The experiment shows that only 13 out of the 2500 patterns and 16 out of the 2000 patterns have not been recognized by the system from the internal and NIST databases, respectively, while recognizing all patterns from the FVC (2004). In this case, the system has an average accuracy of 99.56% from all databases and a total processing time of approximately 30 s. The processing time largely depends on the number of learning patterns, which, in other words mean that a larger number of patterns correspond to a longer processing time. In this experiment, the MMCA can process all patterns in the learning phase, while the network can recognize the unknown patterns in the convergence phase without facing any problems.

The binary techniques (i.e. TH and Bradly) are also compared in another experiment. By using the internal database, the TH technique obtains an accuracy of 98.92%; fails to recognize 27 of the 2500 patterns; and has a processing time of approximately 32 s. However, the Bradly technique has a shorter processing time and a higher accuracy compared with the TH technique. The Figures 5.1 and 5.2 present a comparison between the accuracy and processing time registered by both these techniques, respectively.

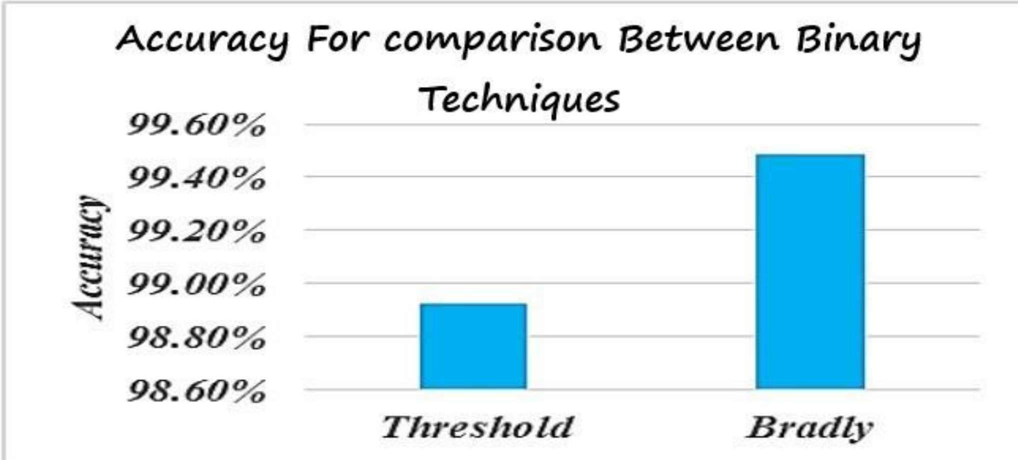


Figure 5.1: Accuracy of the Binary Techniques.

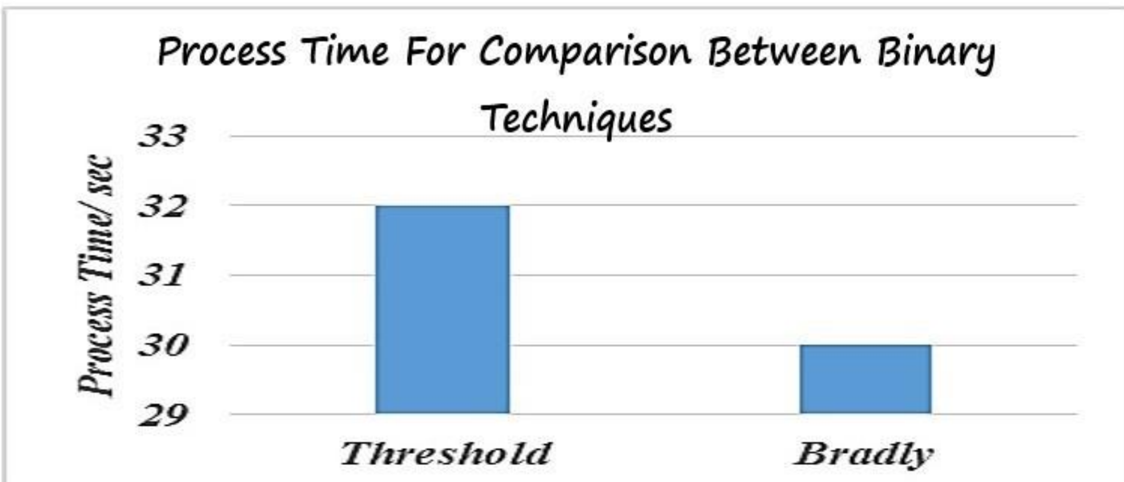


Figure 5.2: Processing Time of the Binary Techniques.

5.5 COMPARISON WITH OTHER WORKS

The literature review shows that only seven papers so far have used ANN to detect fingerprints. Accordingly, the accuracy of the proposed fingerprint system is compared with that of the methods applied in these papers as shown in Figure 5.3.

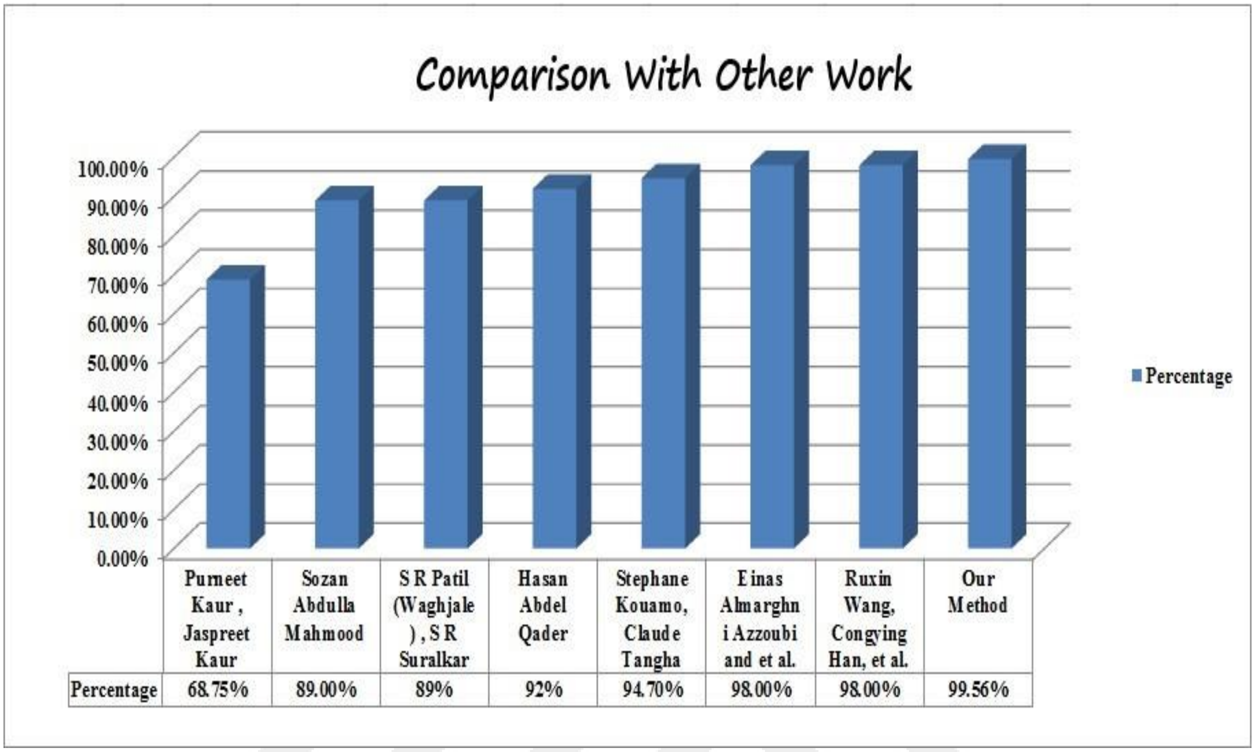


Figure 5.3: Average Accuracy of the Proposed System and of the Other Methods Applied in the Literature.

5.6 SUMMARY

The evaluation results of Fingerprint system are analyze and discussion in chapter 5. By using database as illustrated in Table 4.1, the evaluation of Fingerprint system using MMCA AM network, these system has an average accuracy of 99.56% from all databases and a total processing time of approximately 30 s. in additional in this chapter compression between binary techniques (i.e. TH and Bradly) are also compared in another experiment. By using the internal database, the TH technique obtains an accuracy of 98.92%; fails to recognize 27 of the 2500 patterns; and has a processing time of approximately 32 s. The implementation with MMCA associative memory presents higher accuracy result compare with other works.

6. CONCLUSION

1. This thesis focused in the development of a Fingerprint system via using associative memory with modify multi-connect architecture.
2. The MMCA allows Fingerprint system to recognize fingerprint through two phases: training phase and analyzing phase. In beginning, the training phase applied on the one fingerprint, for the analyzing process, it implemented for unknown fingerprint.
3. The proposed fingerprint biometric system has an average accuracy of 99.56% and a pattern recognition processing time of approximately 30 s, depending on the number of patterns to be processed. The proposed system applies AM with MMCA for identity authentication, thereby paving the way for future works to develop highly efficient, accurate and faster fingerprint biometric systems by modifying AM and using MMCA AM, which shows low complexity and consumes smaller amounts of memory.

REFERENCES

[1] A. K. Jain D. Maltoni, D. Maio and S. Prabhakar., Handbook of Fingerprint Recognition, Springer, New York, 2003.

[2] A. K. Jain, S. Prabhakar, and S. Pankanti, “On the Similarity of Identical Twin Fingerprints”, Elsevier Science, 35(11):2653–2663, 2002.

[3] Abdelwahed Motwakel and Adnan Shaout, Fingerprint Image Quality Fuzzy System, the International Arab Journal of Information Technology, Vol. 13, No. 1A, 2016).

[4] aio D, Maltoni D. A structural approach to fingerprint classification[C]//Pattern Recognition, 1996., Proceedings of the 13th International Conference on. IEEE, 1996, 3: 578-585 .

[5] Alasdair McAndrew, "An Introduction to Digital Image Processing with Matlab", School of Computer Science and Mathematics Victoria University of Technology, 2004.

[6] Anil K. Jain and et.al, Handbook of Biometrics, Springer, New York, 2008.

[7] Ashish B. I, Dr. D. S. C “Speech Emotion Recognition Using Hidden Markov Model and Support Vector Machine”, IJAERS/Vol. I/ Issue III/April-June, 2012/316-318.

[8] Bigun and F. Smeraldi (Eds.): Fingerprint Classification with Combinations of Support Vector Machines AVBPA 2001, LNCS 2091, pp. 253–258, 2001.

[9] Bowen J D. The Home Office automatic fingerprint pattern classification project[C]//Neural Networks for Image Processing Applications, IEE Colloquium on. IET, 1992: 1/1-1/5.

[10] Chaohong Wu, Zhixin Shi, and Venu Govindaraju, “Fingerprint Image Enhancement method Using Directional Median Filter”, Elsevier Science, 2004, pp. 8-9.

[11] Chris Solomon and Toby Breckon, "Fundamentals of Digital Image Processing", Wiley Blackwell, 2012.

[12] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, “Handbook of Fingerprint Springer Recognition”, Verlag, 2003.

[13] Derek Bradley and Gerhard Roth “Adaptive Thresholding Using the Integral . Image” , Journal of Graphics GPU and Game Tools ,Volume:12, Issue:2, pages 13-21, 2007.

[14] Eduardo Zurek and et al., Acoustic Fingerprint Recognition Using Artificial Neural Networks, ISSN 1870-9044; pp. 69–73, Polibits (50) 2014.

[15] Einas Almarghni Azzoubi and et al, “An Enhancement Algorithm Using Gabor Filter”, Journal of Theoretical and Applied Information Technology, 30 Th April 2015. V ol.74 No.3.

[16] Emad I Abdul Kareem, et al., “Multi Connect Architecture (MCA) Associative Memory : A Modified Hopfield Neural Network”. Intelligent Automation and Soft Computing, Vol. 18, No. 3, pp. 291-308, 2012.

[17] Emad I. Abdul Kareem, "A real time visual monitoring module for traffic conditions based on a modified auto-associative memory", Thesis (Ph.D.)--University Sains Malaysia, 2012.

[18] Emad I. Abdul Kareem, Aman Jantan, "MCA: A Developed Associative Memory Using Multi-Connect Architecture", International Journal of Intelligent Information Processing, Volume 2, Number 1, March 2011.

[19] Emad I. Abdul Kareem, et al., "Intelligent traffic light controller based on MCA associative memory", Science Journal of Circuits, Systems and Signal Processing, 2014.

[20] Emad I. Abdul Kareem, et al., "Multi Connect Architecture (MCA) Associative Memory: A Modified Hopfield Neural Network", Intelligent Automation and Soft Computing, Vol. 18, No. 3, pp. 291-308, 2012.

[21] Engelsma J J, Arora S S, Jain A K, et al. Universal 3D Wearable Fingerprint Targets: Advancing Fingerprint Reader Evaluations[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(6): 1564-1578.

[22] F reeman, J.A. and Skapura, D.M. Neural Networks: Algorithms, Applications and Programming Techniques. Addison-Wesley, 1991.

[23] Fan L, Wang S, Wang H, et al. Singular points detection based on zero-pole model in fingerprint images[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2008, 30(6): 929-940.

[24] Faustt, L., "Fundamental of Neural Networks Architecture, Algorithms and Application", Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994.

[25] Feng J, Zhou J, Jain A K. Orientation field estimation for latent fingerprint enhancement[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013, 35(4): 925-940.

[26] Gajanan Pandurang Khetri , Satish L. Padme , Dinesh Ch Jain Vrushsen P. Pawar, Fingerprint Pattern Recognition Using Back Propagation Algorithms, International Journal of Electronics and Computer Science Engineering , ISSN- 2277-1956) .

[27] Gold S, Rangarajan A. Softmax to softassign: Neural network algorithms for combinatorial optimization[J].Journal of Artificial Neural Networks, 1996, 2(4): 381-399.

[28] Gu J, Zhou J, Yang C, et al. Fingerprint recognition by combining global structure and local cues[J]. IEEE Transactions on Image Processing, 2006, 15(7): 1952-1964.

[29] Guo T, Liu X, Shao G. Fingerprint Classification Based on Sparse Representation Using Rotation-Invariant Features[C]//Proceedings of the 2012 International Conference on InformationTechnology and Software Engineering. Springer Berlin Heidelberg, 2013: 631-641.

[30] Gurney Kevin, "An Introduction to Neural Networks", Taylor & Francis, Inc., Bristol, PA, USA, 1997.

[31] Hassan Abdel Qader and et al., Fingerprint Recognition Using Zernike Moments, the International Arab Journal of Information Technology, Vol. 4, No. 4, 2007.

[32] Henry E R. Classification and uses of fingerprints [M]. HM Stationery Office, 1905.

[33] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.

[34] Hong JH,MinJK,ChoUK ,et al. Fingerprint classification using one-vs-all support vector machines dynamically ordered with Bayes classifiers [J]. Pattern Recognition, 2008, 41(2): 662-671.

[35] http://bias.csr.unibo.it/fvc2004.

[36] https://www.nist.gov/srd/nist-special-database-4.

[37] Igor Aleksander, Helen Morton, An introduction to neural computing, Chapman and Hall, 1990

[38] J J Hopfield, Neural networks and physical systems with emergent collective computational abilities, IEEE, PNAS April 1, 1982 79 (8) 2554-2558;

[39] K. Amendt, D. Friend , “Fingerprint Identification”, MIT Course 6.111: Digital Electronics Lab, 2006.

[40] K. Jain, L. Hong and R. Bolle, "On-Line Fingerprint Verification", IEEE Trans. Pattern Anal. Machine Intell., Vol. 19, . No. 4, 1997, pp. 302-314.

[41] Karu K, Jain A K. Fingerprint classification[J]. Pattern recognition, 1996, 29(3): 389-404.

[42] Kawagoe M, Tojo A. Fingerprint pattern classification[J]. Pattern Recognition, 1984, 17(3): 295-303.

[43] KishanMehrotra and C.K.M.a.S.R., Elements of Artificial Neural Networks. Bradford Books, Complex Adaptive Systems, New York, USA, 1996.

[44] L. Almajmaie et al. " Fingerprint recognition system based on modified multi-connect architecture (MMCA) " Cognitive Systems Research 58 (2019) 107-113.

[45] M. Antowiak and K. Chalasinska-Macukow, "Fingerprint Identification by Using Artificial neural network and Optical Wavelet Preprocessing", OPTO-ELECTRONICS REVIEW 11(4), 2003, pp 327-337.

[46] Maio, D., Maltoni, D.," Neural network based minutiae filtering in fingerprints" IEEE Xplore, 1998.

[47] Maltoni D, Maio D, Jain A K, et al. Handbook of fingerprint recognition[M]. springer, 2009.

[48] Mohamed MostafaAbd Allah "Artificial Neural Networks Based Fingerprint Authentication with Clusters Algorithm", Informatica Vol. 29, 2005, pp 303–307.

[49] Mohamed S M, Nyongesa H O. Automatic fingerprint classification system using fuzzy neural techniques[C]//Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on. IEEE, 2002, 1: 358-362.

[50] Nelson J D. Finding useful questions: on Bayesian diagnosticity, probability, impact, and information gain[J]. Psychological review, 2005, 112(4): 979.

[51] Pankaj Mehra , Benjamin W. Wah., Artificial Neural Networks: Concepts and Theory , EEE COMPUTER SOCIETY PRESS TUTORIAL– September 1, 1992.

[52] Paul Viola , Michael J. Jones, " Robust Real-time Face Detection" , Springer, International Journal of Computer Vision, Volume:57, Issue :2, Pages:137-154, 2004.

[53] PurneetKaur, JaspreetKaur, Fingerprint Recognition Using Genetic Algorithm and Neural Network, International Journal of Computational Engineering Research||Vol, 03 Issue, 11 Issn 2250-3005 November 2013.

[54] Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, second Edition, Prentice Hall, 2002.

[55] Rakesh S, Kailash Atal, Ashish Arora, "Cost Effective Optical Mark Reader", International Journal of Computer Science and Artificial Intelligence, Vol. 3 Iss. 2, PP. 44-49, Jun. 2013.

[56] Rao K, Balck K. Type classification of fingerprints: A syntactic approach[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1980 (3): 223-231.M.

[57] Rao, S. B. N. ; Krishna, N. ; Prasad, J. R. ; Raghavan, G. V., 1995. Evaluation of nutritional quality of some fibrous resources by in vitro and nylon bag techniques in goats. Indian J. Anim. Nutr., 12 (2): 85-90.

[58] Richard P. Lippmann, An Introduction' to Computing with Neural Nets, IEEE ASSP Magazine (Volume: 4 , Issue: 2 , Apr 1987).

[59] Richard Szeliski , “Computer Vision Algorithms and Applications”, Springer , 2011.

[60] RusulHusseinHasan, “Optical Mark Reader Based on Modified Multi-Connect Architecture MMCA", M.Sc. Thesis, Al Mustansiriyah University, 2016.

[61] RusulHusseinHasan,EmadIAbdulKareem, "Modify Multi-Connect Architecture (MMCA) associative memory",Conference of the Mathematics and Computer Departments, Al Mustansiriyah University, 2017, Vol2.

[62] Ruxin Wang, Congying Han, Yanping Wu, and TiandeGuo, Fingerprint Classification Based on Depth Neural Network, arXiv:1409.5188v1 [cs.CV] 18 Sep 2014)

[63] S R Patil (Waghjale), S R Suralkar, Neural Network based Fingerprint Classification, International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064, Volume 2 Issue 1, January 2013.

[64] SalilPrabhakar, Anil K. Jain, “Learning Fingerprint Minutiae Location and Type“ International Conference on Pattern Recognition(ICPR), 2000.

[65] Shao G, Han C, Guo T, et al. An NMF-Based Method for the Fingerprint Orientation Field Estimation [M]//Computer and Information Science 2012. Springer Berlin Heidelberg, 2012: 93-104.

[66] Solanki V. Gautam, "Neural Network and Its Application in Pattern Recognition", Seminar Report of Department of Computer Science and Eng, Indian Institute of Technology, Bombay, November 5, 2004.

[67] Sozan Abdulla Mahmood, FINGERPRINT RECOGNITION SYSTEM USING SUPPORT VECTOR MACHINE AND NEURAL NETWORK, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) ISSN(P): 2249-6831; ISSN(E): 2249-7943 Vol. 4, Issue 1, Feb 2014, 103-110.

[68] StephaneKouamo, Claude Tangha, Fingerprint Recognition with Artificial Neural Networks: Application to E-Learning, Journal of Intelligent Learning Systems and Applications, 2016, 8, 39-49.

[69] TerjeKristensen “Two Different Regimes of Fingerprint Identification – a Comparison”, American Journal of Computational and Applied Mathematics 2012, 2(2):pp 1-9.

[70] Wang X F, Huang D S. A Novel Density-Based Clustering Framework by Using Level Set Method[J]. IEEE Transactions on Knowledge & Data Engineering, 2009, 21(11):1515-1531.

[71] Watson C I, Wilson C L. NIST special database 4[J]. Fingerprint Database, National Institute of Standards and Technology, 1992, 17.

[72] Wu Y, Zhao T, Wang S, et al. A SVM-based method for the estimation of fingerprint and palmprint orientation[C]//Intelligent Control and Information Processing (ICICIP), 2013 Fourth International Conference on. IEEE, 2013: 343-348.

[73] Y. Yoon, K. Ban, H.Yoon, J. Lee, J. Ki , "Best Combination of Binarization Methods for License Plate Character Segmentation", ETRI Journal, Volume:35, Number 3, 2013.

[74] Yang J, Yu K, Gong Y, et al. Linear spatial pyramid matching using sparse coding for image classification[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 17941801.

[75] Z. Shi, V. Govindaraju "A chaincode based scheme for fingerprint feature extraction", Pattern Recognition Letters 27, 2006, pp 462–468.

[76] Zhang Q, Yan H. Fingerprint classification based on extraction and analysis of singularities and pseudo ridges[J]. Pattern Recognition, 2004, 37(11): 2233-2243.

[77] Zurada, Jacek M., "Introduction to Artificial Neural System", Publishing Company, USA, 1992.