T.C.

ALTINBAŞ UNIVERSITY

Graduate School of Science and Engineering

Information Technology

FINGERPRINT RECOGNITION BY USING CONVOLOUTIONAL NEURAL NETWORK AND SUPPORT VECTOR MACHINE CLASSIFICATION

Ali Abdulhasan Johni AL-SAEDI

Master of Science

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Istanbul, 2021

FINGERPRINT RECOGNITION BY USING CONVOLOUTIONAL NEURAL NETWORK AND SUPPORT VECTOR MACHINE CLASSIFICATION

by

Ali Abdulhasan Johni AL-SAEDI

Information Technologies

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

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 master of science

Asst. Prof. Dr. Abdullahi IBRAHIM

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	School of engineering and natural science, Altinbas University	___________________
Asst. Prof. Dr. Mohammed ILYAS	School of engineering and natural science,, Altinbas University	___________________
Asst. Prof. Dr. A. Mohammed ABUBAKAR	Faculty of Business Antalya Bilim university	___________________

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

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.

Ali Abdulhasan Johni AL-SAEDI

DEDICATION

I would like to thank Allah Almighty for the power of mind, health, strength, guidance, knowledge and skills to complete this study.

This thesis is wholeheartedly dedicated to my parents. There are no words to describe what you mean to me; there is nothing that I can repay for what you have done for me. I will continue to do my best to achieve your expectations.

Lastly, I dedicated this to my brothers and sisters, relatives, and friends who have been encouraging me during this study.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Professor Dr. Abdullahi Abdu IBRAHIM for all support during my study. It’s great pleasure to express my deepest gratitude to my friends who have shared with me best moments during my study for the Master degree.

ABSTRACT

FINGERPRINT RECOGNITION BY USING CONVOLOUTIONAL NEURAL NETWORK AND SUPPORT VECTOR MACHINE CLASSIFICATION

Ali Abdulhasan Johni AL-SAEDI

M.Sc., Information Technology, Altınbaş University

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: 02/2021

Pages: (54)

Biometrics seeks to solve the problems of traditional verification methods by using certain physiological properties associated with an individual. Among all the biometric indicators, fingerprints have been shown to have good levels of reliability. The most widely used local representation is based on the details (minutiae) of the fingerprints. The pattern of the minutiae on a fingerprint forms a valid representation of the fingerprint. The minutiae that are most used for automatic recognition are branches and endings. However, given fingerprint acquisition techniques, it is common for endings and bifurcations to undergo deformations, which is why they are commonly referred to as minutiae. That is why in this document we will simply refer to these characteristics as minutiae. In this work we describe the results obtained using a methodology proposed for the recognition of minutiae using convolutional neural networks CNN, trained with different databases that contain fingerprints then we use the support vector machine classification to classify newly input images of fingerprints based on the features extracted by the CNN and matched with the dataset, our method proves to have better accuracy and lower MSE than the previous linear methods use for fingerprint recognition.

Key words: CNN, SVM, Neural Network, Minutiae, MSE.

ÖZET

FINGERPRINT RECOGNITION BY USING CONVOLOUTIONAL NEURAL NETWORK AND SUPPORT VECTOR MACHINE CLASSIFICATION

Ali Abdulhasan Johni AL-SAEDI

M.Sc., Information Technology, Altınbaş University

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: 02/2021

Pages: (54)

biyometri, bir bireyle ilişkili belirli fizyolojik özellikleri kullanarak geleneksel doğrulama yöntemlerinin sorunlarini çözmeye çalişir. tüm biyometrik göstergeler arasinda parmak izlerinin iyi düzeyde güvenilirliğe sahip olduğu gösterilmiştir. en yaygin olarak kullanilan yerel temsil, parmak izlerinin ayrintilarina (minutiae) dayanmaktadir. bir parmak izindeki ayrintilarin deseni, parmak izinin geçerli bir temsilini oluşturur. otomatik tanima için en çok kullanilan ayrintilar dallar ve sonlardir. bununla birlikte, parmak izi edinme teknikleri göz önüne alindiğinda, sonlarin ve çatallanmalarin deformasyona uğramasi yaygindir, bu nedenle bunlara genellikle minutiae denir. bu nedenle, bu belgede bu özelliklere sadece küçük ayrintilar olarak değineceğiz. bu çalişmada, parmak izlerini içeren farkli veri tabanlariyla eğitilmiş evrişimli sinir ağlari cnn kullanilarak minutiae'nin taninmasi için önerilen bir metodoloji kullanilarak elde edilen sonuçlari açikliyoruz, ardindan parmak izlerinin yeni giriş görüntülerini çikarilan özelliklere göre siniflandirmak için destek vektör makinesi siniflandirmasini kullaniyoruz. cnn tarafindan ve veri setiyle eşleştirilen yöntemimiz, parmak izi tanima için kullanilan önceki doğrusal yöntemlerden daha iyi doğruluğa ve daha düşük mse'ye sahip olduğunu kanitladi.

anahtar kelimeler: cnn, svm, sinir ağı, minutiae, mse.

TABLE OF CONTENTS

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

KNN	:	K nearest number
DWT	:	Discrete Wavelet Transform
CNN	:	Convolutional neural network
PCA	:	Principal Component Analysis
NN	:	Neural network
GD	:	Gender Detection

LIST OF SYMBOLS

μ	:	Mobility of electron
λ	:	Wave length
ω	:	Angle frequency

INTRODUCTION

1.1 BACKGROUND

Its earliest known use dates back to Ancient Babylon, where kings signed clay tablets by engraving their fingertips before firing them. Later, in China, during the Tangen Dynasty in 650 AD, it was established that to divorce a woman, the husband had to state seven reasons and sign the document with his fingerprints. Also, in India, this mark was soon used in legal documents. In 1901 a fingerprint commission was formed to have a database of civilians, it was created by Edgard Henry, and in 1902 Harry jackson was jailed using this database. In 1902, 1,722 fingerprints were identified in Scotland using the Edgard Henry system; which compares to the almost 400 identifications made by the Bertillion system in the years 1894 and 1900:

Figure 1.1: Babylonian fingerprints in the reign of Hammurabi.

1.2 MOTIVATION

Because biometrics is a relatively new science whose goal is to make computers capable of recognizing each person with certainty based on their unique characteristics, such as fingerprints, features, or the signs of their eyes; A biometric prototype is implemented that is capable of recognizing people through the fingerprint. There are several types of measurement that can be moved for biometrics. The ones most know, such as fingerprint, hand geometry, and eye analysis, are also the best performing today. Thanks to the characteristics of fingerprints; which are unique and invariant over time for each individual, a prototype is implemented in which people are automatically identified, based on the characteristics of the fingerprint of the index finger.

1.3 PROBLEM STATEMENT

It is known that the recognition of fingerprints through electronic systems has been of great importance for military use, security, personnel control, homicide investigations, among others, this because fingerprints are unique to each person and invariant with time and this facilitates having better control and security of individuals as the case may be This is why many fingerprint-based recognition systems have been implemented over time. Initially these fingerprint recognition systems were more appropriate for the various police institutions or security departments in European powers and the United States (FBI) to find out crimes and have a fingerprint base of people who for various reasons have visited the jail. Current fingerprint recognition systems have advanced so far that personal computers and even certain usable removable memories have a fingerprint scanner so that only the authorized person (s) can access the files contained in them:

Figure 1.2: Current fingerprint recognition systems.

1.4 CONTRIBUTION

The main innovation is to build a system that uses the features extracted from a scanned digital image of a fingerprint and classify these features into labels in order to build a dataset of fingerprint features, then we use the convolutional neural network to extract the features of a new scanned fingerprint and classify the fingerprint into a label from the dataset using the support vector machine algorithm.

1.5 THESIS STRUCTURE

The thesis is structured as follows: Section 2 is where we review some of the previous work and implementation on smart grids. Section 3 is where we give a background about all the components. Section 4 is where we explain our model in details and implementation of that model in details, Section 5 is where we simulate our model and record the results obtained. Section 6 is where we conclude our work and put a future scope into perspective.

LITERATURE REVIEW

Feng et.al [1] implemented a field-oriented scheme for enhancement of latent fingerprint, while Munussami et.al [2] proved that there are archaeological findings found that indicate that fingerprints have been used in the identification of individuals since 6000 B.C. by various Chinese and Assyrian populations. Among them, it is worth noting, the clay ceramic remains with fingerprints, identifying their use as a means of identifying the potter. Some Chinese documents of the time also include stamps stamped with the signer's thumb print. The bricks used to fingerprint the houses of the ancient city of Jericho were sometimes marked by the thumb print of the workers. Park et.al [3] reviewed the first published scientific study on the structure of ridges, valleys and pores of fingerprints dates from 1684, carried out by the English morphologist Nehemiah Grew. Since then, many researchers have worked in this field. In 1823, Purkinje proposed a footprint classification scheme into nine classes based on the configuration of the ridge structure. In general, the studies of the early 1800s reached two important conclusions that, until today, have served as the basis for biometric recognition, especially in forensic settings: the absence of two different individual footprints with a coincident crest pattern, and the invariance over time of these patterns throughout the life of the individual. In the early 1900s Chen et.al [4] admitted the following biological characteristics of fingerprints as the basis for identifying individuals Details The details of the ridge and valley structures, as well as the minutiae, are particular to each individual and unchanged in time. The first and third characteristics specify the principles that govern the identification of individuals by their fingerprints. The second characteristic constitutes the principle that allows the classification of fingerprints. Difference between ridges and valleys. In 1684 [5], the English and pioneer of fingerprint Nehemiah Grew, published the first scientific treatise explaining the structure of the ridges, valleys and pores of fingerprints. These studies led to the use of fingerprints in criminal identification, first in Argentina, in 1896; in Scotland Yard, in 1901 [6]; and in other countries in the early 1900s. In 1888, Sir Francis Galton made a detailed and extensive study of fingerprints, introducing the concept of minutiae as an identity characteristic of individuals. From the early 1800s the identification of individuals from fingerprints was formally accepted, beginning to be routine practice of forensic applications. Throughout the world, law enforcement agencies were established.

fingerprint identification and be created in criminal databases. Different techniques of latent fingerprint acquisition, identification, classification and pattern comparison were developed. In 1924 [7], the FBI's first fingerprint identification division was installed. Automatic fingerprint recognition began in the early 1960s. Since then, automatic fingerprint identification systems have been used in law enforcement institutions around the world. In the 1980s, with the development of personal equipment and electronic capture devices, automatic identification systems were used to use in non-criminal applications, such as access control in security environments. In the late 1990s, the development of solid-state capture devices, their low cost, and the development of accurate and reliable pattern recognition algorithms, have contributed to the rapid expansion of fingerprint-based biometric recognition systems [8].

MATERIALS AND METHODS

In this chapter we review some of the essential components that we used to implement our model, we give a brief introduction to the component, and then we show how we simulate that component inside MATLAB environment.

1.6 FINGERPRINT FEATURES

Fingerprints have certain aspects of their own and these are mentioned below:

1- pernnialism: The impressions appear in the individual from the sixth month of intrauterine life to the breakdown of the skin after death.

2- Immutability: They are unalterable since the number of lines does not increase or decrease, nor does any detail change, nor are the proportions altered by growth.

3-Infinite Variety: Fingerprints are absolutely different for each individual, there are no 2 equal impressions. [5]

The fingerprint patterns are divided into 4 main types, all of them mathematically detectable. This classification is useful at the time of verification in identification. There is no exact definition to say what the left and right loop is, but as we can see in figure 3.1 we see that this is an arch which has an inclination and is on the right or left side of the track. Instead the arch is drawn to the center of the print. The Verticals, as can be seen in the image, has an “S” shape drawn on the fingerprint.

Identifiable Points:

In figure 3.2 the 8 characteristic points on a finger appear, these are repeated interchangeably to form between 60 and 120 (for example 10 orches 12 joints 15 islets, etc.). These points are also called minutiae, term used in forensic medicine that means "characteristic point" [6]. Each fingerprint has arcs, angles, swirling loops etc. (called minutiae)

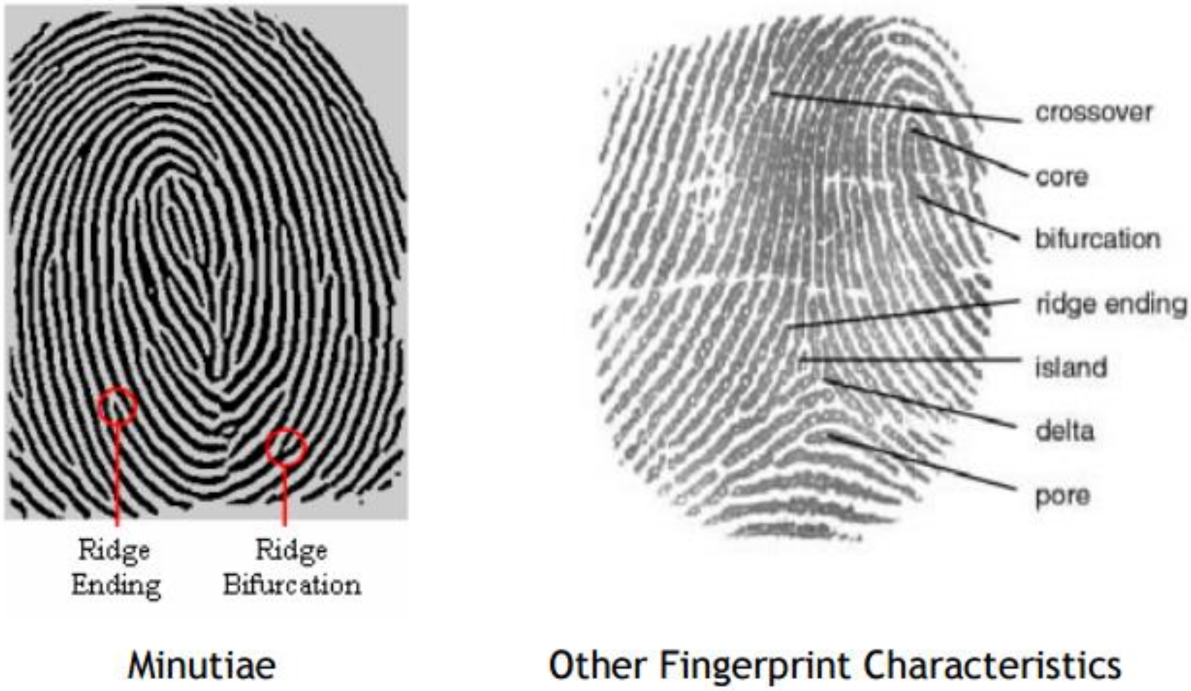


Figure 3.1 Fingerprint features

As we observe in the previous image, there are different figures that can be formed in the fingerprint. Here, for example, the islet that is a curve, the bifurcation when the line breaks in two and so on, but within each of these there is a variation. for each of the shapes. There is a science in charge of studying people's fingerprints and this science is called fingerprint. It is defined below: Fingerprinting is the science that aims to identify people physically considered by means of the physical printing or reproduction of the drawings formed by the capillary ridges on the fingertips.

1.6.1 The Dactylogram and Its Topography

The dactylogram is the digital impression taken directly on the paper, with ink (lithographic). See annexes which is lithographic ink and its preparation. Each dactylogram is made up of three zones of invasion as follows Since the fingerprint has many patterns to be identified, we can use them in an artificial neural network to classify them because they can differentiate figures or shapes with the help of unique patterns. Due to these unique characteristics that fingerprints have on people, they can help us identify the person without having to take their entire hand.

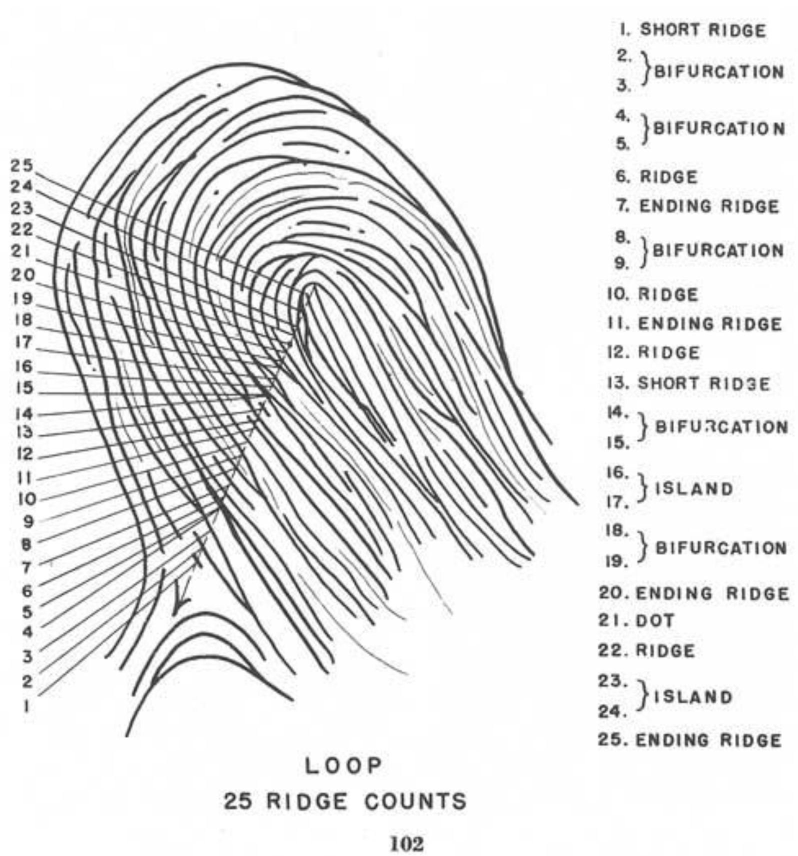
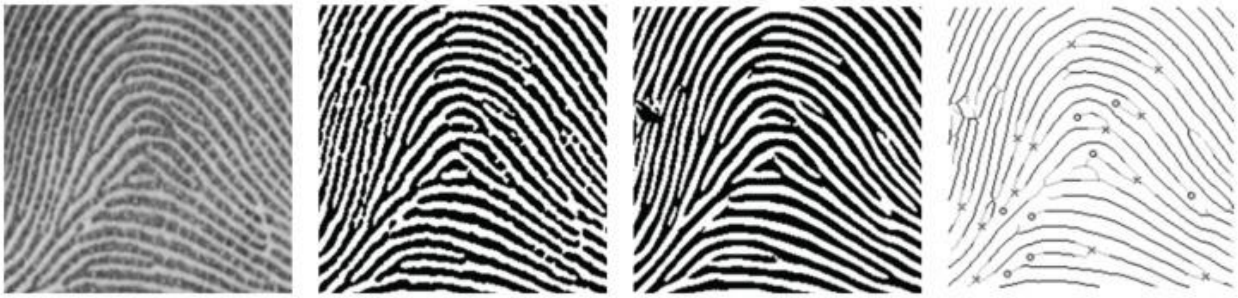


Figure 3.2 Fingerprint features

1.6.2 Pre-Processing for Digitization of Fingerprint Images

The fingerprint recognition prototype in its initial form contains the image of the fingerprint (inked) on paper, then it is scanned; for image processing. This processing turned out to be quite complex and inefficient; since the image obtained by this technique contains many levels of noise (the lines of the fingerprints are not well distinguished, too many variations in fingerprints of the same person, and the inconvenience of inking the finger to the user who wanted to be recognized) and the pre-processing of these images obtained turned out to be very complex to carry out their binarization.



(a) Original image (b) Binarized image (c) Smooth image (d) Minutiae extraction after thinning

Figure 3.3 Fingerprint preprocessing

For this reason, a fingerprint scanner was acquired to make the process of digitizing the image of the fingerprint easier to recognize; and seek the comfort of the user to be recognized. This is why you get a reliable and efficient fingerprint scanner for fingerprint imaging.

The main technical advantage in the use of a fingerprint scanner is that the digitized image in this .bmp format reduces noise in it and is segmented (it consists of separating the black area background from the rest of the image); For this reason, the acquisition of a scanner in the recognition prototype becomes necessary; the features required for this scanner are:

1. Economic feasibility
2. Proper interface to the computer
3. Adequate quality of the digitized image
4. A standard sensor area for the index finger.

The sensor will only be used for educational purposes in the realization of the prototype, so economic feasibility becomes essential, the USB 2.0 interface is required to decrease the digitization time of the fingerprint, the quality of the digitized image decreases the time of image pre-processing.

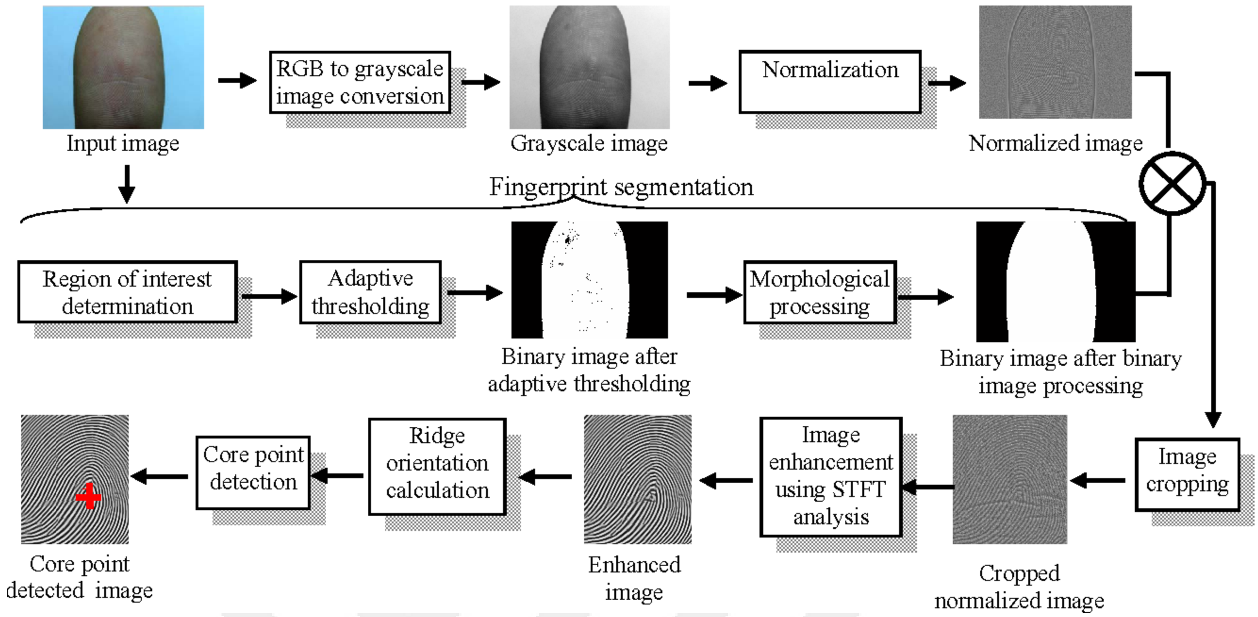


Figure 3.4 Fingerprint acquisition method

3.2 CONVOLOTIONAL NEURAL NETWORK

Biological neural networks are vast networks of neurons interconnected by chemical and electrical processes. All these neurons connected to each other allow the human being to be able to feel, memorize, learn, etc. All this involves asking if it is possible to build or create some rudimentary type of neural network and train it to be able to carry out certain tasks. The answer to the previous question is yes, and the main objective of this section is to carry out simple abstractions of complex neural networks in order to implement and train them to solve specific problems. As previously discussed, the simplified and abstract model of artificial neural networks arises from trying to imitate the behavior of neurons found in the brain.

In essence the brain is made up of a huge number of neurons (approximately 1011) connected to each other creating the neural network. Neurons have three main parts: the dendrites, the body, and the axon. Dendrites are nerve fibers that carry electrical impulses to the neuron's body. The body of the neuron is in charge of recognizing and working with the signals that come to it. Finally, the axon is a single nerve fiber that communicates the body of the neuron with the others. The point of contact between the axon and the dendrite of another neuron is called the synapse. The arrangement of neurons and the forces of each synapse individually, determined by chemical reactions, establish the functioning of the brain. This structure of neurons can be seen in the figure 3.5

Part of the neural network that forms our brain is determined when a human being is born, however, many other parts develop over time. Through learning these parts are developed by creating new connections between neurons or modifying existing ones, strengthening or weakening. Even taking into account the great differences, biological and artificial networks share certain characteristics. First, they share the general structure in which neurons are have many connections between them. Second, the force in the connections between neurons determines the behavior of the entire network.

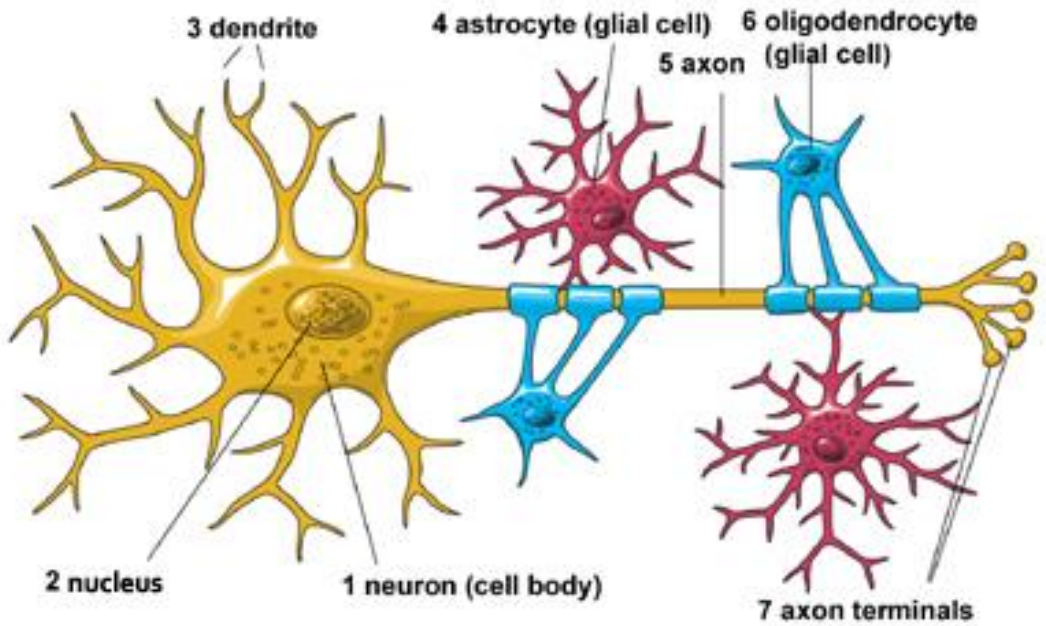


Figure 3.5 biological neural network

Getting to design and build machines capable of carrying out processes with a certain intelligence has been one of the main objectives of scientists throughout history. From the attempts made in this sense, the fundamental lines for obtaining intelligent machines have been defined: Initially, efforts were directed at obtaining automata, in the sense of machines that carried out, with more or less success, some typical function of human beings. Nowadays, it continues to study in this same line, with surprising results, there are ways to carry out processes similar to smart ones and that we can fit within the so-called Artificial Intelligence (AI).

The other line of research has tried to apply physical principles that govern in nature to obtain machines that do heavy work in our place. In the same way, one can think about the form and capacity of human reasoning; you can try to obtain machines with this capacity based on the same principle of operation.

It is not about building machines that compete with human beings, but rather perform certain tasks of an intellectual rank with which to help you, the basic principle of Artificial Intelligence.

The first theoretical explanations about the brain and thought were already given by Plato (427-347 B.C.) and Aristotle (348-422 B.C.). The same ideas were also maintained by Descartes (1569-1650) and the empiricist philosophers of the eighteenth century.

The class of so-called cybernetic machines, to which neural computing belongs, has more history than is believed: Herón (100 BC) built a hydraulic automaton.

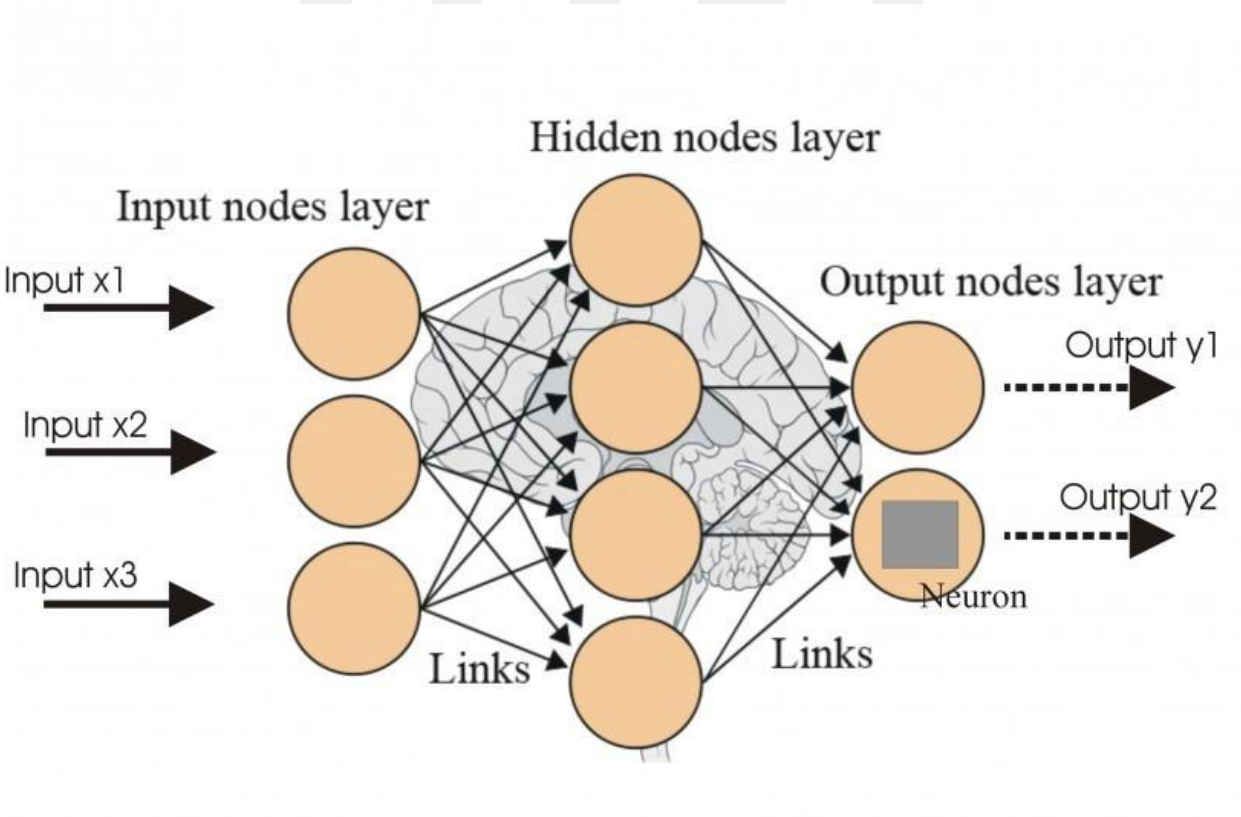


Figure 3.6 Neural network layers

3.2.1 CNN Architecture

When working with images, as in the case of convolutional neural networks, it is usually done with images with very high resolutions (for example, the current standard is Full HD consisting of 1,980 x 1,080 pixels). This entails a serious problem in conventional neural networks, which have been seen previously. These, being full-connected, have each of the neurons that make up the first hidden layer connected to each and every one of the pixels of the input image at the same time. If these input images are of an acceptable resolution, for example 28 x 28, the number of elements that will have to be trained will be moderate enough so that the system can work correctly, although the whole process will be significantly slower. However, if the resolution increases slightly (let alone current resolutions) the training and testing times become enormous.

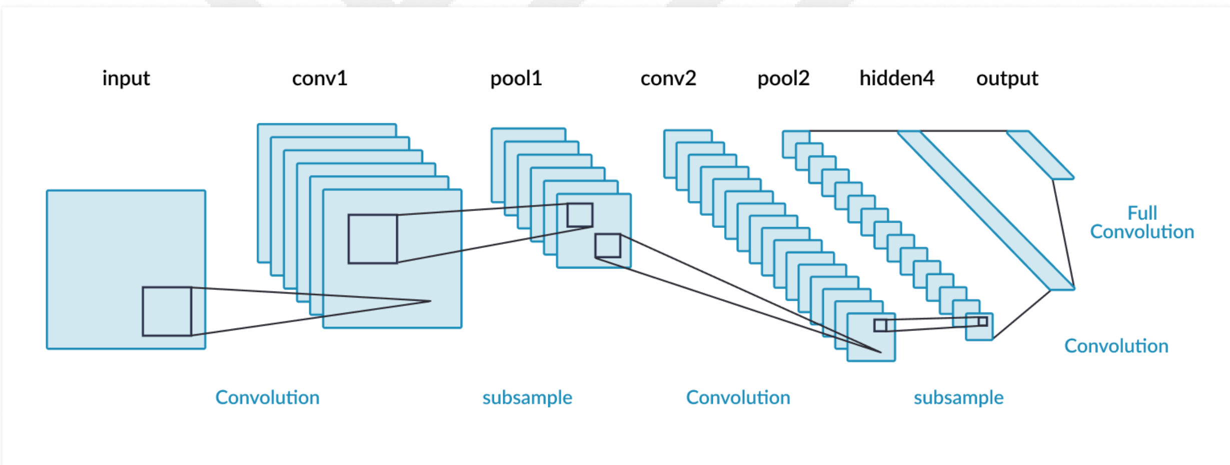


Figure 3.7 CNN layers

Next, the magnitude of the previous problem will be presented in a quantitative way. If, for example, 96 x 96 images are used, there would be around 104 input elements and, if it is assumed that you want to learn 100 characteristics, the number of elements to train would be of the order of 106. This enormous number of elements that they must be trained makes the duration of the training extraordinarily long, greatly hindering the use of conventional neural networks for imaging applications.

Once you are aware of the magnitude of the problem, the need to use a different type of system that allows you to get around this issue becomes evident. In the following points in this topic, you will see how this problem is solved.

A convolutional neural network is a type of multilayer network consisting of several alternate convolutional and pooling (subsampling) layers, and ultimately has a series of full-connected layers such as a multilayer perceptron network. The input of a convolutional layer network is usually an image i j i j k, where i is both the height and the width of the image and k is the number of channels (as will be seen later, in this project we work with grayscale so k). The convolutional layers have l filters (or kernels) whose dimensions are m, where and m are chosen by the designer (generally n is usually equal to 5). Each filter generates by convolution a map of features or characteristics of size 6> 6>, being the number of filters, you want to use. Then each map is sub-sampled in the pooling layer with the “mean pooling” or “max pooling” operation on contiguous regions of size where it can take values from 2 for small images to, commonly, no more than 5 for large images Before or after subsampling, a sigmoidal activation function plus a bias is applied to each feature map.

Returning to the problem presented in this chapter, which consisted of reducing the computational load of the system, the convolutional layer plays an important role in this regard. To do this, during this stage of the network, a very simple solution is carried out: the number of possible connections between the neurons of the hidden layer and the elements of the input image is restricted. In this way, each hidden neuron will only be connected to a small subset of elements of the total image. This idea is inspired by how the biological visual system works, because the neurons that work in the visual cortex have a series of receptive fields that respond only to stimuli located in a specific region or area.

In addition, another notable feature is that natural images have the property of being "stationary", this means that the characteristics or features that exist in a certain part of the image may be the same as others that are in another totally different area.

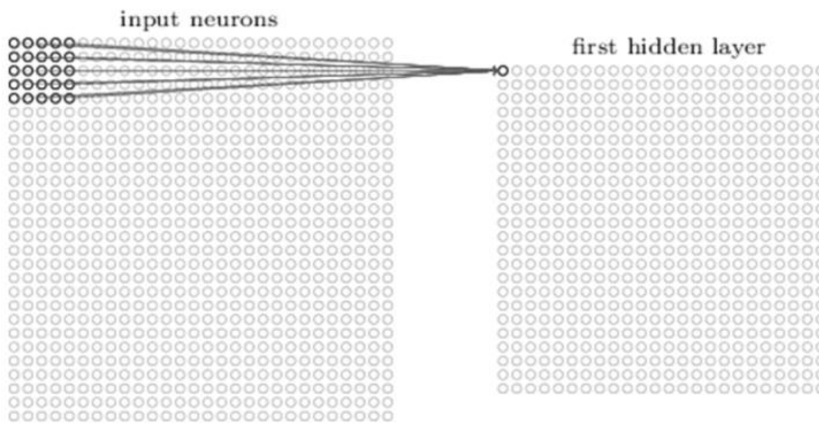


Figure 3.8 CNN hidden layer

On the other hand, convolution is an operation of products and sums between the input image and a filter (or kernel) that generates a characteristics map. The advantage is that the same filter (= neuron) serves to extract the same feature in any part of the image, taking into account the stationary nature of the images that has been commented on in the previous paragraph. This allows to reduce the number of connections and the number of parameters to train compared to a full-connected multilayer network.

In summary, the convolutional layer allows both to reduce the number of elements that make up the network and to detect a series of characteristics that will be useful when analyzing the image. Considering the above, it can be concluded that the convolutional layer is very useful in image analysis, reducing the complexity of the system and, at the same time, extracting useful features from them that help with its analysis.

As explained in point 4.1, the input of a convolutional layer is an image i j i j k, a filter with dimensions' p j p j NY is applied to this image, finally, as a result, a matrix of characteristics is obtained of size i> p 6>. The above behavior is achieved by performing the procedure explained in the following paragraphs:

When the input image arrives at the network, the filter and this are superimposed and the two-dimensional convolution between the respective elements of the image and the kernel is calculated. Once the result of the previous operation is obtained, it is stored in a position of the activation matrix, the filter is then moved or slid one position to the right on the image and the convolution is recalculated, storing the result at the next position in the activation matrix. In this way, this iterative process is repeated throughout the entire image, moving from left to right and moving down one unit when reaching an edge. Once the entire image has been traversed, the complete activation matrix is obtained, which contains the characteristics sought in the image for each filter.

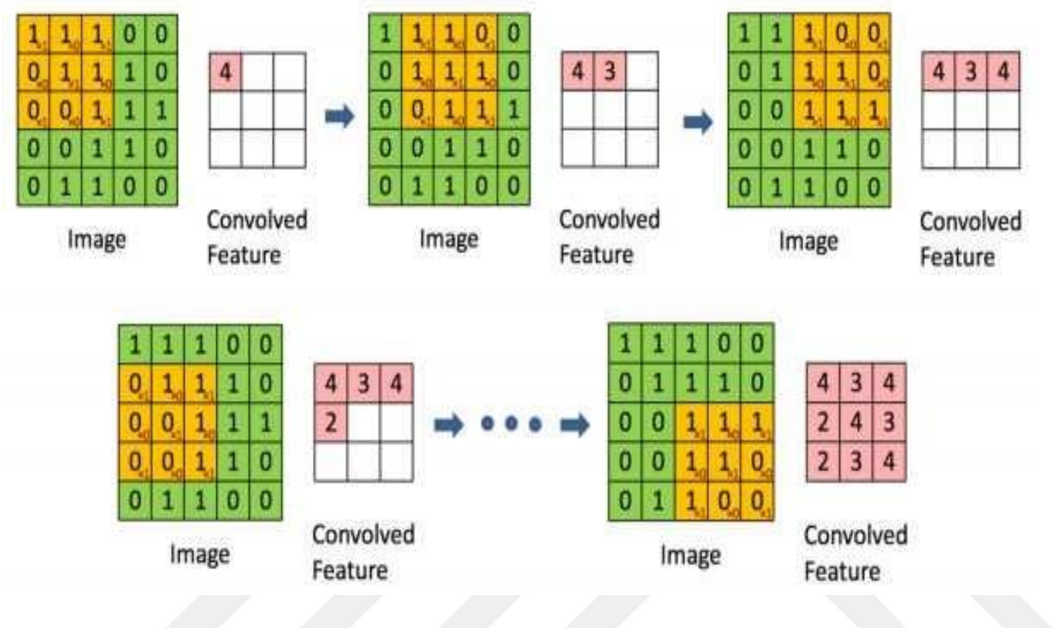


Figure 3.9 CNN features extracted

Following the scheme in Figure 4-1, immediately after the convolution layer is the pooling layer. After you have obtained the characteristics in the convolution layer, the next step in the list is to use them for image classification. In theory, any type of classifier could already be used to carry out this task, however, this process can still be very computationally expensive. For example, the following is considered: if the input image is D2 D2 pixels, at the output of the convolutional layer, the number of elements will be D2> q D2> =, D and, if they are, by example 400 traits the resulting vector is D 000

), 2,000 features. A network with this huge number of characteristics will behave very slowly and, surely, over-adjustment errors will occur during learning.

As we saw in section 4.3, the images have the property of being stationary and a characteristics map has been created in the convolutional layer. Thus, the next natural step is to see in which area of the image those predominant features are. To perform this task, it is possible to calculate the average or find the maximum value of a characteristic throughout a region of the image. The matrix resulting from the previous operation results in considerably smaller dimensions than the matrix of characteristics, obtained in the convolution layer.

So, the objective of this layer is to further decrease the computational load of the system and, at the same time, help with the characterization of the image obtaining and locating the predominant features in it. Furthermore, as mentioned in the paragraph, it is important to highlight that there are two types of pooling: mean pooling or average-pooling and max-pooling. Figure 4-6 shows the effects of applying one type of pooling or another.

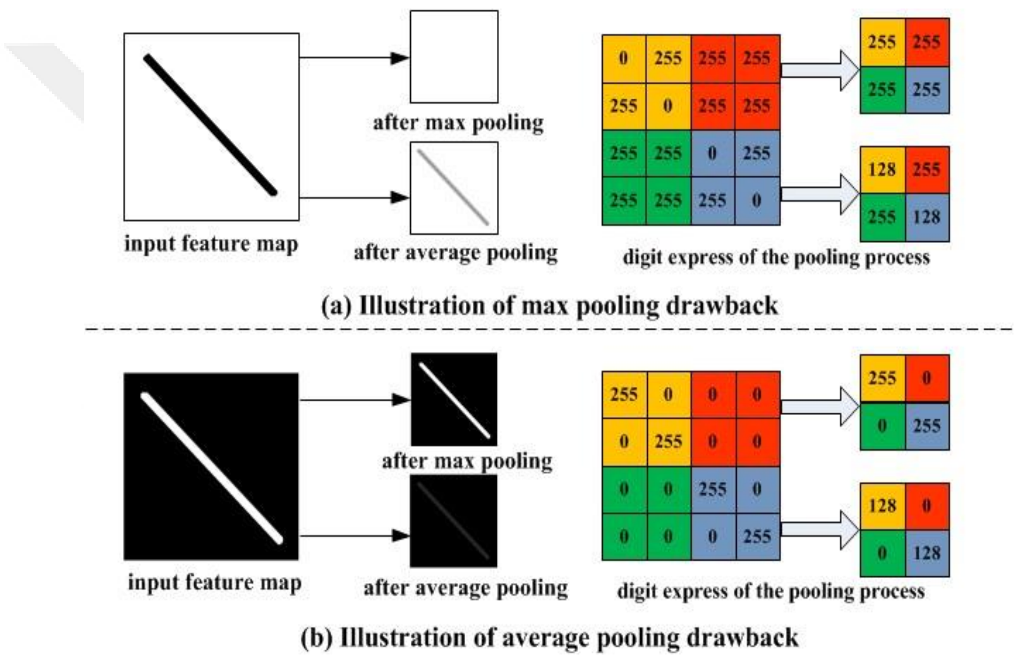


Figure 3.10 Max pooling in CNN

This is the last layer of the convolutional neural network scheme and it is a classifier that determines which class the input image belongs to, that is, your job in this project is to indicate what number the network is 'creating'. has provided you with the input.

The fully connected layer is just after the last pooling layer. It is made up of a number of neurons equal to the number of classes, in the case of this project there are 10 neurons (there are 10 classes: the numbers from 0 to 9). The scheme that follows this layer is the one explained in section 2, when we spoke of fully connected neural networks. That is, each of these neurons is in turn connected to each and every one of the elements of the matrices of the immediately preceding layer.

At the exit of this layer you will find a vector of 10 components. Each of these components represents the probability that the input image has of belonging to a certain class, which is the same, the probability that it has of being one number or another. Finally, the network will determine which of these elements of the vector is greater and indicates this.

3.3 SUPPORT VECTOR MACHINE SVM

The Vector Support Machines (SVMs) classification-regression method was developed in the 1990s, within the field of computer science. Although originally selected as a binary classification method, its application has been extended to multiple classification and regression problems. SVMs have turned out to be one of the best classifiers for a wide range of situations, which is why it is considered one of the benchmarks in the field of statistical learning and machine learning.

Support Vector Machines are based on the Maximal Margin Classifier, which in turn is based on the hyperplane concept. Throughout this essay each of these concepts is introduced in order. Understanding the fundamentals of SVMs requires a solid understanding of linear algebra. This essay does not delve into the mathematical aspect, but you can find a specific description in the book Support of vector machines succinctly by Alexandre Kowalczyk

In R, the e1071 and LiblineaR libraries contain the algorithms necessary to obtain simple, multiple and regression classification models, specific to Support Vector Machines. In a p-dimensional space, a hyperplane is defined as a flat and related subspace of dimensions' $p - 1$. The related term means that the subspace does not have to go through the origin. In a two-dimensional space, the hyperplane is a 1-dimensional subspace, that is, a line. In three-dimensional space, a hyperplane is a two-dimensional subspace, a conventional plane. For dimensions $p > 3$ it is not intuitive to visualize a hyperplane, but the concept of subspace with $p - 1$ dimensions remains.

The mathematical definition of a hyperplane is quite simple. In the case of two dimensions, the hyperplane is described according to the equation of a line:

$$\beta 0 + \beta 1 x 1 + \beta 2 x 2 = 0$$

Given the parameters β0, β1 and β2, all pairs of values x = (x1, x2) for which equality is true are points on the hyperplane. This equation can be generalized for p-dimensions:

$$\beta 0 + \beta 1x1 + \beta 2x2 + \ldots + \beta pxp = 0$$

and in the same way, all the points defined by the vector (x = x1, x2, ..., xp) that satisfy the equation belong to the hyperplane.

When x does not satisfy the equation:

$$\beta 0 + \beta 1x1 + \beta 2x2 + \ldots + \beta pxp < 0$$

$$\beta 0 + \beta 1x1 + \beta 2x2 + \ldots + \beta pxp > 0$$

point x falls to one side or the other of the hyperplane. Thus, it can be understood that a hyperplane divides a p-dimensional space into two halves. To find out on which side of the hyperplane a certain point x is, you just have to calculate the sign of the equation.

The following image shows the hyperplane of a two-dimensional space. The equation describing the hyperplane (a line) is 1 + 2x1 + 3x2 = 0. The blue region represents the space where all the points for which 1 + 2x1 + 3x2> 0 are and the red region that of the points for which 1 + 2x1 + 3x2 <0. When n observations are available, each with p predictors and whose response variable has two levels (hereinafter identified as +1 and −1), hyperplanes can be used to construct a classifier that allows predicting which group an observation belongs to. based on their predictors. This same problem can also be addressed with other methods (logistic regression, LDA, classification trees ...) each with advantages and disadvantages.

To facilitate understanding, the following explanations are based on a two-dimensional space, where a hyperplane is a line. However, the same concepts are applicable to higher dimensions.

If the distribution of the observations is such that they can be perfectly linearly separated into the two classes (+1 and −1), then a hyperplane of separation fulfills that:

$$\beta 0 + \beta 1x1 + \beta 2x2 + \ldots + \beta pxp > 0, if\ yi = 1$$

$$\beta 0 + \beta 1x1 + \beta 2x2 + \ldots + \beta pxp < 0, if\ yi = -1$$

By identifying each class as +1 or −1, and since multiplying two negative values results in a positive value, the above two conditions can be simplified into one:

$$yi\ (\beta 0\ +\ \beta 1x1\ +\ \beta 2x2 + \ldots + \beta pxp) > \ 0, for\ i\ =\ 1 \ldots n$$

Under this scenario, the simplest classifier is to assign each observation to a class depending on which side of the hyperplane it is on. That is, the observation x ^ * is classified according to the sign of the function f (x ^ *) = β0 + β1x ∗ 1 + β2x ∗ 2 + ... + βpx ∗ p. If f (x ∗) is positive, the observation is assigned to class +1, if negative, to class −1. Furthermore, the magnitude of f (x ∗) allows knowing how far away the observation of the hyperplane is and with it the confidence of the classification.

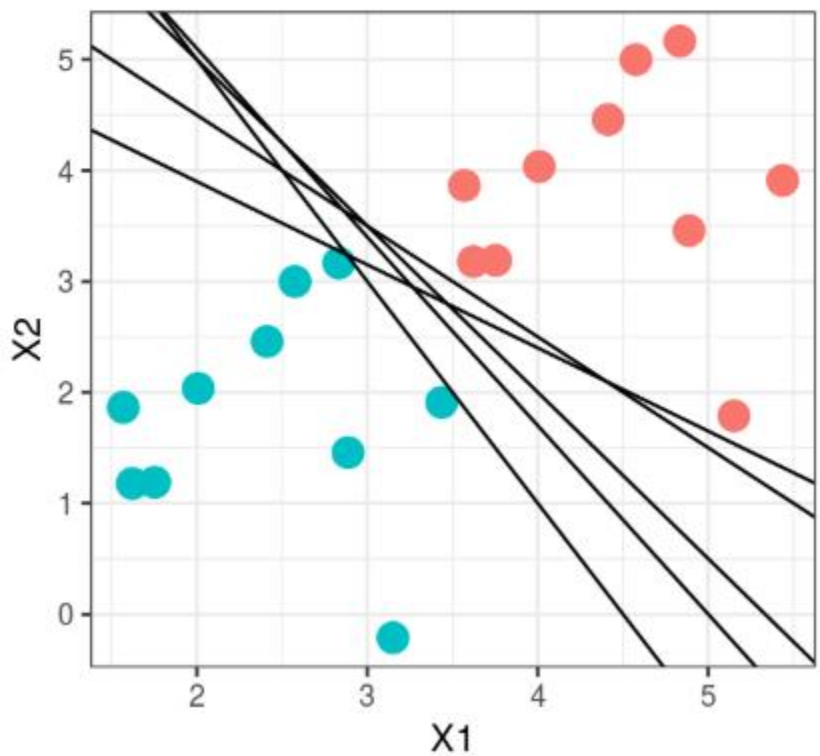


Figure 3.11 Linear classification

The definition of hyperplane for perfectly linearly separable cases results in an infinite number of possible hyperplanes, which requires a method that allows selecting one of them as the optimal classifier. The solution to this problem is to select as the optimal classifier what is known as the maximal margin hyperplane or optimal hyperplane of separation, which corresponds to the hyperplane that is furthest from all training observations. To obtain it, the perpendicular distance of each observation to a certain hyperplane must be calculated. The smallest of these distances (known as the margin) determines how far away the hyperplane of the training observations is.

The maximal margin hyperplane is defined as the hyperplane that achieves the greatest margin, that is, the minimum distance between the hyperplane and the observations is as large as possible. Although this idea sounds reasonable, it is not possible to apply it, since there would be infinite hyperplanes against which to measure distances. Instead, optimization methods are used. To find a more detailed description of the optimization solution, consult (Support Vector Machines Succinctly by Alexandre Kowalczyk).

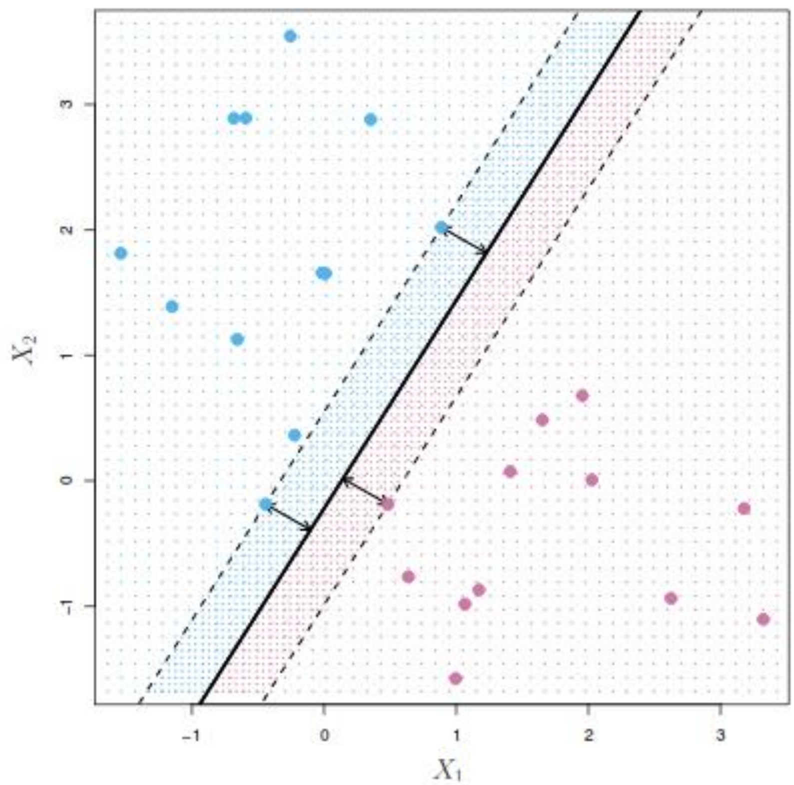


Figure 3.12 SVM classification

The image above shows the maximal margin hyperplane for a training dataset. The three equidistant observations with respect to the maximal margin hyperplane are found along the dashed lines indicating the width of the margin. These observations are known as support vectors, since they are vectors in a p-dimensional space and support (define) the maximal margin hyperplane. Any modification in these observations (support vectors) entails changes in the maximal margin hyperplane. However, modifications in observations that are not a support vector have no impact on the hyperplane.

The Maximal Margin Classifier described in the previous section has little practical application, since there are seldom cases where the classes are perfectly and linearly separable. In fact, even when these ideal conditions are met, in which there is a hyperplane capable of perfectly separating the observations into two classes, this approach still has two drawbacks:

Since the hyperplane has to perfectly separate the observations, it is very sensitive to variations in the data. Including a new observation can lead to very large changes in the separation hyperplane (low robustness).

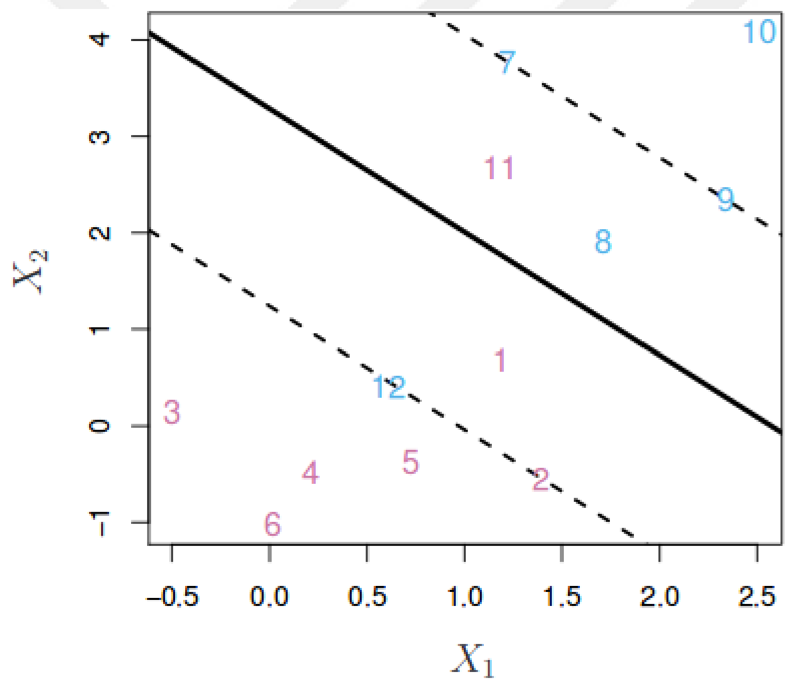


Figure 3.13 Feature classification using SVM

Fitting the maximal margin hyperplane perfectly to training observations to separate them all correctly often leads to overfitting issues.

For these reasons, it is preferable to create a classifier based on a hyperplane that, although it does not perfectly separate the two classes, is more robust and has greater predictive capacity when applied to new observations (fewer overfitting problems). This is exactly what support vector classifiers, also known as soft margin classifiers or Support Vector Classifiers, achieve. To do this, instead of looking for the widest possible sorting margin, you get the observations to be on the right side of the margin; Certain observations are allowed to be on the wrong side of the margin or even the hyperplane.

The following image shows a support vector classifier fitted to a small set of observations. The solid line represents the hyperplane and the dashed lines represent the margin on each side. Observations 2, 3, 4, 5, 6, 7 and 10 are on the correct side of the margin (also hyperplane) and are therefore well classified. Observations 1 and 8, even though they are within the margin, are on the correct side of the hyperplane, so they are also well classified. Observations 11 and 12 are on the wrong side of the hyperplane, their classification is incorrect. All those observations that, being inside or outside the margin, are on the wrong side of the hyperplane, correspond to poorly classified training observations.

Identifying the hyperplane of a support vector classifier, which correctly classifies most of the observations except for a few, is a problem of convex optimization. Although the mathematical proof is out of the scope of this introduction, it is important to mention that the process includes a tuning hyper parameter C. C controls the number and severity of margin violations (and hyperplane) that are tolerated in the adjustment process. If $C = \infty$, no margin violation is allowed and therefore the result is equivalent to the Maximal Margin Classifier (keeping in mind that this solution is only possible if the classes are perfectly separable). The closer C gets to zero, the fewer errors are penalized and the more observations can be on the wrong side of the margin or even the hyperplane. C is ultimately the hyper parameter in charge of controlling the balance between bias and variance of the model. In practice, its optimal value is identified by cross-validation.

The optimization process has the peculiarity that only the observations that are just in the margin or that violate it influence the hyperplane. These observations are known as support vectors and are those that define the classifier obtained. This is why parameter C controls the balance between bias and variance. When the value of C is small, the margin is wider, and more observations violate the margin, becoming support vectors. The hyperplane is, therefore, supported by more observations, which increases the bias but reduces the variance. The greater the value of C, the smaller the margin, the fewer observations will be support vectors and the resulting classifier will have less bias but greater variance.

Another important property that derives from the hyperplane being dependent only on a small proportion of observations (support vectors), is its robustness compared to observations very far from the hyperplane. This makes the vector classification method different from other methods such as Linear Discriminant Analysis (LDA), where the classification rule depends on the mean of all the observations.

The Support Vector Classifier described in the previous sections achieves good results when the separation limit between classes is approximately linear. If it is not, its capacity drops dramatically. One strategy to deal with scenarios in which the separation of the groups is of a non-linear type is to expand the dimensions of the original space.

The fact that the groups are not linearly separable in the original space does not mean that they are not in a larger space. The following images show how two groups, whose two-dimensional separation is not linear, are linear when adding a third dimension.

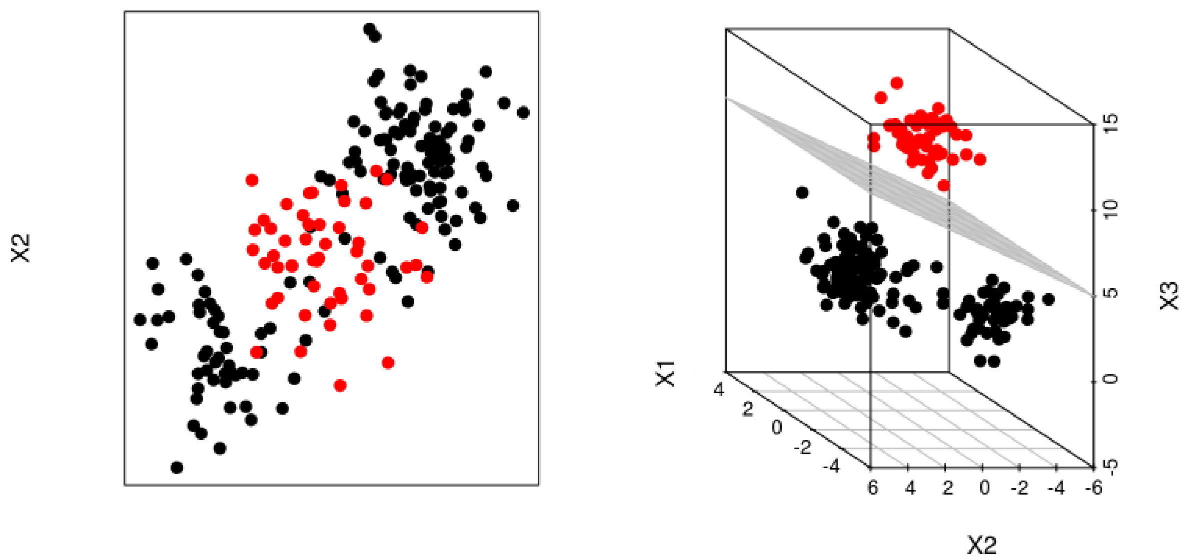


Figure 3.14 3D view of the SVM features

PROPOSED METHOD

For the training of a convolutional neural network it is important to capture information that adequately represents the pattern that is trying to be recognized for the creation of characteristic vectors. In the case of the current project, it was decided to use three different characteristic vector schemes for the generation of databases for the training of neural networks that can adequately represent images, the three schemes used are listed below:

•Image

•Discrete Cosine transform

• Discrete wavelet transform

Once the different schemes for the characteristic vectors were chosen, a treatment was carried out to adapt them to the training of neural networks. It is necessary that the information used to train the convolutional neural networks is representative of the sample that is trying to be recognized and that the information administered to it is the most important. To achieve this, normalization and principal component analysis were used, which means that the information used for neural network training is not correlated and that there is no information that has more weight for network training.as shown in the diagram below:

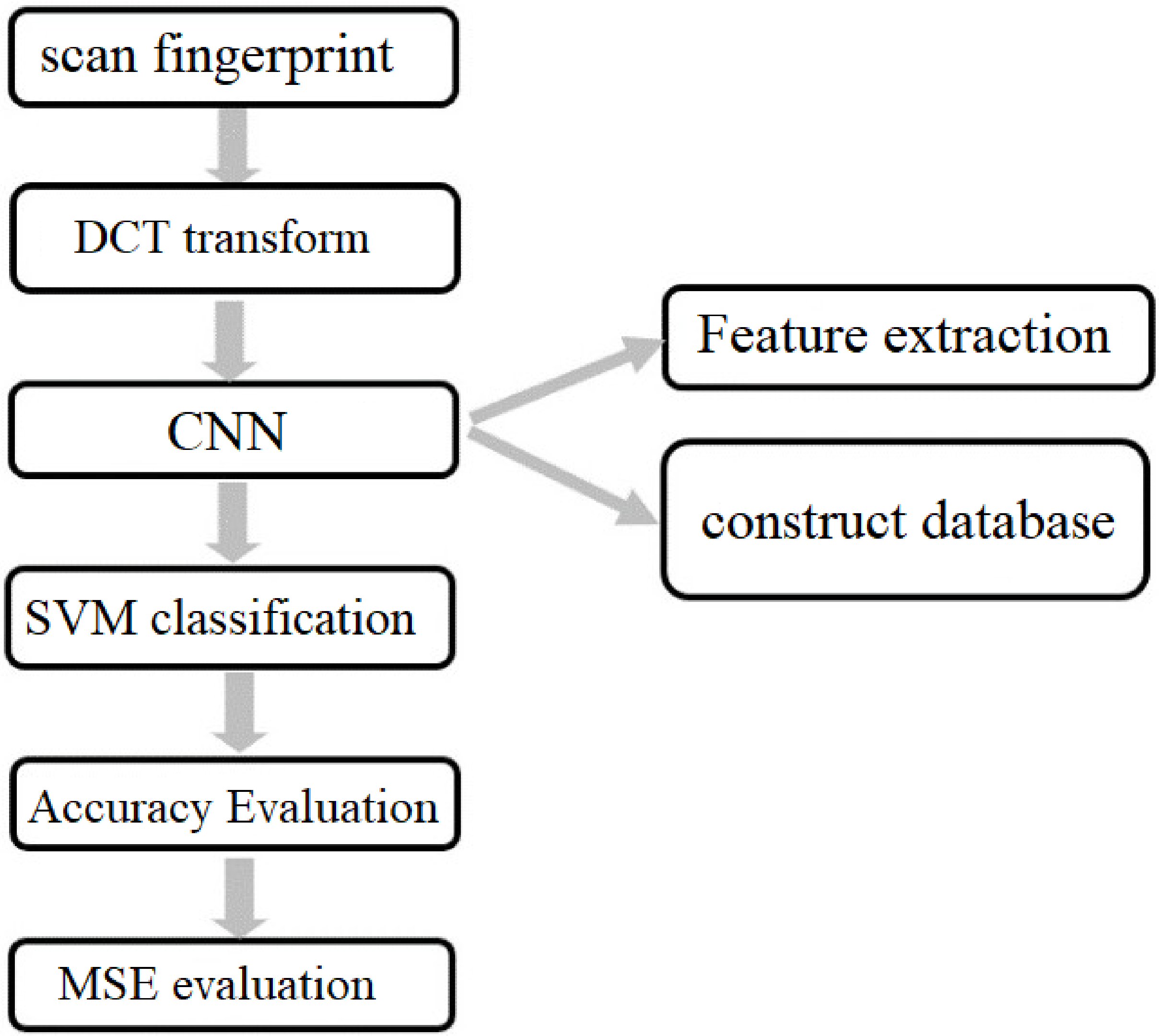


Figure 4.1 Proposed SVM+CNN method

The original image is 416X256 pixels. Therefore, the DCT matrix is 416x25 coefficients (106496 coefficients). By carrying out the previously described process, 101189 DCT coefficients were eliminated, which are approximately 95% of the coefficients. In other words, with 5% of the DCT coefficients of the image, an image of reasonable quality can be reconstructed. To get an idea of which coefficients were eliminated, the resulting matrix was graphed with the DCT coefficients eliminated in gray scale. Keep in mind that in a grayscale image 0 is black (see figures 21 and 22). A greater importance of the low frequency coefficients can be noticed (upper left of the matrix)

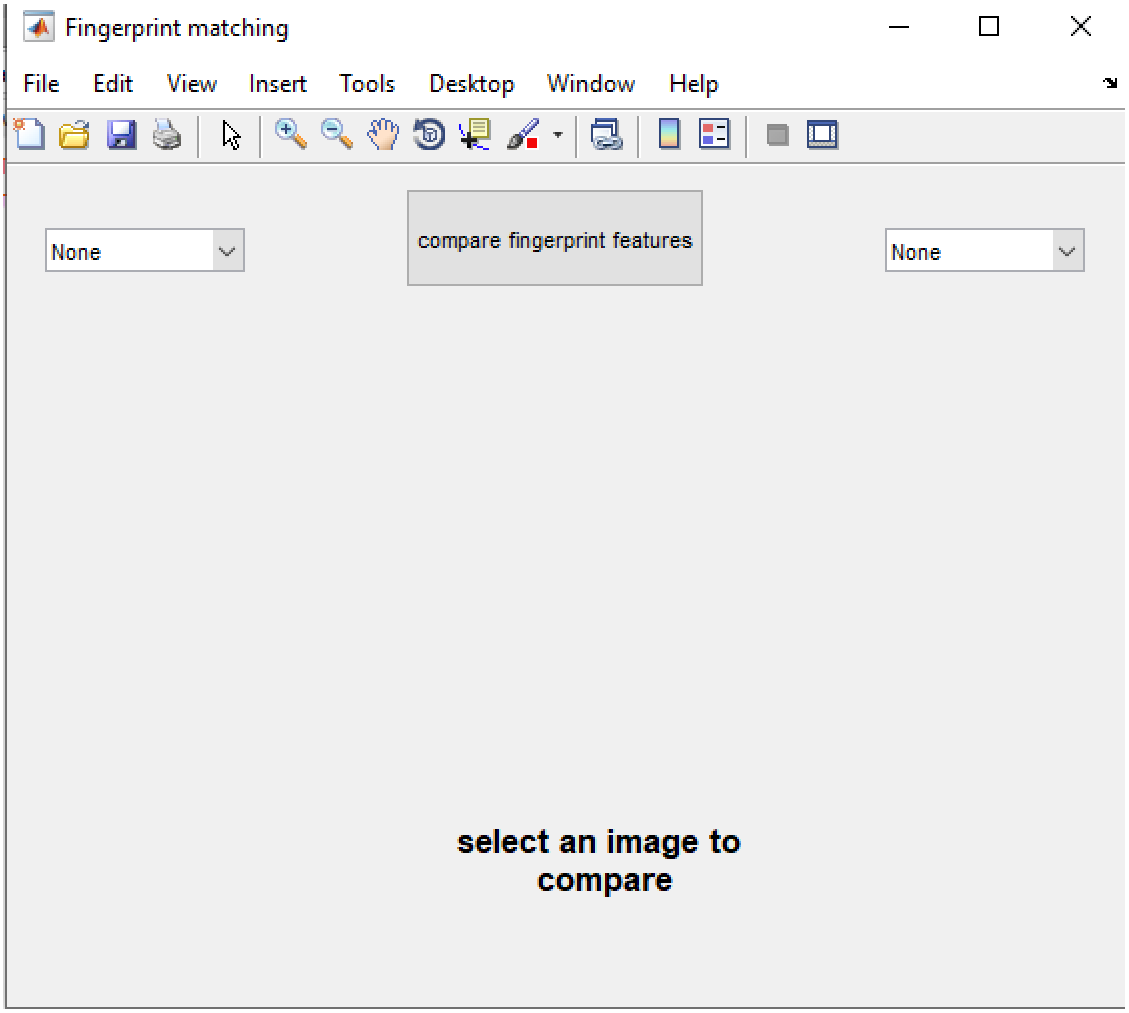


Figure 4.2 Select an image

Given two representations (the test and the mold), the "matching" module determines if the sample comes from the same footprint. The matching phase usually defines a measure of similarity between two tracks. Also in this phase, a threshold is defined to decide whether two samples belong to the same finger or not.

To make the decision if two fingerprints come from the same finger, they should be aligned beforehand to make the decision. In addition, it must be taken into account that the dirt that interferes with the fingerprinting process, finger injuries, distortion introduced by the pressure exerted when taking the fingerprint, among other factors, make the alignment process complex.

To carry out the alignment, several techniques are used; Optimization of the correlation of the images, alignment of ridges, and in the recognition with minutiae, an attempt has been made to align a certain number of them. The literature indicates that the alignment of three minutiae is adequate.

Once the samples have been aligned, the aligned structures provide a basis for determining the hit score. Depending on the type of alignment that has been used, different types of methods are used to find the score, for example, the correlation coefficient is used, or algorithms that try to take into account the added distortion when taking the footprint due to the pressure exerted.

There are several methods for scoring similarity of two tracks. Below is a schematic where one of these methods is used. Mainly a minutia is taken as a reference and the distance between the reference minutia and the other minutiae of the reference footprint is measured. Then, after the two footprints are aligned, the footprint to be compared is entered and the distances from the reference minutia are measured. An attempt is made to reproduce the reference footprint and each correction that must be made to adjust it has a penalty. The fewer modifications that must be made to make them the same, the lesser the penalty. A score can be obtained from the sum of the penalties. The higher the score (less penalties), the greater the probability that the entered fingerprint will come from the same finger as the reference fingerprint. If the score obtained is outside a threshold, the fingerprint will not belong to the individual. Below is the comparison of two footprints

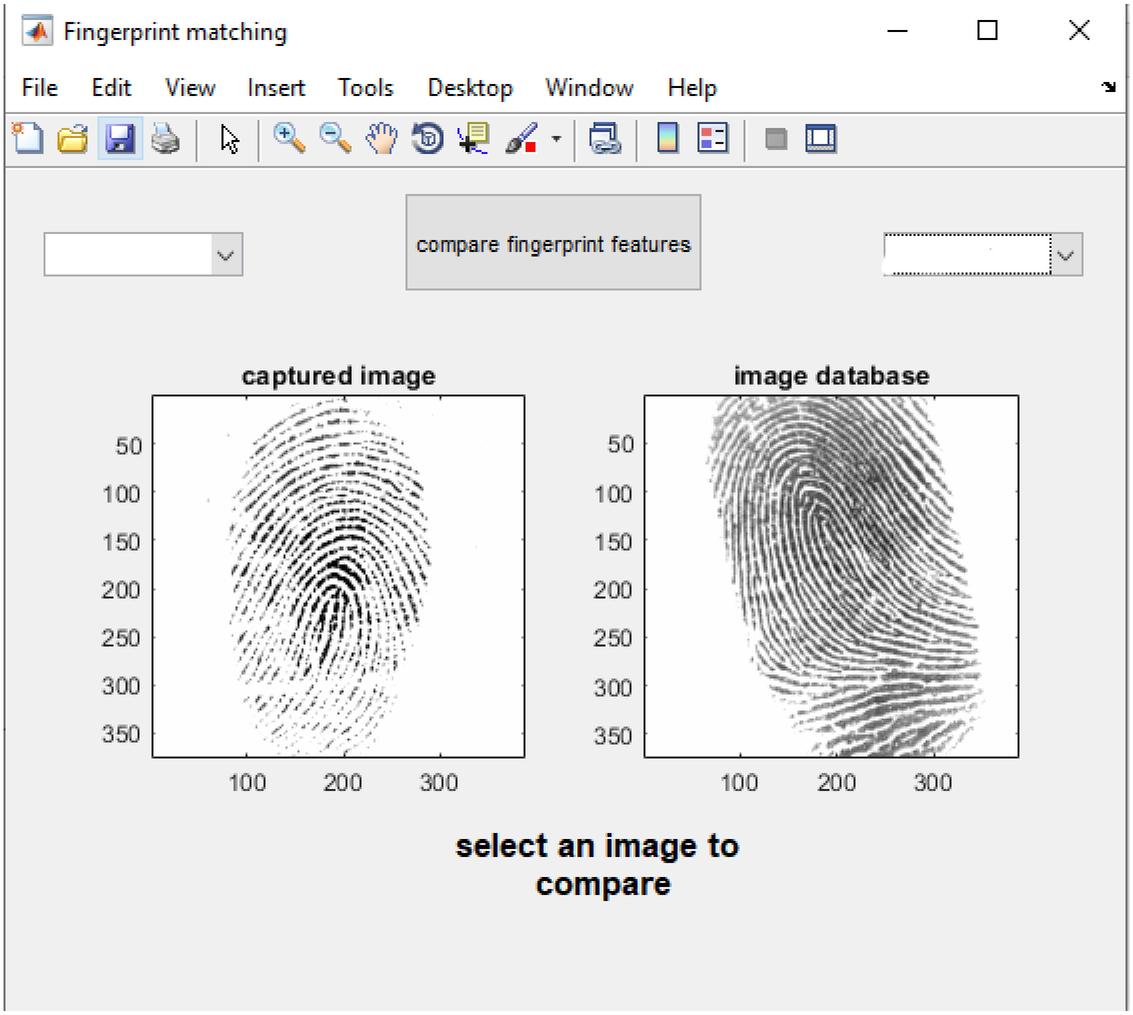


Figure 4.3 Displaying selected images

The extraction method consists mainly of three phases:

1. Estimation of the orientation of the ridges

2. Extraction of ridges

3. Extraction of minutiae and post processing

Gradient methods are used to find the orientation of the ridges. To make use of these methods, the image is divided into a large grid and the gradient of each of the small grids is obtained. Obtaining the direction of greatest change, the orientation can be determined.

The segmentation of the image seeks to eliminate the information that is not relevant to give information about the footprint, for example, grime, remains of a previous acquisition, etc. One of the methods used is the certainty level, which consists of finding in an orientation box how much the orientation of the pixel gradient agrees with the orientation of the block. If the level of certainty is below a certain threshold, the block is marked as not relevant information. This method works very well to determine areas of interest.

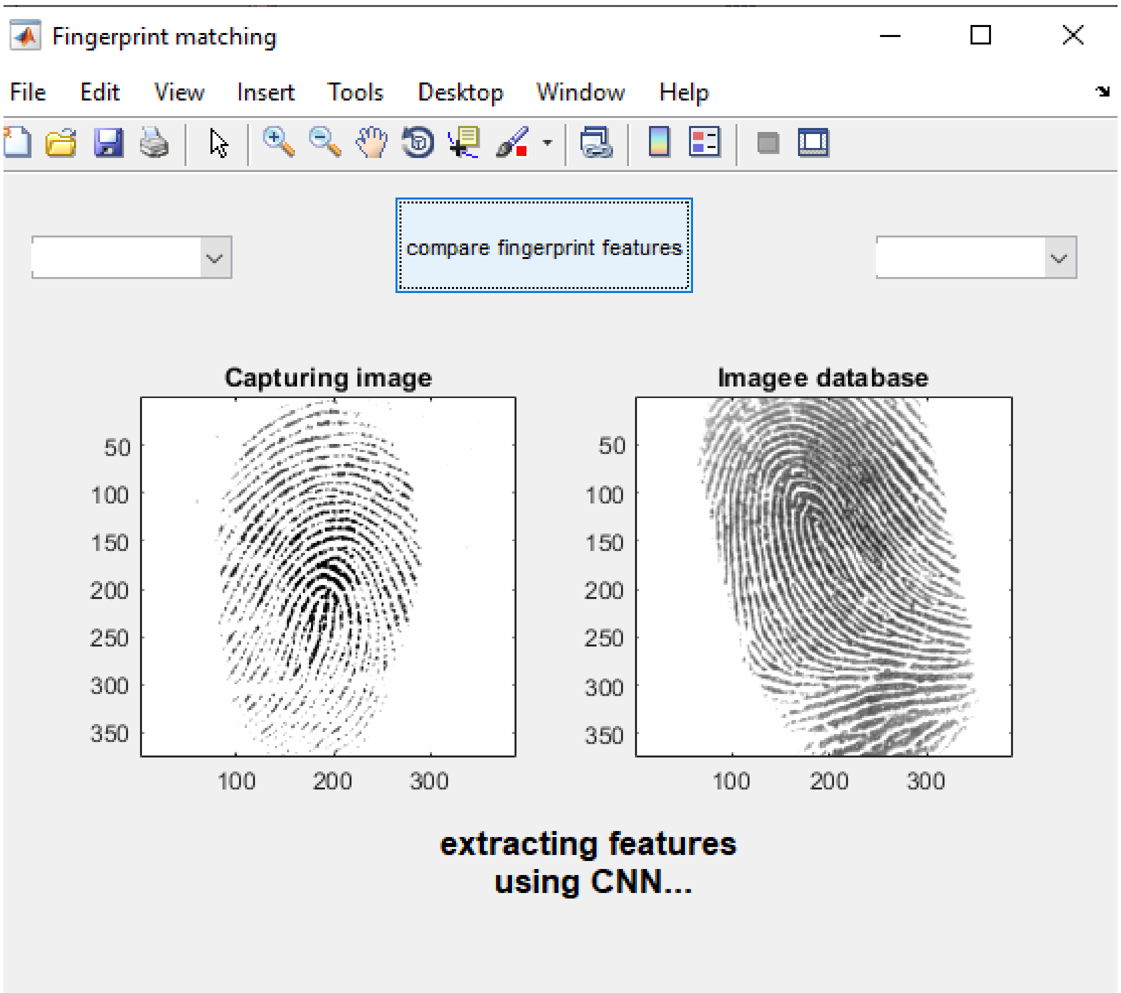


Figure 4.4 Extracting features of the selected images

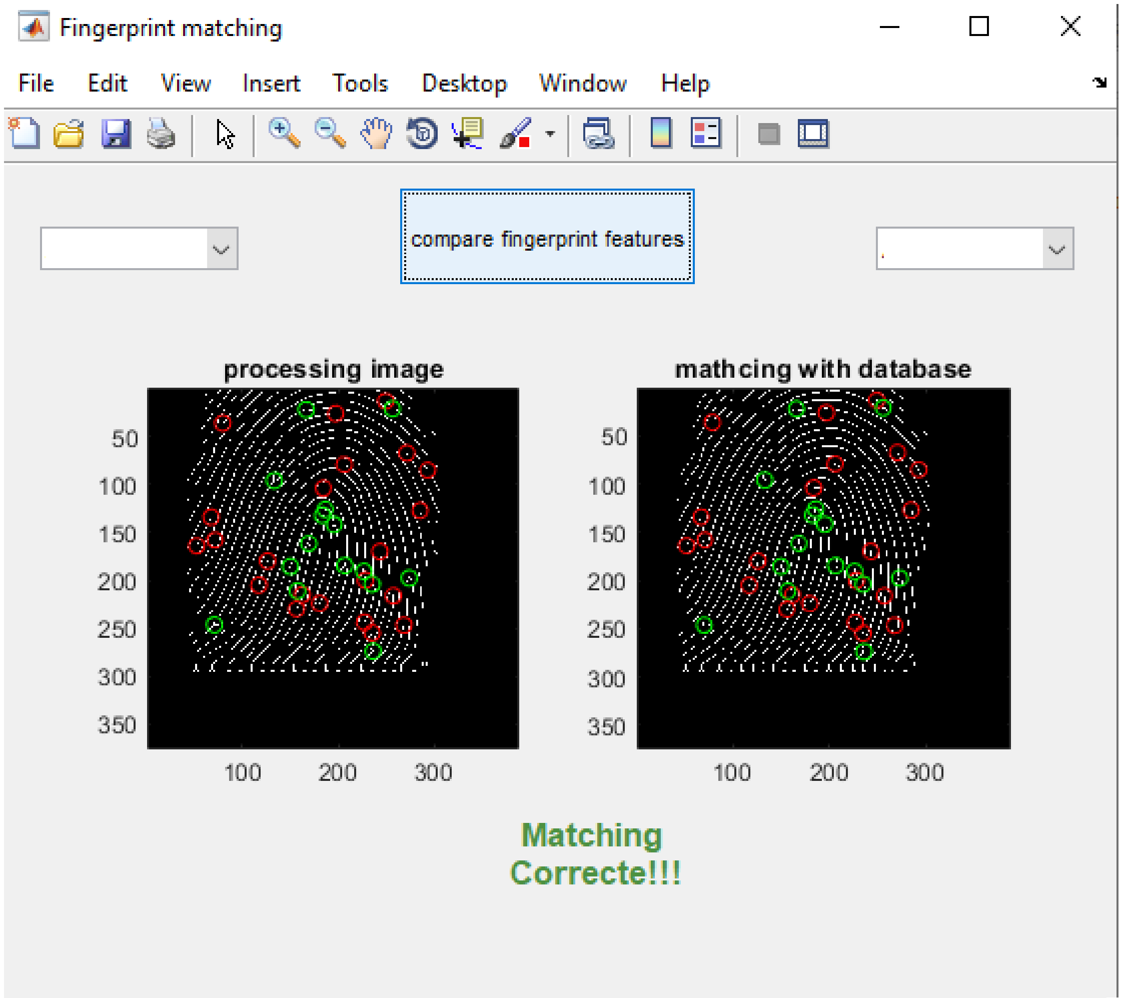


Figure 4.5 matching features of the selected images

Below are the results obtained from training the different learning algorithms by varying the number of neurons in the intermediate layer. Figures below show the result of training the specimens using convolutional neural networks with each of the 100 characteristic vectors, the results are evaluated using the mean square error index MSE and the Accuracy index:

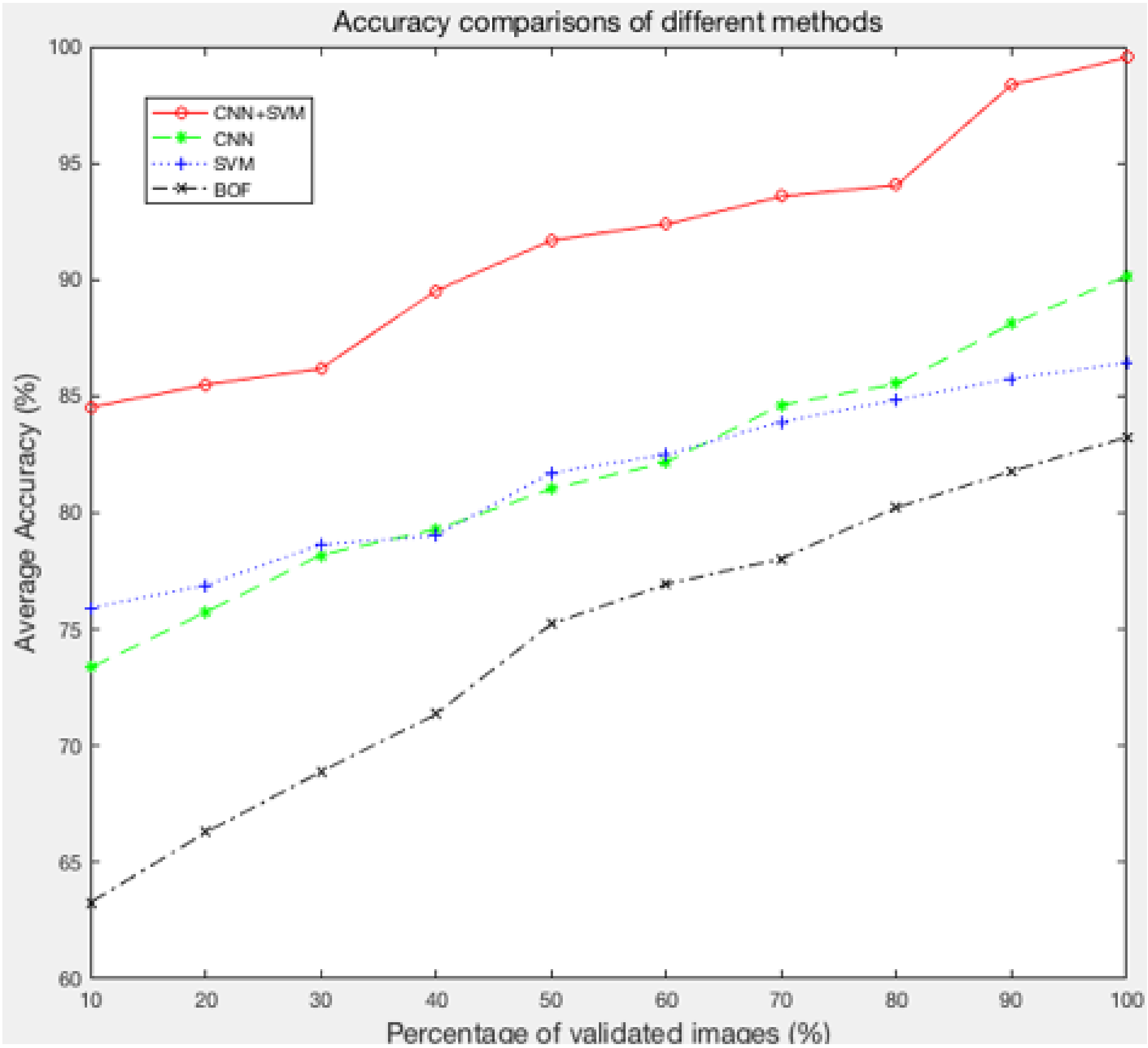


Figure 4.6 Accuracy of the proposed method

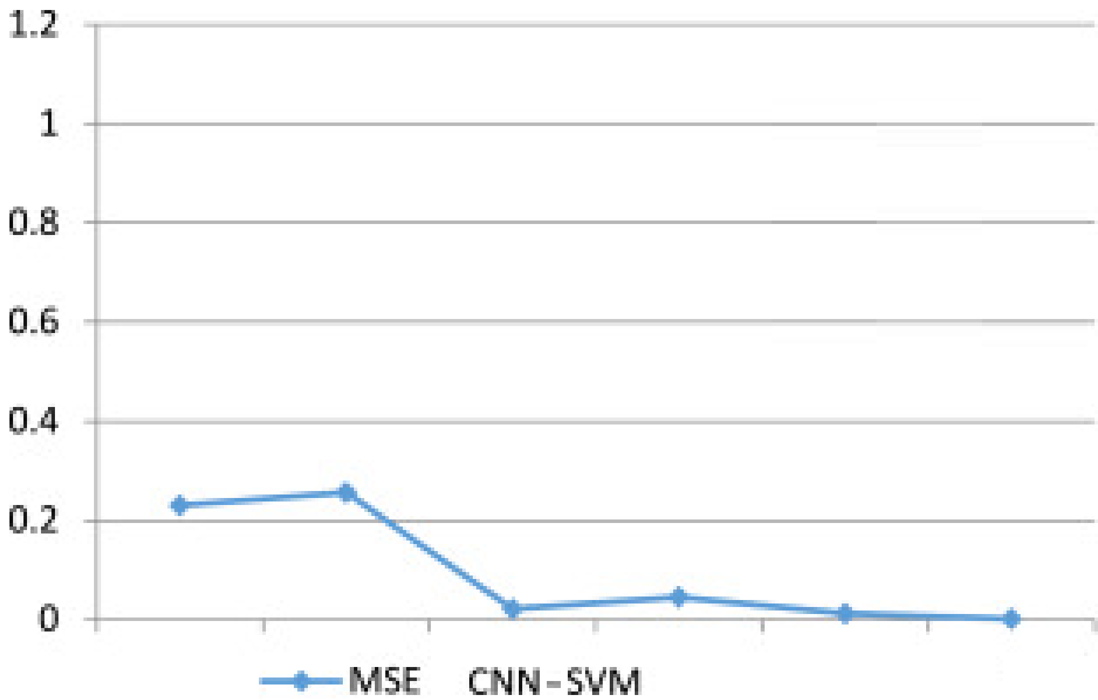


Figure 4.7 MSE of the proposed method

CONCLUSION

The best results were obtained with the discrete cosine transform database in both learning algorithms. We assume that these results are derived from the generation of a “low pass filter” that is created by using the upper triangle of the coefficient matrix of the cosine transform. Despite using all the coefficients of the cosine transform, the best results are not obtained.

discrete cosine transform may indicate that the input image to the networks is not properly filtered. The results obtained using the image database are very good considering that no transforms are required, it only takes into account a spatial relationship between the pixels of the image. Furthermore, the characteristic vector using the complete image obtained is the smallest one.

The proposed methodology demonstrates the feasibility of recognizing minutiae with the use of convolutional neural networks and an easily achievable hardware implementation.

The use of support vector machine classification to increase the bases can give the network robustness by introducing different patterns. However, it should be verified that the results obtained are good despite the use of the deformation algorithm. There are existing databases that have the details of minutiae and non-minutiae marked by fingerprint experts (GT, ANSI / NIST databases, etc.) [2], [15]. Conducting training with databases of this type could help to solve this question.

From the recognition of minutiae, it is possible to carry out the recognition of the fingerprint by comparing the spatial relationship of the minutiae obtained and comparing this information with a previously stored database.

Given the nature of the application where the fingerprint is acquired through a controlled process, the option of performing fingerprint recognition with OCR techniques from the non-minutia minutia ratio on the grid is proposed.

In the traditional techniques of minutia extraction, digital image treatment is used, where the fingerprint is binarized, thinned and the minutiae are recognized [2]. Our implementation obviates the binarization and thinning of the image, obtaining good results. The improvement in performance could be evaluated if the binarized and thinned image is used for the training of the neural networks exposed in this work.

To improve the performance of fingerprint recognition, it is possible to think about the integration of methods that are based on the determination of global characteristics of the fingerprint and the spatial relationship of the same generated by the recognition of minutiae [15].

REFERENCES

[1] N. Lavanya and T. Shankar, "Energy optimization in wireless sensor network using NSGA-II," *ARPN J. Eng. Appl. Sci.*, vol. 12, no. 23, pp. 6698–6704, 2017.

[2] J. Feng, J. Zhou, and A. K. Jain, "Orientation field estimation for latent fingerprint enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 925–940, 2013, doi: 10.1109/TPAMI.2012.155.

[3] G. Munussami, S. Sokalingam, S. Edwardraja, J. R. Kim, S. Chung, and S. G. Lee, "Computational screening of potential non-immunoglobulin scaffolds using overlapped conserved residues (OCR)-based fingerprints," *Korean J. Chem. Eng.*, vol. 35, no. 3, pp. 717–724, 2018, doi: 10.1007/s11814-017-0350-4.

[4] Y. Park, U. Jang, and E. C. Lee, "Statistical anti-spoofing method for fingerprint recognition," *Soft Comput.*, vol. 22, no. 13, pp. 4175–4184, 2018, doi: 10.1007/s00500-017-2707-3.

[5] Aravindan and S. M. Anzar, "Robust partial fingerprint recognition using wavelet SIFT descriptors," *Pattern Anal. Appl.*, vol. 20, no. 4, pp. 963–979, 2017, doi: 10.1007/s10044-017-0615-x.

[6] Y. L. Chen, C. M. Liu, C. W. Yu, and Y. P. Hsu, "Efficient overlapping fingerprint feature identification based on a multiple filter set and support vector machine classifiers," *J. Imaging Sci. Technol.*, vol. 62, no. 3, pp. 1–7, 2018, doi: 10.2352/J.ImagingSci.Technol.2018.62.3.030404.

[7] S. Edwardraja, G. Munussami, A. Goyal, and S. G. Lee, "Generation of efficient fingerprint for GFP-like fold and computational identification of potential GFP-like homologs," *Biotechnol. Bioprocess Eng.*, vol. 21, no. 6, pp. 712–719, 2016, doi: 10.1007/s12257-016-0362-7.

[8] B. Stojanović, A. Nešković, and O. Marques, "A novel neural network based approach to latent overlapped fingerprints separation," *Multimed. Tools Appl.*, vol. 76, no. 10, pp. 12775–12799, 2017, doi: 10.1007/s11042-016-3696-4.

[9] J. H. Yang and J. J. Yoh, "Reconstruction of chemical fingerprints from an individual's time-delayed, overlapped fingerprints via laser-induced breakdown spectrometry (LIBS)

and Raman spectroscopy," *Microchem. J.*, vol. 139, no. 2017, pp. 386–393, 2018, doi: 10.1016/j.microc.2018.03.027.

[10]B. Stojanović, O. Marques, and A. Nešković, "Latent overlapped fingerprint separation: a review," *Multimed. Tools Appl.*, vol. 76, no. 15, pp. 16263–16290, 2017, doi: 10.1007/s11042-016-3908-y.

[11]M. A. Medina-Pérez, A. M. Moreno, M. Á. Ferrer Ballester, M. García-Borroto, O. Loyola-González, and L. Altamirano-Robles, "Latent fingerprint identification using deformable minutiae clustering," *Neurocomputing*, vol. 175, pp. 851–865, 2016, doi: 10.1016/j.neucom.2015.05.130.

[12]B. Stojanovic, O. Marques, and A. Neškovic, "Deep learning-based approach to latent overlapped fingerprints mask segmentation," *IET Image Process.*, vol. 12, no. 11, pp. 1934–1942, 2018, doi: 10.1049/iet-ipr.2017.1227.

[13]K. Tejas, C. Swathi, D. Aravind Kumar, and R. Muthu, "Automated region masking of latent overlapped fingerprints," *2017 Innov. Power Adv. Comput. Technol. i-PACT 2017*, vol. 2017-Janua, pp. 1–6, 2017, doi: 10.1109/IPACT.2017.8245111.

[14]R. Venkatesh, N. U. Maheswari, and S. Jeyanthi, "Multiple Criteria Decision Analysis Based Overlapped Latent Fingerprint Recognition System Using fuzzy Sets," *Int. J. Fuzzy Syst.*, vol. 20, no. 6, pp. 2016–2042, 2018, doi: 10.1007/s40815-018-0487-2.

[15]J. H. Yang, S. J. Choi, and J. J. Yoh, "Towards reconstruction of overlapping fingerprints using plasma spectroscopy," *Spectrochim. Acta - Part B At. Spectrosc.*, vol. 134, pp. 25–32, 2017, doi: 10.1016/j.sab.2017.06.001.

[16]J. H. Yang and J. J. Yoh, "Forensic Discrimination of Latent Fingerprints Using Laser-Induced Breakdown Spectroscopy (LIBS) and Chemometric Approaches," *Appl. Spectrosc.*, vol. 72, no. 7, pp. 1047–1056, 2018, doi: 10.1177/0003702818765183.

[17]K. Sundararajan and D. L. Woodard, "Deep Learning for Biometrics," *Deep Learn. Biometrics A Surv. ACM Comput. Surv*, vol. 51, pp. 3–31, 2017, doi: 10.1007/978-3-319-61657-5.

[18]F. Chen, J. Feng, A. K. Jain, J. Zhou, and J. Zhang, "Separating overlapped fingerprints," *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no. 2, pp. 346–359, 2011, doi: 10.1109/TIFS.2011.2114345.

[19]S. Jeyanthi, N. Uma Maheswari, and R. Venkatesh, “An Efficient Automatic Overlapped Fingerprint Identification and Recognition Using ANFIS Classifier,” *Int. J. Fuzzy Syst.*, vol. 18, no. 3, pp. 478–491, 2016, doi: 10.1007/s40815-015-0053-0.

[20]H. C. Huang, C. T. Hsieh, M. N. Hsiao, and C. H. Yeh, “A study of automatic separation and recognition for overlapped fingerprints,” *Appl. Soft Comput. J.*, vol. 71, pp. 127–140, 2018, doi: 10.1016/j.asoc.2018.06.008.

[21]NIST Special Database 27. Fingerprint Minutiae from Latent and Matching Ten-print Images, http://www.nist.gov/srd/nistsd27.htm

[22]O. Al-Kadi, D. Watson, Texture analysis of aggressive and nonaggressive lung tumor CE CT images. IEEE Trans. Biomed. Eng. 55(7), 1822–1830 (2008)

[23]Arel, D.C. Rose, T.P. Karnowski, Deep machine learning - a new frontier in artificial intelligence research [research frontier]. IEEE Comput. Intell. Mag. 5(4), 13–18 (2010)

[24]Arshad, G. Raja, A. Khan, Latent fingerprints segmentation: feasibility of using clusteringbased automated approach. Arabian J. Sci. Eng. 39(11), 7933–7944 (2014)

[25]M. Barros Filho, F. Sobreira, Accuracy of lacunarity algorithms in texture classification of high spatial resolution images from urban areas, in XXI Congress of International Society of Photogrammetry and Remote Sensing (2008)

[26]M.A. Carreira-Perpinan, G. Hinton, On contrastive divergence learning, in AISTATS, vol. 10 (Citeseer, 2005), pp. 33–40

[27]H. Choi, A.I.B.M. Boaventura, A. Jain, Automatic segmentation of latent fingerprints, in 2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS) (2012), pp. 303–310

[28]D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11, 625–660 (2010)

[29]J. Ezeobiejesi, B. Bhanu, Latent fingerprint image segmentation using fractal dimension features and weighted extreme learning machine ensemble, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2016)

[30]R.-E. Fan, C.-J. Lin, A Study on Threshold Selection for Multi-label Classification. Department of Computer Science (National Taiwan University, 2007), pp. 1–23

[31]G. Hinton. A Practical Guide to Training Restricted Boltzmann Machines, Version 1 (2010)

[32]C. Huang, C. Moraga, A diffusion-neural-network for learning from small samples. Int. J. Approx. Reason. 35(2), 137–161 (2004)

[33]S. Karimi-Ashtiani, C.-C. Kuo, A robust technique for latent fingerprint image segmentation and enhancement, in 15th IEEE International Conference on Image Processing, 2008, ICIP 2008 (2008), pp. 1492–1495

[34]D. Kaye, T. Busey, M. Gische, G. LaPorte, C. Aitken, S. Ballou, L.B. ..., K. Wertheim, Latent print examination and human factors: improving the practice through a systems approach, in NIST Interagency/Internal Report (NISTIR) - 7842 (2012)

[35]J. Keller, S. Chen, R. Crownover, Texture description and segmentation through fractal geometry. Comput. Vis. Graph. Image Process. 45(2), 150–166 (1989)

a Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in Advances in Neural Information Processing Systems (2012), pp. 1097– 1105

[36]M. Lai, Deep learning for medical image segmentation (2015), arXiv:1505.02000

[37]A.D.K.T. Lam, Q. Li, Fractal analysis and multifractal spectra for the images, in 2010 International Symposium on Computer Communication Control and Automation (3CA), vol. 2 (2010), pp. 530–533

[38]P. Lamblin, Y. Bengio, Important gains from supervised fine-tuning of deep architectures on large labeled sets, inNIPS* 2010 Deep Learning and Unsupervised Feature Learning Workshop (2010)

[39]S. Li, J.T. Kwok, Y. Wang, Combination of images with diverse focuses using the spatial frequency. Inf. Fus. 2(3), 169–176 (2001)

[40]. Y. Liu, E. Racah, P.J. Correa, A. Khosrowshahi, D. Lavers, K. Kunkel, M. Wehner, W.D. Collins, Application of deep convolutional neural networks for detecting extreme weather in climate datasets (2016), CoRR abs/1605.01156

[41]Z. Makó, Approximation with diffusion-neural-network, in 6th International Symposium of Hungarian Researchers on Computational Intelligence (2005), pp. 18–19 4 Latent Fingerprint Image Segmentation … 107

[42]. B. Mandelbrot, The Fractal Geometry of Nature. Einaudi paperbacks (Henry Holt and Company, New York, 1983) 24. A. Ng, J. Ngiam, C.Y. Foo,

[43] Y. Song, J. Ji, Y. Wang, C. Liu, *A New Evolutionary Algorithm for Solving Multiobjective Optimization*, Fifth International Conference on Natural Computation (ICNC), 2009.

[44] D. Phan, J. Suzuki, *Boosting Indicator-Based Selection Operators for Evolutionary Multiobjective Optimization Algorithms*, 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2011.

[45] C. Chi, Y. Shiu, "A Multiobjective Evolutionary Algorithm That Diversifies Population by Its Density", *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, 2012.

[46] L. Ke, A. Fialho, S. Kwong, Z. Qingfu, "Adaptive Operator Selection With Bandits for a Multiobjective Evolutionary Algorithm Based on Decomposition", *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, 2014.

[47] K. Parsopoulos, M. Vrahatis, *Multi-Objective Particles Swarm Optimization Approaches*, IGI Global, 2008.

[48] K. Parsopoulos, M. Vrahatis, *Particle swarm optimization method in multiobjective problems*, In Proceedings of the ACM Symposium on Applied Computing, 2002.

[49] U. Baumgartner, C. Magele, W. Renhart, "Pareto optimality and particle swarm optimization", *IEEE Transactions on Magnetics*, vol. 40, no. 2, pp. 1172-1175, 2004.

[50] M. Mahfouf, M. Chen, D. Linkens, "Adaptive weighted particle swarm optimisation for multi-objective optimal design of alloy steels", *Lecture notes in computer science*, vol. 3242, pp. 762-771, 2004.

[51] X. Li, "A non-dominated sorting particle swarm optimizer for multi-objective optimization", *Lecture notes in computer science*, vol. 2723, pp. 37-48 2003.

[52] X. Hu, R. Eberhart, *Multi-objective optimization using dynamic neighborhood particle swarm optimization*, IEEE Congress Evolutionary Compututation, 2002.

[53] X. Hu, R. Eberhart, Y. Shi, *Particle swarm with extended memory for multi-objective optimization*, IEEE Swarm Intelligence Symposium, 2003.

[54] K. Parsopoulos, M. Vrahatis, *On the computation of all global minimizers through particle swarm optimization*, IEEE Transactions on Evolutionary Computation, 2004.

[55] C. Chow, H. Tsui, *Autonomous agent response learning by a multi-species particle swarm optimization*, IEEE Congress on Evolutionary Computation, 2004.

[56] J. Fieldsend, S. Singh, *A multiobjective algorithm based upon particle swarm optimisation, An efficient data structure and turbulence*, In Proceedings of the UK Workshop on Computational Intelligence, 2002.

[57] T. Ray, K. Liew, “A swarm metaphor for multi-objective design optimization”, *Engineering Optimization*, vol. 34. no. 2, pp. 141-153, 2002.

[58] T. Bartz-Beielstein, P. Limbourg, J. Mehnen, K. Schmitt, K. Parsopoulos, M. Vrahatis, *Particle swarm optimizers for Pareto optimization with enhanced archiving techniques*, IEEE Congress on Evolutionary Computation, 2003.

[59] D. Srinivasan, T. Seow, *Particle swarm inspired evolutionary algorithm (PSEA) for multi-objective optimization problem*, IEEE Congress on Evolutionary Computation, 2003.

[60] S. Mostaghim, J. Teich, *Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)*, IEEE Swarm Intelligence Symposium, 2003.

[61] X. Li, “Better spread and convergence: Particle swarm multi-objective optimization using the maximin fitness function”, *Lecture notes in computer science*, vol. 3102, pp. 117-128, 2004.

[62] G. Toscano, C. Coello, “Using clustering techniques to improve the performance of a particle swarm optimizer”, *Lecture notes in computer science*, vol. 3102, pp. 225-237, 2004.

[63] M. Reyes-Sierra, C. Coello, *Online adaptation in multi-objective particle swarm optimization*, IEEE Swarm Intelligence Symposium, 2006.

[64] S. Ho, S. Yang, G. Ni, E. Lo, H. Wong, “A particle swarm optimization-based method for multi-objective design optimizations”, *IEEE Transactions on Magnetics*, vol. 41, no. 5, pp. 1756-1759, 2005.

[65] C. Raquel, P. Naval, *an effecive use of crowding distance in multi-objective particle swarm optimization*, In Proceedings of the GECCO, 2005.