**Chapter 1**

**INTRODUCTION**

Any automatically measurable, robust and distinctive physical characteristic or personal trait that can be used to identify an individual or verify the claimed identity of an individual is called Biometrical Identification or simply Biometrics. It’s a combination of two Greek words: Bios means Life and Metrics means To Measure. Biometrics serves the identification of humans from their personal traits. As a rapidly growing field, it is initially pushed forward by a need for robust security and surveillance applications, but its potential as a natural and effortless means of identification also paved the way for a host of smart applications that automatically identify the user and provide customized services. With increasing awareness of its psychological, privacy-related and ethical aspects, there is no doubt that biometrics will continue to contribute to many technological solutions of our daily lives.

Biometrics is the science of establishing the identity of an individual based on the physical, chemical or behavioral attributes of the person. The relevance of biometrics in modern society has been reinforced by the need for large-scale identity management systems whose functionality relies on the accurate determination of an individual identity in the context of several different applications . Different biometric indicators are suited for different kinds of identification applications due to their variations in intrusiveness, accuracy, cost, and ease of sensing. A new measurement that purports to belong to a particular entity is compared against the data stored in relation to that entity. If the measurements match, the assertion that the person is whom they say they are is regarded as being authenticated. Some building access schemes work in this way, with the system comparing the new measure against the company’s employee database. Also authentication of the identity of a person is frequently used in order to grand access to premises or data.

The technology of biometrics relies on the input from a number of fields, starting with various kinds of sensors that are used to sample the biometric. Signal processing and pattern recognition methods are obviously relevant, as the acquired data need to be prepared for accurate and robust decisions. At its final stage, the system outputs a decision, which links the acquired and processed biometric trait to an identity. Algorithms and mathematical models developed by the machine learning community are frequently used in biometric systems to implement the decision function itself, but this is surely not the only contribution worth mentioning. Machine learning methods are useful in selecting appropriate feature representations that will facilitate the job of the decision function, in dealing with temporal information, and in fusing multimodal information. The goal is to familiarize the machine learning with the problems of biometrics, to show which techniques are employed to solve them, and what challenges are open in the field that may benefit from future machine learning applications. It is also intended to familiarize the biometrics to the methods and ways of machine learning and its correct research methodology, and to provide the rudiments of a toolbox of machine learning.

Security issues seem to be one of the most important problems of contemporary computer science. One of the most important branches of security is identification of users. Identification may be required for access control to buildings, rooms, devices or information. In case of computer systems we say about access to software and data. The basic aim of identification is to make it impossible for unauthorized persons to access to the specified resources. There are generally three solutions for performing secure identification:

* Token Methods
* Memory Based Methods
* Biometric Methods

The token method has two significant drawbacks. Firstly, the token may be lost or stolen. A person who finds or steals a token may have an access to all the resources that the proper owner of the token was able to access, and there is no possibility to find out if they are the person they claim to be. Secondly, the token may be copied. The easiness of making a copy is of course different for different kinds of tokens, but it is always technically possible. Memory based methods identify people by checking their knowledge. The most popular memory methods are of course different kinds of passwords. The main drawback of this kind of methods is the unconscious selectivity of human memory. People may do their best to remember a password but they cannot guarantee that the information will not be forgotten. Similarly to the token method when a malicious user knows a password it is impossible to check if they are the person they claim to be. The problems with token and memory-based methods are the main cause of increasing interest in methods of identification based on biometric information of a person.

Since authentication takes place instantaneously and usually only once, identity fraud is possible. An attacker can bypass the biometrics authentication system and continue undisturbed. A cracked or stolen biometric system presents a difficult problem. Unlike passwords or smart cards, which can be changed or reissued, absent serious medical intervention, a fingerprint or iris is forever. Once an attacker has successfully forged those characteristics, the end user must be excluded from the system entirely, raising the possibility of enormous security risks and/or reimplementation costs. Static physical characteristics can be digitally duplicated. In addition static biometrics could be intolerant of changes in physiology such as daily voice changes or appearance changes.

**1.1 EVALUATION OF BIOMETRIC METHODS**

The main problem of every biometric method is uncertainity of measurement. Every measurement of the same property usually gives different results. The main goal of identification method is to establish algorithms that are able to extract properties of the measurements that are as constant as possible for subsequent trials. These methods focus on creating general models that describe specific properties of human being which may be found in every trial. Of course creating such an exact model is difficult and, in most cases, even impossible. That is why biometric identification methods[3] use statistical algorithms to answer the question what is the probability that user is who he claims to be. There are generally two techniques of biometric identification :

•Identification.

•Authorization.

During the identification process, system collects a sample and then tries to match it with one of the stored templates. Commonly it counts for each template a probability that the sample was collected from the user and chooses one with the highest probability.

Another kind of test is an authorization test. In such test users are first explicitly asked for their names or logins and then system measures a sample of their biometric attributes. After that the system evaluates similarity of the sample to the template of the specified person and accepts or rejects authorization. It is obvious that authorization is much more reliable than identification. Furthermore it is easier to provide and generally faster to perform.

**Fig1.1- Fingerprint Identification Fig1.2-Various Traits**

**Chapter 2**

**LITERATURE REVIEW**

A literature review is a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Most often associated with academic-oriented literature, such reviews are found in academic journals. Literature reviews are a basis for research in nearly every academic field.

**2.1 2017 | Fingerprint Recognition using Image Segmentation | Sangram Bana and Dr.Davinder Kaur**

This paper specifies a study and implementation of a fingerprint recognition system based on Minutiae based matching[7] techniques. This approach mainly involves extraction of minutiae points from the sample fingerprint images and then performing fingerprint matching based on the number of minutiae pairings among two fingerprints in question.

**2.2 2014 | Fingerprint Based Identification System | Jyoti Rajharia, Dr. PC Gupta, Arvind Sharma**

With Reference to this Paper[1], we concluded that different methods and technique are used to identify a person through its fingerprint as the fingerprint is fast and accurate for more reliable and secure system. The methodology of the biometric identification system is represented with the help of diagrams and flow charts which can be used to enhance the quality of the image as well as to verify the identity of a person. Future research work can be carried out to improve the quality of the image by improving the image enhancement technique and develop a better matching technique.

**2.3 2014 | A Review on Fingerprint-Based Identification System | Ritu and Matish Garg**

This paper says that biometric fingerprints are the personal identification[5] tool because of their individuality, uniqueness and reliability. A fingerprint image consists of valleys & ridges on human fingertips. Fingerprint authentication is possibly the most sophisticated method of all biometric techniques. Fingerprint authentication has been thoroughly verified through various applications. All human recognition techniques using fingerprints are based on one of the following three methods: Minutiae-based, correlation-based, and hybrid. This paper provides a review of various fingerprint recognition techniques and then discusses a general minutiae-based fingerprint.

**2.4 2013 | Fingerprint Recognition Using Minutiae Extractor | Manisha Redhu and Dr.Balkishan**

It says that the popular biometrics[2] are used to authenticate a person’s fingerprint which is unique and permanent throughout the person life. Fingerprint Recognition refers to the automated methods of verifying a match between two human fingerprints. Fingerprints are widely used in daily life for more than 100 years due to its feasibility, distinctiveness, permanence, accuracy, reliability, and acceptability. In this paper they projected Fingerprint Recognition using Minutia Score matching method.

**2.5 2012 | Fingerprint Recognition using Robust Local Features | Madhuri and Richa Mishra**

It says that there are many existing human recognition techniques[4] which are based on fingerprints. Most of these techniques use minutiae points for fingerprint representation and matching. These techniques are not rotation invariant and fail when enrolled image of a person is matched with a rotated test image and such techniques fail when partial fingerprint images are matched. This paper proposes a fingerprint recognition technique which uses local robust features for fingerprint representation and matching.

**Chapter 3**

**PROBLEM STATEMENT AND OBJECTIVE**

A trait like face, fingerprint etc. is taken and a dataset comprising of images of that given trait is obtained and is used as input. A MATLAB application is designed to communicate and act as an interface. First, the device is trained using a set of images and then, test data, that is, set of images of trait used as test input, is fed to the device and the above mentioned algorithms are applied. Thus, the output will be obtained for every algorithm and performance will be analysed. This will help in determining the algorithm with maximum accuracy in recognizing the individual whose trait whose trait will be fed as test input.

## We will obtain the output of Euclidean Distance, Support Vector Machine, Minutiae Matching and Back Propagation using ANN algorithms on a dataset of images containing one of the traits to be taken for Biometric Identification, that is, the device will be first trained using above mentioned machine learning algorithms and then, tested on a set of images . We will compare the performance analysis of every output after applying every algorithm and analyzing the obtained graphs and thus, will determine the best algorithm, that is, the algorithm offering the highest accuracy and maximum efficiency (Throughput).

**Chapter 4**

**FINGERPRINT RECOGNITION**

A fingerprint is the pattern of ridges that appear on the surface of the fingertip. It is perhaps the most popular and reliable biometric characteristic used for human authentication. These observations rely basically on two of its properties, namely, individuality and persistence. The former refers to the fact that the fingerprint is unique across individuals and across fingers of the same individual, whereas the second property means that the basic fingerprint characteristics do not change over time. Because of these reasons, fingerprint-based authentication systems have been widely adopted in many applications where high security levels are a need. Moreover, fingerprints have been adopted as a proof of criminalistics evidence in many law courts over the world, becoming therefore a standard mechanism in forensics. The fingerprint structure comprises two levels of features:

**4.1 Global Level:** At the most global level, the fingerprint features correspond to the patterns of ridges and valleys that appear on the fingerprint. These patterns are typically used for fingerprint classification purposes and they do not own any kind of properties for establishing the identity of an individual. Due to these facts, those patterns are denominated macro features. In a typical fingerprint image, the dark lines are related to the ridge patterns, whereas the brighter ones are referred to the valleys.



**Fig 4.1 -Global Level**

**4.2 Local Level:** At the local level, the fingerprint features are known as minutiae. They correspond to the points where the ridge patterns either bifurcate (ridge bifurcations) or terminate abruptly (ridge endings). These features possess the discriminating information for establishing the individuality of fingerprints and therefore the identity of an individual.



**Fig4.2- Local Level**

The fingerprint recognition problem can be grouped into two sub-domains: one is fingerprint verification and the other is fingerprint identification. The fingerprint recognition system constitutes of fingerprint acquiring device, minutia extractor and minutia matcher. The testing database is from the available fingerprints provided by Fingerprint Verification Competition 2002 (FVC2002). So no acquisition stage is implemented.There are two approaches for fingerprint recognition, first approach is minutia based, represents the fingerprint by its local features, like termination and bifurcations. The second approach, which uses image-based methods, tries to do matching on the global features of a whole fingerprint image. To implement a minutia extractor, a three-stage approach is widely by researchers which include preprocessing, minutia extractor and post processing stage. In preprocessing stage Histogram equalization and Fourier Transform do the image enhancement, and then the fingerprint image is binarized using locally adaptive threshold method. The image segmentation is fulfilled by a three step approach which includes block direction estimation,segmentation by direction intensity and region of interest extraction by Morphological operations.

**Chapter 5**

**MACHINE LEARNING**

Machine learning is a branch of artificial intelligence that employs a variety of statistical, probabilistic and optimization techniques that allows computers to “learn” from past examples and to detect hard-to-discern patterns from large, noisy or complex data sets. In this project we conducted a broad survey of the different types of machine learning methods being used, the types of data being integrated and the performance of these methods in biometric identification.

**5.1 MACHINE LEARNING METHODS:**

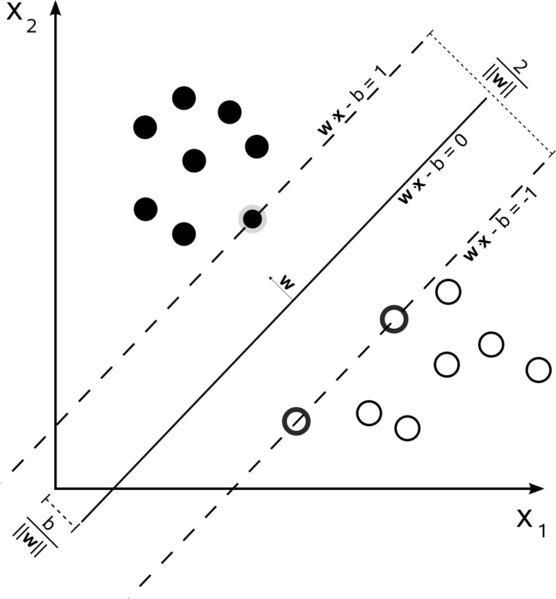
Machine learning, like statistics, is used to analyze and interpret data. Unlike statistics, though, machine learning methods can employ Boolean logic (AND, OR, NOT), absolute conditionality (IF, THEN, ELSE), conditional probabilities (the probability of X given Y) and unconventional optimization strategies to model data or classify patterns. These latter methods actually resemble the approaches humans typically use to learn and classify. Machine learning still draws heavily from statistics and probability, but it is fundamentally more powerful because it allows inferences or decisions to be made that could not otherwise be made using conventional statistical methodologies. For instance, many statistical methods are based on multivariate regression or correlation analysis. While generally very powerful, these approaches assume that the variables are independent and that data can be modeled using linear combinations of these variables. When the relationships are nonlinear and the variables are interdependent (or conditionally dependent) conventional statistics usually flounders. It is in these situations where machine learning tends to shine. Many biological systems are fundamentally nonlinear and their parameters conditionally dependent. Many simple physical systems are linear and their parameters are essentially independent.

Success in machine learning is not always guaranteed. As with any method, a good understanding of the problem and an appreciation of the limitations of the data is important. So too is an understanding of the assumptions and limitations of the algorithms being applied. If a machine learning experiment is properly designed, the learners correctly implemented and the results robustly validated, then one usually has a good chance at success. Obviously if the data is of poor quality, the result will be poor. Likewise if there are more variables than events to predict then it is also possible to create a series of redundant learners. The problem of too many variables and too few examples is called the “curse of dimensionality”. This curse is not restricted to machine learning. It also affects many statistical methods as well. The only solution is to reduce the number of variables (features) or increase the number of training examples. As a general rule, the sample-per-feature ratio should always exceed 5:1. Not only is the size of the training set important, so too is the variety of the training set. Training examples should be selected to span a representative portion of the data the learner expects to encounter. Training too many times on too few examples with too little variety leads to the phenomenon of over-training or simply training on noise. An over-trained learner will generally perform poorly when it tries to process or classify novel data.

There are two general types of machine learning algorithms:

**5.1.1 Supervised Learning:** In supervised learning algorithms a prescient provider or teacher gives the learning algorithm a labeled set of training data or examples. These labeled examples are the training set that the program tries to learn about or to learn how to map the input data to the desired output. For instance a labeled training set might be a set of corrupted images of the number “8” . Since all the images are labeled as being the number “8” and the desired output is the uncorrupted “8”, the learner is able to train under the supervision of a teacher telling it what it is supposed to find. This is the process by which most school children learn.

**5.1.2 Unsupervised Learning:** In unsupervised learning, a set of examples are given, but no labels are provided. Instead it is up to the learner to find the pattern or discover the groups. This is somewhat analogous to the process by which most graduate students learn. Unsupervised learning algorithms include such methods as self-organizing feature maps (SOMs), hierarchical clustering and K-means clustering algorithms. These approaches create clusters from raw, unlabeled or unclassified data. These clusters can be used later to develop classification schemes or classifiers.



**Fig 5.1- Supervised and Unsupervised Learning**

**5.2 LIMITATIONS OF USING MACHINE LEARNING:**

A minimum requirement for any machine learning exercise is having a sufficiently large data set that can be partitioned into disjoint training and test sets or subjected to some reasonable form of n-fold cross-validation for smaller data sets. Typically 5-fold (iteratively taking 20% of the training data out to serve as testing data) or 10-fold cross-validation (iteratively taking 10% of the training data out to serve as testing data) is sufficient to validate any learning algorithm. Data size is not the only limitation for effective machine learning. Data set quality and careful feature selection are also equally important.

Just as data quality is important so too is feature quality. Certainly the subset of features chosen to train a model could mean the difference between a robust, accurate model and one that is flawed and inaccurate. Ideally features should be chosen that are precisely measurable.

It is also important to remember that the machine learning process is essentially a computational experiment. Like any experiment it is based on a hypothesis, it follows defined procedures and it requires data to be validated. Because machine learners represent true experimental procedures, they should be treated as such. Therefore detailed methodological documentation is of paramount importance. Ideally, the data sets used for training and testing should be described in detail and made available to the public. Information about training and testing data should also be well-described including the way in which the sets were partitioned. Likewise the details regarding the algorithms used and their implementations should be provided or recorded to permit others to verify and reproduce the results.

**Chapter 6**

**PROPOSED TECHNIQUES**

**6.1 Algorithms to be used:**

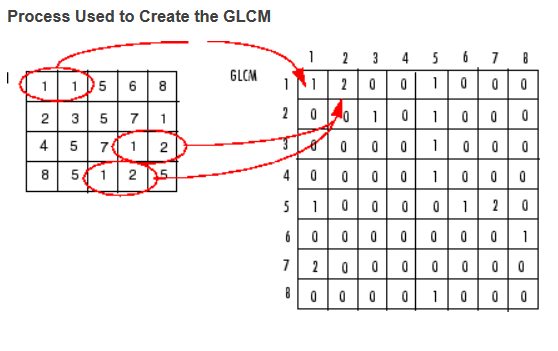
**6.1.1 Gray Level Co-occurrence Matrix:** A statistical method[12] of examining texture that considers the spatial relationship of pixels is the gray-level co-occurrence matrix (GLCM), also known as the gray-level spatial dependence matrix. The GLCM functions characterize the texture of an image by calculating how often pairs of pixel with specific values and in a specified spatial relationship occur in an image, creating a GLCM, and then extracting statistical measures from this matrix. (The texture filter functions, described in Texture Analysis cannot provide information about shape, that is, the spatial relationships of pixels in an image.)

To create a GLCM, use the [graycomatrix](https://in.mathworks.com/help/images/ref/graycomatrix.html) function. The function creates a gray-level co-occurrence matrix (GLCM) by calculating how often a pixel with the intensity (gray-level) value i occurs in a specific spatial relationship to a pixel with the value j. By default, the spatial relationship is defined as the pixel of interest and the pixel to its immediate right (horizontally adjacent), but you can specify other spatial relationships between the two pixels. Each element (i,j) in the resultant GLCM is simply the sum of the number of times that the pixel with value i occurred in the specified spatial relationship to a pixel with value j in the input image.

The number of gray levels in the image determines the size of the GLCM. By default, graycomatrix uses scaling to reduce the number of intensity values in an image to eight, but one can use the NumLevels and the GrayLimits parameters to control this scaling of gray levels.

The gray-level co-occurrence matrix can reveal certain properties about the spatial distribution of the gray levels in the texture image. If most of the entries in the GLCM are concentrated along the diagonal, the texture is coarse with respect to the specified offset.

After one has created the GLCMs, using graycomatrix, one can derive several statistics from them using graycoprops.



**Fig 6.1-GLCM Matrix**

The above figure shows how graycomatrix calculates the first three values in a GLCM. In the output GLCM, element (1,1) contains the value 1 because there is only one instance in the input image where two horizontally adjacent pixels have the values 1 and 1, respectively. glcm(1,2) contains the value 2 because there are two instances where two horizontally adjacent pixels have the values 1 and 2.

The Gray Level Co-occurrence Matrix (GLCM) and associated texture feature calculations are image analysis techniques. Given an image composed of pixels each with an intensity (a specific gray level), the GLCM is a tabulation of how often different combinations of gray levels co-occur in an image or image section. Texture feature calculations use the contents of the GLCM to give a measure of the variation in intensity (a.k.a. image texture) at the pixel of interest.

### Algorithm:

1. Quantize the image data. Each sample on the echogram is treated as a single image pixel and the value of the sample is the intensity of that pixel. These intensities are then further quantized into a specified number of discrete gray levels as specified under **Quantization**.
2. Create the GLCM. It will be a square matrix N x N in size where N is the **Number of levels** specified under **Quantization** and then, normalize the matrix.
3. Calculate the selected **Feature**. This calculation uses only the values in the GLCM.
4. The sample s in the resulting virtual variable is replaced by the value of this calculated feature.

**6.1.2 Euclidean Distance:** In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space. With this distance, Euclidean space[12] becomes a metric space. The associated norm is called the Euclidean norm. Older literature refers to the metric as Pythagorean metric. A generalized term for the Euclidean norm is the L2 norm or L2 distance. The Euclidean Distance between two points p and q is given by the formulae:-

d(p,q) = d(q,p) =

=

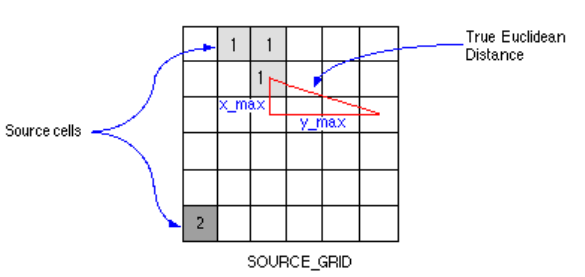
The position of a point in a Euclidean *n*-space is a [Euclidean vector](https://en.wikipedia.org/wiki/Euclidean_vector). So, p and q are Euclidean vectors, starting from the origin of the space, and their tips indicate two points. The [Euclidean norm](https://en.wikipedia.org/wiki/Euclidean_norm), or Euclidean length, or magnitude of a vector measures the length of the vector.

In image analysis, the distance transform measures the distance of each object point from the nearest boundary and is an important tool in computer vision, image processing and pattern recognition. In the distance transform, binary image specifies the distance from each pixel to the nearest non-zero pixel. The euclidean distance is the straight-line

distance between two pixels and is evaluated using the euclidean norm. The city block distance metric measures the path between the pixels based on a four connected

neighbourhood and pixels whose edges touch are one unit apart and pixels diagonally touching are two units apart.

The input source data is a feature class; it will first be converted internally to a raster before the euclidean analysis is performed. The resolution will be smaller of the height or width of the extent of the feature class, divided by 250 and the resolution can be set with the output cell size parameter.



**Fig 6.2-Euclidean Function**

The euclidean distance is calculated from the center of the source cells to the center of each of the surrounding cells and true distance is calculated to each cell in the distance functions. The euclidean algorithm works as follows: for each cell, the distance is calculated to each source cell by calculating the hypotenuse, with the x-max and y-max.

The shortest distance to a source is determined if it is less than the specified maximum distance and the value is assigned to the cell location on the output raster. The output

values for the euclidean distance raster arefloating-point distance values and the cell is at an equal distance to two or more sources, which assigned to the source as encountered

first in the scanning process. The euclidean distance raster tells how close each cell is to the nearest source and raster defines which source zoneand cell value is the closest. The

euclidean direction identifies the direction to the closest source cell.

M X N images can be easily discussed in an MN dimensional euclidean space, called image space. It is natural to adopt the base to form a coordinate system of the image space. The origin of the image space is an image whose gray levels are zero everywhere

Although the algebra of the image space can be easily formulated as above, the euclidean distance of images could not be determined until the metric coefficients of the basis are given. For images of fixed size M by N , every MNth order and positive definite matrix G induces a euclidean distance. Clearly, the information about the spatial relationship, i.e. the distances between the pixels, cannot be reflected by all mutually perpendicular base vectors. Such information, however, often appears in intuitive image distance. A slightly deformed image is very similar to the original one. Here, slightly deformed means that pixels in the deformed image are close to the corresponding pixels in the original image. This implies that a good euclidean distance for images should contain the information of pixel distances. Accordingly, the metric coefficients, which define the euclidean distance, have to be related to the pixel distances. If the metric coefficients depend properly on the pixel distances, the obtained euclidean distance is insensitive to small deformation.

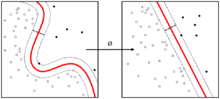
‖ p ‖ = p 1 2 + p 2 2 + ⋯ + p n 2 = p ⋅ p , {\displaystyle \left\|\mathbf {p} \right\|={\sqrt {p\_{1}^{2}+p\_{2}^{2}+\cdots +p\_{n}^{2}}}={\sqrt {\mathbf {p} \cdot \mathbf {p} }},}

**6.1.3 Support Vector Machine:** In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support vector clusteringalgorithm created by [Have](https://en.wikipedia.org/wiki/Hava_Siegelmann) Heighman and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.

More formally, a support vector machine constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

[](https://en.wikipedia.org/wiki/File:Kernel_Machine.png)

**Fig 6.3-Kernel machine**

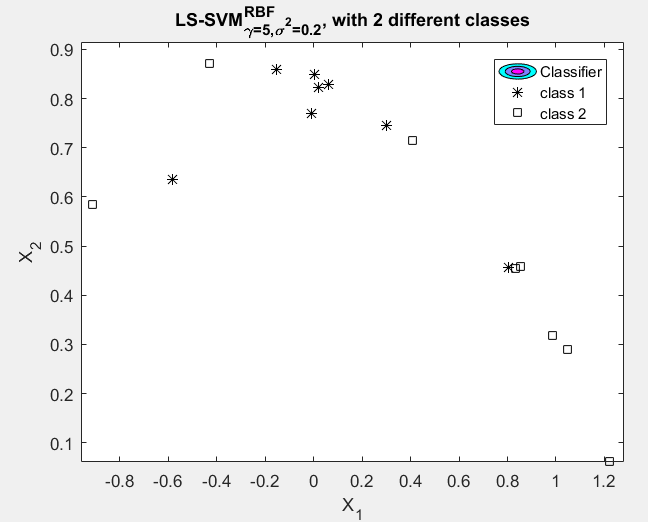
Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function k(x,y)  {\displaystyle k(x,y)}selected to suit the problem. The hyper planes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters ai {\displaystyle \alpha \_{i}}of images of feature vectors {\displaystyle x\_{i}}that occur in the data base. With this choice of a hyperplane, the points {\displaystyle x}in the feature space that are mapped into the hyperplane are defined by the relation: {\displaystyle \textstyle \sum \_{i}\alpha \_{i}k(x\_{i},x)=\mathrm {constant} .}Note that if {\displaystyle k(x,y)} becomes small as {\displaystyle y}grows further away from {\displaystyle x},each term in the sum measures the degree of closeness of the test point {\displaystyle x}to the corresponding data base point {\displaystyle x\_{i}}. In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points {\displaystyle x}mapped into any hyperplane can be quite convoluted as a result, allowing much more complex discrimination between sets which are not convex at all in the original space.

**Applications** :

SVMs can be used to solve various real world problems:

* SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labeled training instances in both the standard inductive and [transductive](https://en.wikipedia.org/wiki/Transduction_(machine_learning)) settings.
* Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback. This is also true of image segmentation systems, including those using a modified version SVM that uses the privileged approach as suggested by Vapnik.
* Hand-written characters can be recognized using SVM.
* The SVM algorithm has been widely applied in the biological and other sciences. They have been used to classify proteins with up to 90% of the compounds classified correctly. Permutation tests based on SVM weights have been suggested as a mechanism for interpretation of SVM models. Support vector machine weights have also been used to interpret SVM models in the past.Posthoc interpretation of support vector machine models in order to identify features used by the model to make predictions is a relatively new area of research with special significance in the biological sciences.

SVM is fundamentally a binary classification algorithm. It falls under the umbrella of machine learning. Image Processing and SVM are related in the way that an SVM might be used to perform image classification, that is, given an input image, the classification task is to decide whether an image is a cat or a dog. The image, before being input into the SVM might have gone through some image processing filters so that some features might be extracted such as edges, color and shape.



**Fig 6.4- Binary SVM Classifier**

**6.1.4 Back Propagation using Artificial Neural Network:** Artificial neural networks (ANNs) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. For example, in image recognition, they might learn to identify images by analyzing example images that have been manually labeled. They do this without any prior knowledge and automatically generate identifying characteristics from the learning material that they process.

An ANN is based on a collection of connected units or nodes called artificial neurons which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

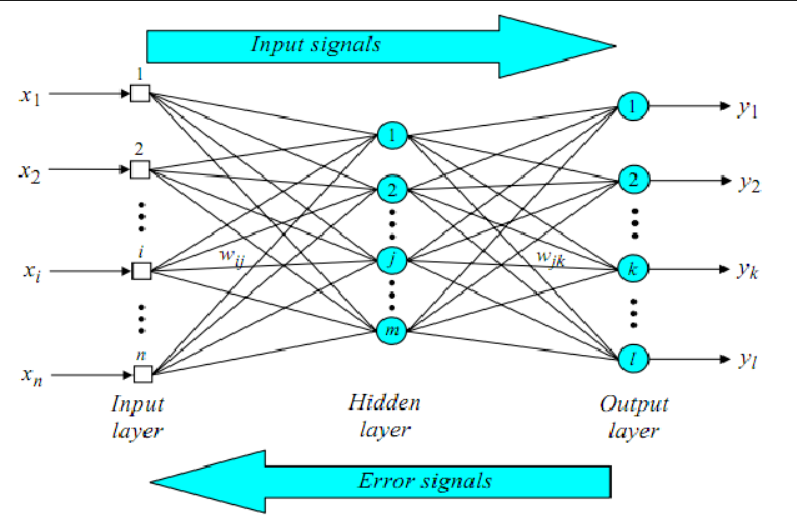
In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called 'edges'. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. ANNs have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

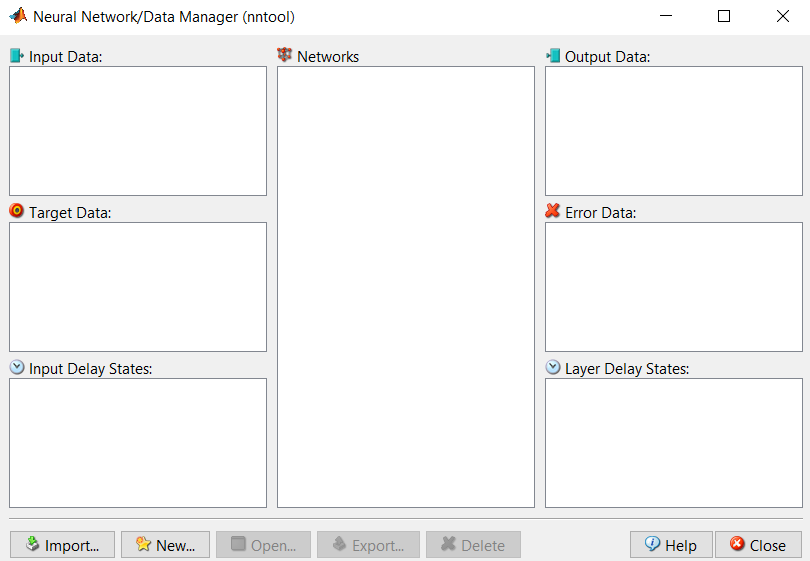
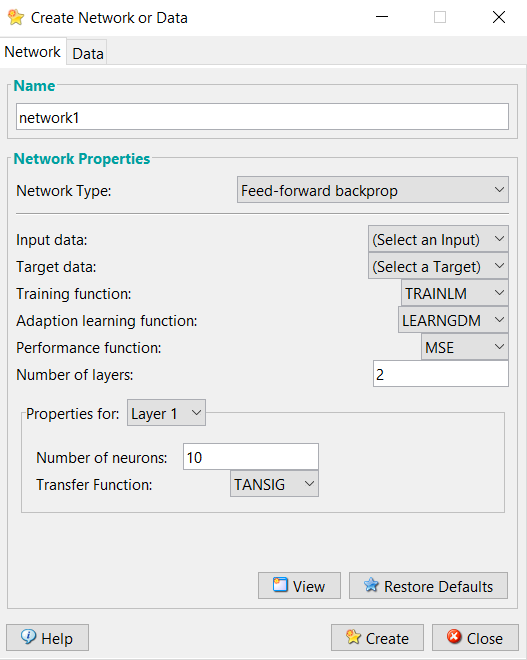
**6.1.4.1 BACKPROPAGATION:** A key trigger for renewed interest in neural networks and learning was Werbos's backpropagation algorithm that effectively solved the exclusive-or problem and more generally accelerated the training of multi-layer networks. Backpropagation distributed the error term back up through the layers, by modifying the weights at each node.

Support vector machines and other, much simpler methods such as linear classifiers gradually overtook neural networks in machine learning popularity.Earlier challenges in training deep neural networks were successfully addressed with methods such as unsupervised pre-training, while available computing power increased through the use of GPUs and distributed computing. Neural networks were deployed on a large scale, particularly in image and visual recognition problems. In 2010, Backpropagation training was accelerated by GPUs and shown to perform better than other pooling variants.

As errors propagate from layer to layer, they shrink exponentially with the number of layers, impeding the tuning of neuron weights that is based on those errors, particularly affecting deep networks.To overcome this problem, Schmidhuber adopted a multi-level hierarchy of networks (1992) pre-trained one level at a time by unsupervised learning and fine-tuned by backpropagation. Once sufficiently many layers have been learned, the deep architecture may be used as a generative model by reproducing the data when sampling down the model from the top level feature activations.

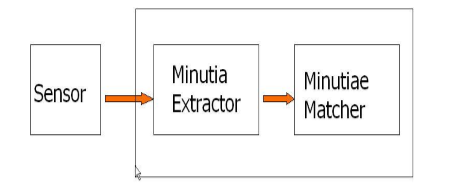


**Fig 6.5-Back Propagation using ANN**

****  ****

**Fig 6.6-nntool Interface Fig 6.7-Creating network**

**6.1.5 Minutiae Matching:** A fingerprint recognition device is constituted of a fingerprint acquiring device for image acquisition step, minutia extractor for extraction of valuable minutiae and minutiae matcher for matching the minutiae .

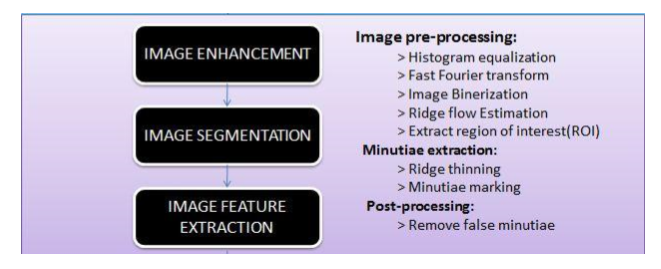


**Fig 6.8**-**Basic Model of Fingerprint Recognition system**

For fingerprint acquisition step the acquiring devices which are optical or semiconductor sensors, are widely used. They exhibit high accuracy and efficiency unless and until users finger is too dirty or dry. However, we are using an online database for the result verification in our project.

We will go through the minutiae extractor and minutiae matcher modules in the next part where algorithm level design is explained and we will extend the discussion in other subsequent sections as well.

A minutiae extractor[2] can be implemented as a three stage approach and is widely used by researchers. They are preprocessing, minutiae extraction and post processing stages.



**Fig 6.9-Minutiae Extractor**

For the image preprocessing steps, we have used histogram equalization followed by Fast Fourier Transform to do the image enhancement and then image binerization is done by locally adaptive threshold[6] method. The image segmentation has two parts, one is ridge flow estimation and other one is by the extraction of region of interest (ROI) using morphological methods. Most of the pre-processing stages used here are a part of standard studies taken by many researchers but here they are carried in our project on basis of a lot of practical results taken by us.

For minutiae extraction stage, we take the help of a three thinning algorithm and we got a morphological thinning operation with a very fine thinning quality and high efficiency. Then the minutiae marking is a simple one just some regular MATLAB functions can handle them. For the post-processing stages, a better and a very fine algorithm is required to remove false minutiae like H-breaks and isolated points etc. The basic concept for minutiae matcher is to take a reference point or line then decide the origin for the co-ordinates and now translate and rotate the whole image in order to get the match. So, it first takes any two random minutiae as a reference pair and then matches their associated ridges. If, the ridges are matched very well then both the fngerprints are aligned and matching is done for all the extracted minutiae.

**6.2 FINGERPRINT IMAGE PREPROCESSING**

**6.2.1 HISTOGRAM EQUALISATION**:

Histogram is a process that attempts to spread out the gray levels in an image so that they are evenly distributed across their range. It basically reassigns the brightness value of each pixel based on the image histogram. Histogram is a technique to produce more visually pleasing result across a wider range of images to produce as flat as possible histogram of the image. The histogram of an image is a graphical plot of the number of occurrences of gray levels in the image against the gray level value.

Procedure to perform histogram equalization:

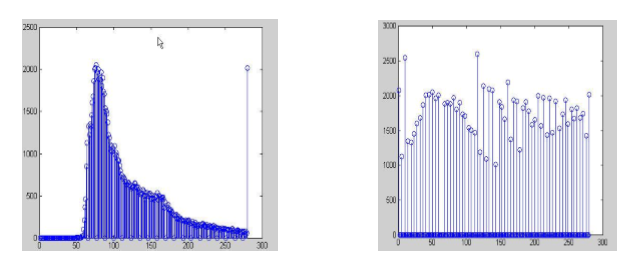
1. Find the running sum of histogram values.

2. Normalize the value from step (1) by dividing by the total number of pixels.

3. Multiply the values from step (2) by the maximum gray-level value and round.

4. Map the gray level values to the results from step (3) using a one-to-one correspondence.

In MATLAB histogram equalization is done using an ingenious MATLAB function histeq (image).



**Fig 6.10-Original Histogram Vs Histogram After Equalization**

6.2.2 **FINGERPRINT IMAGE BINARIZATION**

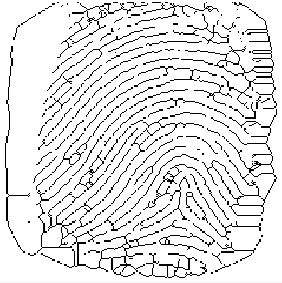
Fingerprint image binarization[5] is done to transform a 8-bit gray image to a 1-bit binarized image where 0-value holds for ridges and 1-value for furrows. And after the binarization operation ridges are highlighted with black color and furrows are highlighted with white color. Here, we will use a locally adaptive binarization method called as adaptive thresholding to binarize the fingeprint image. In this method we transform the gray level to 0 if it is below threshold value and to 1 if it is above threshold value. The threshold value is the mean taken from the gray level of the current block to which the pixel belong.

**Fig 6.11- Actual Image Fig 6.12- Binarized Image**

**6.2.3 FINGERPRINT RIDGE THINNING**

Thinning is the process of reducing binary objects or shapes to strokes whose width is one pixel wide. Here in fingerprint recognition thinning is done to thin the ridges so that each is one pixel thick. In each scan of the fingerprint image, the algorithm removes the redundant pixels in small image window .In our algorithm, for thinning purposes we had invoked an inbuilt morphological operation in MATLAB.

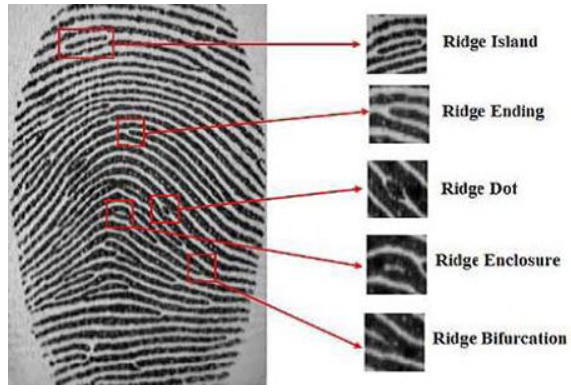
****

**Fig 6.13-Thinned Image**

**6.2.4 FINGERPRINT MINUTIAE EXTRACTION**

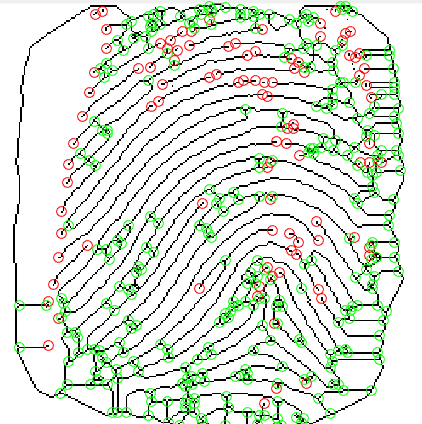
In biometrics and forensic science, minutiae are major features[4] of a fingerprint, using which comparisons of one print with another can be made. Minutiae points are the major features of a fingerprint image and are used in the matching of fingerprints. These minutiae points are used to determine the uniqueness of a fingerprint image. A good quality fingerprint image can have 25 to 80 minutiae depending on the fingerprint scanner resolution and the placement of finger on the sensor. Minutiae include:

* Ridge ending – the abrupt end of a ridge
* Ridge bifurcation – a single ridge that divides into two ridges
* Short ridge, or independent ridge – a ridge that commences, travels a short distance and then ends
* Island – a single small ridge inside a short ridge or ridge ending that is not connected to all other ridges
* Ridge enclosure – a single ridge that bifurcates and reunites shortly afterward to continue as a single ridge
* Spur – a bifurcation with a short ridge branching off a longer ridge
* Crossover or bridge – a short ridge that runs between two parallel ridges
* Delta – a Y-shaped ridge meeting
* Core – a U-turn in the ridge pattern



**Fig 6.14- Minutiae Points**

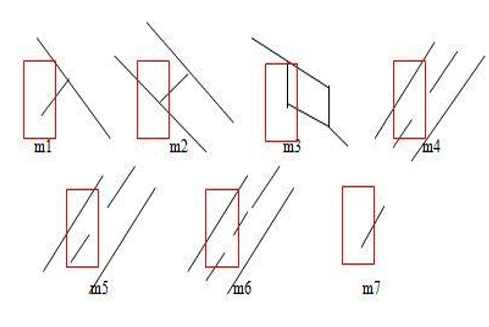
The most popular and widely used biometric identification method is fingerprint recognition. Fingerprints are unique and remain permanent throughout a person’s life. Fingerprint identification has a great utility in forensic science and aids criminal investigations etc. Most of the automatic fingerprint recognition systems are based on local ridge features[5] known as minutiae. Hence it is extremely important to mark these minutiae accurately and reject the false ones. However, fingerprint images are prone to degradation and corruption due to factors such as skin variations and impression conditions such as scars, dirt, humidity and non-uniform contact with the scanning device. Thus it is necessary to apply some type of image enhancement techniques before minutiae extraction. The most important step in automatic fingerprint matching is to reliably extract the minutiae from the captured fingerprint images.

****

**Fig 6.15-Minutiae Extraction**

**6.2.5 FALSE MINUTIA REMOVAL**

The pre-processing stage of a fingerprint image does not remove all the errors. For instance, false ridge breaks and ridge cross-connections due to insufficient amount of inking and over inking are not completely eliminated. Actually all the previous stages themselves occasionally introduce some errors which further lead to spurious minutia. This false minutia[11] significantly affects the accuracy of matching only if they are regarded as genuine minutia. So, some mechanisms of removing these false minutiae are essential in order to keep the fingerprint verification system effective. There are several types of false minutiae, but here in our project we have considered only seven types of false minutiae.



**Fig 6.16-False Minutiae**

1) In the m1 case a spike pierces into a valley.

2) In m2 a spike falsely connects two ridges.

3) In m3 two near bifurcations present in the same ridge.

4) In the m4 case we have two ridge broken points separated by a very short distance and same orientation.

5) m5 is almost similar to that of m4 case with an exception that one part of the broken ridge is so short that it’s another termination is generated.

6) m6 is the extension of the m4 case with an extra property that a third ridge is found in between two parts of a broken ridge.

7) m7 has a very short ridge found in the threshold window.

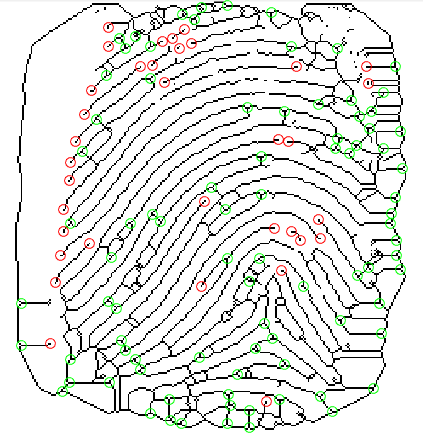
Our approach for removal of false minutiae:

1. If the distance between a bifurcation and a termination is less than D and both the minutia are in the same ridge(m1 case) ,then both of them are removed .Here D is the average interridge width which represents the average distance between two parallel neighboring ridges.

2. If two bifurcations are present in the same ridge and the distance between them is less than D, then both the bifurcations are removed. (m2, m3 cases).

3. If the distance between two terminations is less than D and their directions are almost coincident with only a small angle variation. And they satisfy the condition that no other termination is located in between the two terminations. Then, both the terminations are regarded as false minutia and is considered as part of a broken ridge, hence removed. (case m4,m5, m6).

4. If the distance between two terminations of a very short ridge is less than D ,then it is considered as a false minutia and is removed.

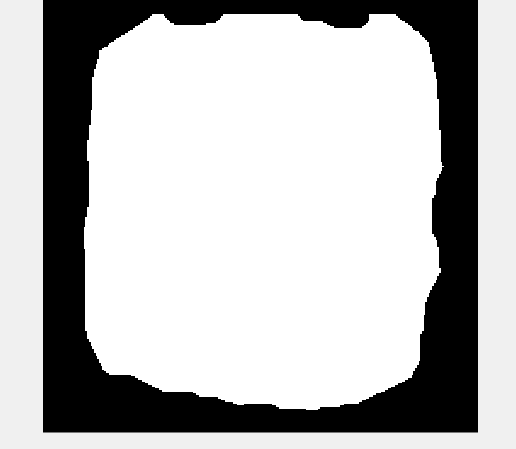
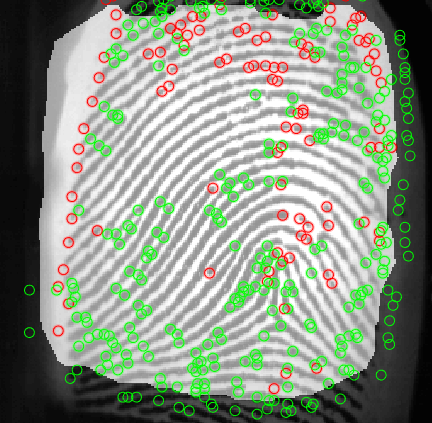


**Fig 6.17- False Minutiae Removal**

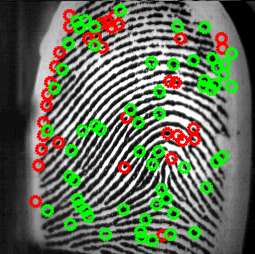
**6.2.6 ROI EXTRACTION BY MORPHOLOGICAL OPERATION**

Two morphological operations „OPEN‟ and „CLOSE‟ are adopted. The OPEN operation expand images and remove peaks which are generally introduced by background noise.

The CLOSE operation usually shrink images and eliminates small cavities . The bound region is obtained after subtraction of closed area from the opened area.

** **

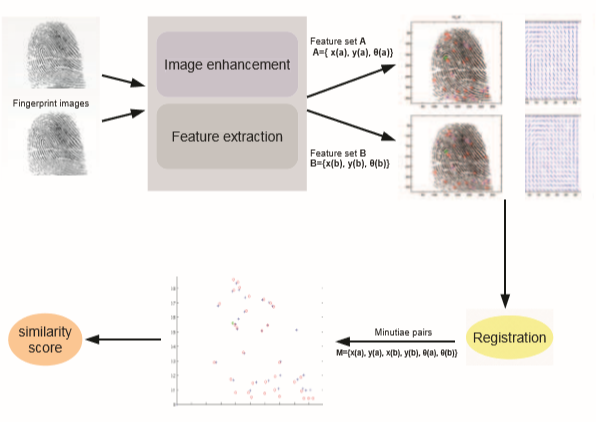
**Fig 6.18- ROI Extraction Fig 6.19-ROI With Minutiae**



**Fig 6.20- Suppressed Minutiae**

**6.2.7 MINUTIAE REPRESENTATION**

Minutiae-based matching algorithms[10] are largely dependent on extracted minutiae information. Robust minutiae-based matching algorithms have to deal with occurrences of missing and spurious minutiae, where missing minutiae can occur as a result of inaccurate feature detection, feature post-processing, or image noise obscuring minutiae detail, and spurious minutiae can be introduced by dry skin, creases, feature detection algorithms, and other potential noise causing agents. The general processes of a ﬁngerprin matching algorithm is presented.



**Fig 6.21 –Minutiae Representation**

In minutiae-based matching, minutiae are commonly represented as minutiae structures called minutia triplets, where a minutia, mi, is described as mi = {x,y,θ}with x,y representing the x-y coordinate of the minutia and θ the angular direction of the main ridge (see Figure 5 left). The main focus of minutiae-based matching is to perform a one-to-one mapping or pairing of minutiae points from a test image minutiae set A = {mA1,mA2,...,mAp},where mAi = {xAi,yAi,θAi}and 1 ≤ i ≤ p to a template image minutiae set B = {mB1,mB2,...,mBq},where mBj = {xBj,yBj,θBj}and 1 ≤ j ≤ q, (4) forming the minutiae pairs (mAk,mBπ(k)) with π(k) as the mapping permutation of pairs from set A to B. Unfortunately, we cannot proceed to ﬁnd minutiae pairs from triplets without some pre-processing for the following critical reasons:

* Fingerprint impressions can differ in orientation, deeming the direction ﬁeld in the triplet useless
* Fingerprint impressions can differ in offset, deeming the x-y ﬁelds in the triplet useless, and
* Skin elasticity creates non-linear distortion or ’warping’ to occur when different directional pressure is applied causing triplet x-y variations to occur.

In general, the lack of invariant characteristics of the triplet structure prohibits it to aid the process of ﬁnding minutiae pairs.

**6.2.8 REGISTRATION AND MATCHING**

In order to address the issues concerning the lack of invariance of the triplet structure, global registration is required. Global registration[7] concerns the alignment and overlay of the template and test ﬁngerprints so that corresponding regions of the ﬁngerprints have minimal geometric distance to each other. Registration can be achieved geometrically by applying (to either the test or template ﬁngerprint minutiae set) a heuristically guided afﬁne transform, where minutiae triplet ﬁeld values are updated with

| =| + |

where θΔ is the orientation difference and (xΔ,yΔ) is the displacement difference in order to super-impose one ﬁngerprint impression on top of the other with accurate overlap and uniform direction. Even with the advent of high distortion, minutiae points within a ﬁngerprint image are still expected to keep their general global location in relation to the majority of other minutiae points and other key landmarks (such as cores and deltas) when alignment is achieved. Speciﬁcally speaking, the spatial distribution or geometric properties of neighbouring minutiae should have minimal difference even in distorted images. If we consider that there are clear limitations in terms of minutiae landmark relative to positioning variability (even with high distortion), while recognising that different ﬁngerprint impressions have orientation and displacement differences, then the global registration process notably reduces the search space. This reduces algorithm complexity for ﬁnding minutiae pairs, since matching pairs are formed in smaller local neighbourhoods (i.e. constraints added for minutiae mappings) once aligned. This allows a naive brute force minutiae pairing process to be avoided.

Following the registration process, we can now produce geometric constraints for the discovery of minutiae matching pairs, including geometric distance:

distr(ma,mb) =

or to account for scale difference (i.e. if we are comparing images collected from different resolution scanners)

distr(ma,mb) =

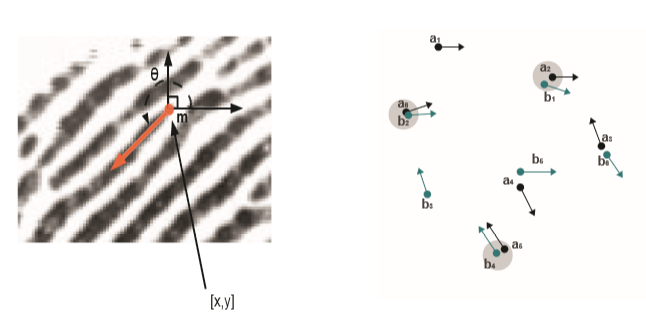
and minutiae angle difference,

(ma,mb) =-, 360--) <

The geometric tolerance rδ is in place to account for distortion that may occur, whereas rθ is the tolerance for angular differences that may arise due to orientation estimations from the ridge orientation images. Following global registration, a local search can now be performed, in order to match minutiae in the δ-neighbourhood that meet the constraints in equations 7-9 (see Figure 5 right). Once genuine minutiae pairs are produced, a metric of similarity, usually called the similarity score[9], can then be calculated. The similarity score must accurately describe how similar two ﬁngerprints are, taking into account all of the relevant information obtained from earlier stages, such as number of genuine minutiae pairs and how similar each pair is. One similarity score given in Liang & Asano (2006) is deﬁned as

Sim(A,B)=(N2match)/(NA\*NB)

where Nmatch is the number of matching minutiae pairs, and NA,NB are the number of minutiae in the overlapped regions of the template and test ﬁngerprints following registration.



**Fig 6.22 Left**: Minutia triplet structure representation. **Right**: Minutiae points from 2 different ﬁngerprints being mapped after registration, with gray circles representing pairs with constraints upheld.

**Chapter 7**

**FINGERPRINT IMPLEMENTATION**

****

**Fig 7.1 Implementation**

**Chapter 8**

**RESULTS**

Performance Evaluation[8] Index Two indexes are well accepted for determining the performance of a fingerprint recognition system:

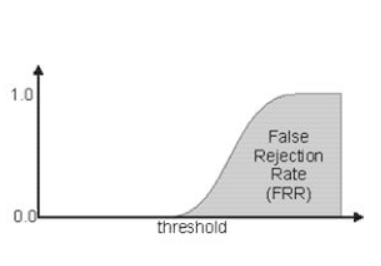
**8.1 False Rejection Rate(FRR):-** The false recognition rate, or FRR, is the measure of the likelihood that the biometric security system will incorrectly reject an access attempt by an authorized user.For an image database, each minutia sample is matched against the remaining samples of a particular finger to compute the FRR.A system’s FRR is basically calculated by the following formula.

(% )FRR = (FR/N )\*100

FR=number of false rejections.

N=number of samples.

If a classification threshold that is too high is applied to the classification scores, some of the client patterns are falsely rejected. Depending on the value of the threshold, between none and all of the client patterns will be falsely rejected. The fraction of the number of rejected client patterns divided by the total number of client patterns is called false recognition Rate (FRR).



**Fig 8.1-False Rejection Rate**

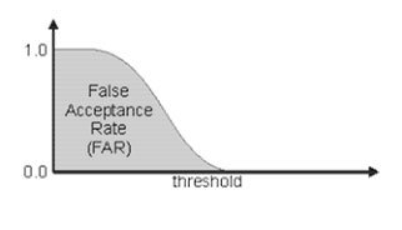
**8.2 False Acceptance Rate (FAR):-** The false acceptance rate, or FAR, is the measure of the likelihood that the biometric security system will incorrectly accept an access attempt by an unauthorized user.Also in a database the first sample of each finger is matched with the first sample of the remaining fingers in order to compute the FAR. A system’s FAR is calculated by the formula.

(%) FAR = (FA/N )\*100

FA=number of cases of false acceptances

N=number of samples.

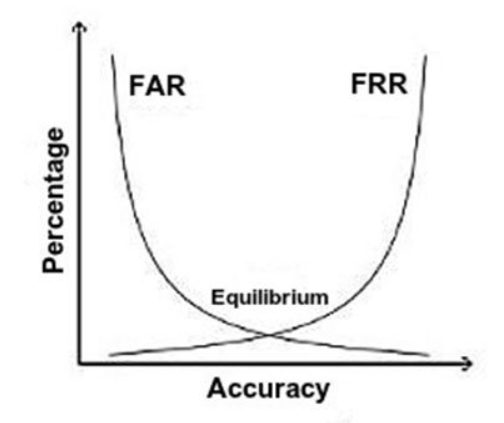
Depending on the choice of the classification threshold, between all and none of the impostor patterns are falsely accepted by the system. The threshold depending fraction of the falsely accepted patterns divided by the number of all impostor patterns is called False Acceptance Rate (FAR). Its value is one, if all impostor patterns are falsely accepted and zero, if none of the impostor patterns is accepted.



**Fig 8.2-False Acceptance Rate**

In a practical scenario a low FAR & a high FRR would ensure that any unauthorized person will not be allowed access. It would also mean that the authorized people will have to put their finger on the device several times before they are allowed access.

Therefore, it is good to have a very low FAR as well as a very low FRR.



**Fig 8.3 FAR and FRR Equilibirium**

**Euclidean Distance Algorithm:-**

For **FRR**, considering the Genuine class, For **FAR**, considering the Imposter class,

**1st image vs others 1st image vs other samples**

N=8, FR=1, Threshold=0.1 N=72, FA=5, Threshold=0.1

(% )FRR = (FR/N )\*100 (% )FAR = (FA/N )\*100

FRR = (1/8)\*100 FAR = (5/72)\*100

FRR = 12.5% FAR = 6.9%

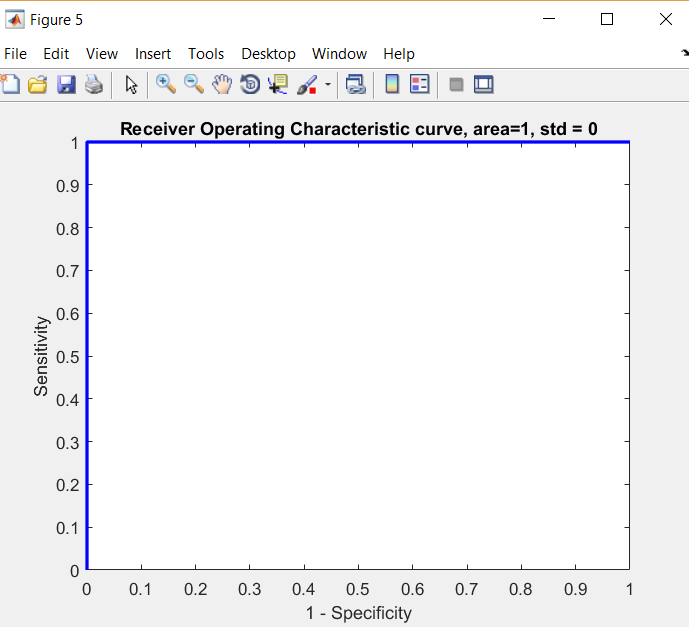
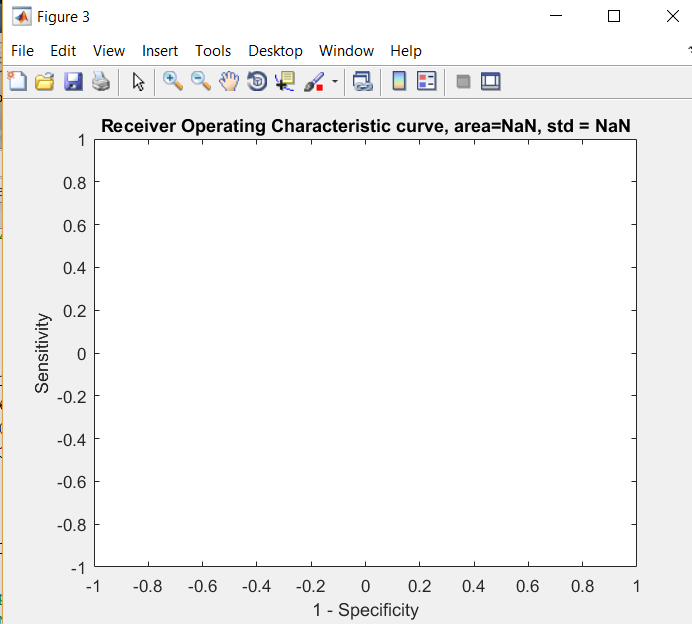
Similarly, after calculating the FAR and FRR ratio for every image, the overall accuracy calculated is found to be :-

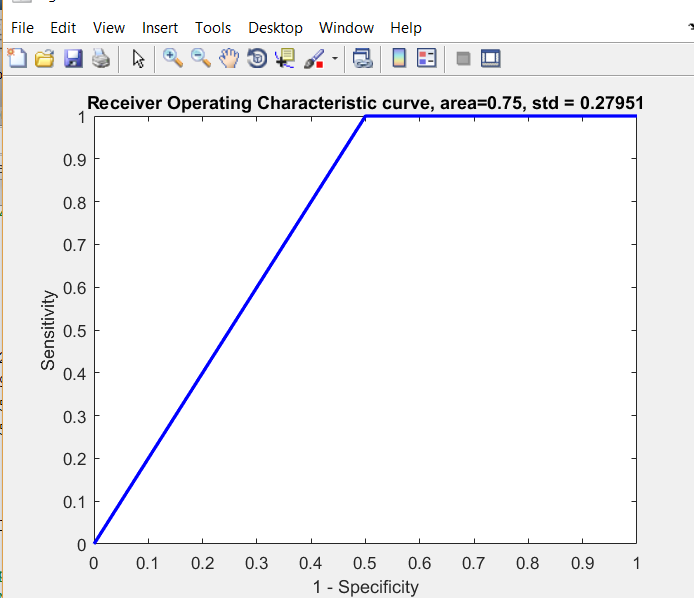
(%)FRR= (FR/N)\*100 (%)FAR=(FA/N)\*100

FRR= (12/48)\*100 FAR=(75/240)\*100

FRR=75.0% FAR=31.25%

**Support Vector Machine Algorithm:-**SVM gives an accuracy of 100% if the testing inputs belong to one or both of the classes given as training input. But there is a decrease in accuracy(upto 80%) if the testing input does not matching with the training input.

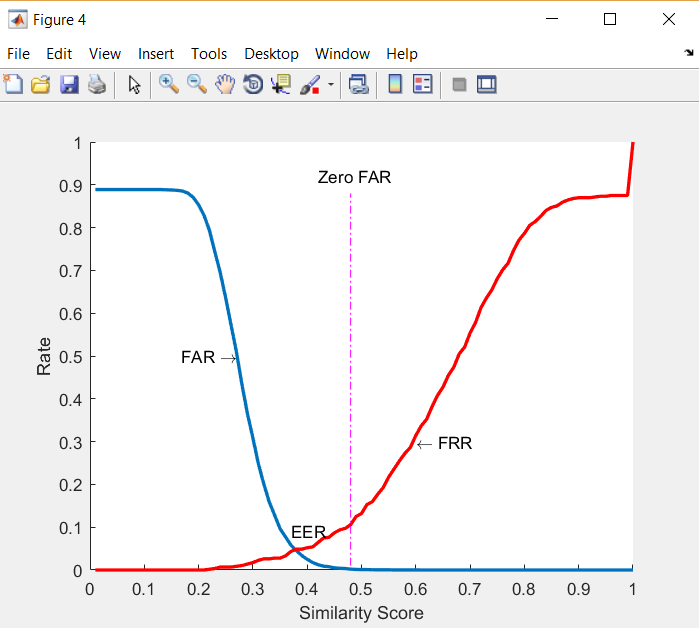
 **Fig 8.4 When testing inputs are of different Fig 8.5 When testing inputs are of same classes but belongs to training inputs class but belongs to training inputs**



**Fig 8.6 When testing inputs are of different**

**classes and does not belong to training inputs**

**Minutiae Matching Algorithm:-**The optimum value for threshold is found to be 0.58 in which the algorithm correctly matches the fingerprint images belonging to the Genuine class with 100% accuracy. There is a slight decrease in accuracy if the threshold value is changed.



**Fig 8.7- FAR and FRR**

**Backpropagation using Artificial Neural Network:-** Since, neural network requires a large number of features to predict the output correctly,thus,as the features were insufficient, and we tried using the algorithm with the available features. Hence, we were not able to get the desired results.

**Chapter 9**

**CONCLUSION AND FUTURE SCOPE**

We have grouped many methods in our project to make Fingerprint matching possible using various approaches and this grouping of methods comes from a wide range of research papers studies and many journals play a vital role in gathering those studies and getting a conclusion to get an efficient recognition system. In our project we realized that fingerprint matching can be done through many approaches and even it is not necessary that the most widely used approach is the optimum one. As, our study helped us to reach a conclusion that minutiae matching is a very good approach but not ideal one when we have the distorted image and more even for investigating the crimes, this approach mostly fails. Hence texture based image is also used by us which is a global matching scheme and also said to be the future of fingerprint matching.

This approach will give the technology new amplitude in order to provide a secure way of authentication. The proposed algorithms performs better than existing recognition algorithms and fusion algorithms. Along with other advantages, in all biometric systems fingerprint based systems are more efficient than other multimodal system, so it minimizes FAR and FRR.

In our project we realized that fingerprint matching can be done through many approaches and even it is not necessary that the most widely used approach is the optimum one. As, our study helped us to reach a conclusion that minutiae matching is a very good approach but not ideal one when we have the distorted image and more even for investigating the crimes, this approach mostly fails. Hence texture based image is also used by us which is a global matching scheme and also said to be the future of fingerprint matching.

Moreover, using ANN for fingerprint matching can also prove to be one of the best and efficient algorithm. But for this purpose we need a big dataset and a feature extraction algorithm which could provide us with more exact features.

**References**

[1] Hong, Lin. *Automatic personal identification using fingerprints*. Michigan State University, 1998.

[2] Maio, Dario, and Davide Maltoni. "Direct gray-scale minutiae detection in fingerprints." *IEEE transactions on pattern analysis and machine intelligence* 19.1 (1997): 27-40.

[3] Jain, Anil, Lin Hong, and Ruud Bolle. "On-line fingerprint verification." *IEEE transactions on pattern analysis and machine intelligence* 19.4 (1997): 302-314.

[4] Ratha, Nalini K., Shaoyun Chen, and Anil K. Jain. "Adaptive flow orientation-based feature extraction in fingerprint images." *Pattern Recognition* 28.11 (1995): 1657-1672.

[5] Farina, Alessandro, Zsolt M. Kovacs-Vajna, and Alberto Leone. "Fingerprint minutiae extraction from skeletonized binary images." *Pattern recognition* 32.5 (1999): 877-889.

[6] Lee, Chih-Jen, and Sheng-De Wang. "Fingerprint feature extraction using Gabor filters." *Electronics Letters* 35.4 (1999): 288-290.

[7] Tico, Marius, Pauli Kuosmanen, and Jukka Saarinen. "Wavelet domain features for fingerprint recognition." *Electronics Letters* 37.1 (2001): 21-22.

[8] Hong, Lin, Yifei Wan, and Anil Jain. "Fingerprint image enhancement: Algorithm and performance evaluation." *IEEE transactions on pattern analysis and machine intelligence* 20.8 (1998): 777-789.

[9] Image Systems Engineering Program, Stanford University. Student project By Thomas Yeo, Wee Peng Tay, Ying Yu Tai.

http://ise0.stanford.edu/class/ee368a\_proj01/dropbox/project22/finger/

[10] FVC2000. http://bias.csr.unibo.it/fvc2000/

[11] FVC2002. http://bias.csr.unibo.it/fvc2002/

[12] Jain, L. C., Halici, U., Hayashi, I., Lee, S. B., & Tsutsui, S. (Eds.). (1999). *Intelligent biometric techniques in fingerprint and face recognition* (Vol. 10). CRC press.

[13] Fingerprint recognition paper by WUZHILI (Department of computer science and engineering, Hong-Kong Bapist University,2002).

**APPENDICES**

clear all,

close all,

clc

I=imread('1\_1.bmp');

imshow(I)

set(gcf,'position',[1 1 600 600]);

%% Binarize

J=I(:,:,1)>160;

imshow(J)

set(gcf,'position',[1 1 600 600]);

%% Thining

K=bwmorph(~J,'thin','inf');

imshow(~K)

set(gcf,'position',[1 1 600 600]);

%% Minutiae

fun=@minutie;

L = nlfilter(K,[3 3],fun);

%% Termination

LTerm=(L==1);

imshow(LTerm)

LTermLab=bwlabel(LTerm);

propTerm=regionprops(LTermLab,'Centroid');

CentroidTerm=round(cat(1,propTerm(:).Centroid));

imshow(~K)

set(gcf,'position',[1 1 600 600]);

hold on

plot(CentroidTerm(:,1),CentroidTerm(:,2),'ro')

%% Bifurcation

LBif=(L==3);

LBifLab=bwlabel(LBif);

propBif=regionprops(LBifLab,'Centroid','Image');

CentroidBif=round(cat(1,propBif(:).Centroid));

plot(CentroidBif(:,1),CentroidBif(:,2),'go')

D=6;

%% Process 1

Distance=DistEuclidian(CentroidBif,CentroidTerm);

SpuriousMinutae=Distance<D;

[i,j]=find(SpuriousMinutae);

CentroidBif(i,:)=[];

CentroidTerm(j,:)=[];

%% Process 2

Distance=DistEuclidian(CentroidBif);

SpuriousMinutae=Distance<D;

[i,j]=find(SpuriousMinutae);

CentroidBif(i,:)=[];

%% Process 3

Distance=DistEuclidian(CentroidTerm);

SpuriousMinutae=Distance<D;

[i,j]=find(SpuriousMinutae);

CentroidTerm(i,:)=[];

%%

hold off

imshow(~K)

hold on

plot(CentroidTerm(:,1),CentroidTerm(:,2),'ro')

plot(CentroidBif(:,1),CentroidBif(:,2),'go')

hold off

%% ROI

Kopen=imclose(K,strel('square',7));

KopenClean= imfill(Kopen,'holes');

KopenClean=bwareaopen(KopenClean,5);

imshow(KopenClean)

KopenClean([1 end],:)=0;

KopenClean(:,[1 end])=0;

ROI=imerode(KopenClean,strel('disk',10));

imshow(ROI)

imshow(I)

hold on

imshow(ROI)

alpha(0.5)

hold on

plot(CentroidTerm(:,1),CentroidTerm(:,2),'ro')

plot(CentroidBif(:,1),CentroidBif(:,2),'go')

hold off

%% Suppress extrema minutiae

[m,n]=size(I(:,:,1));

indTerm=sub2ind([m,n],CentroidTerm(:,1),CentroidTerm(:,2));

Z=zeros(m,n);

Z(indTerm)=1;

ZTerm=Z.\*ROI';

[CentroidTermX,CentroidTermY]=find(ZTerm);

indBif=sub2ind([m,n],CentroidBif(:,1),CentroidBif(:,2));

Z=zeros(m,n);

Z(indBif)=1;

ZBif=Z.\*ROI';

[CentroidBifX,CentroidBifY]=find(ZBif);

imshow(I)

hold on

plot(CentroidTermX,CentroidTermY,'ro','linewidth',2)

plot(CentroidBifX,CentroidBifY,'go','linewidth',2)

%% Orientation

Table=[3\*pi/4 2\*pi/3 pi/2 pi/3 pi/4

5\*pi/6 0 0 0 pi/6

pi 0 0 0 0

-5\*pi/6 0 0 0 -pi/6

-3\*pi/4 -2\*pi/3 -pi/2 -pi/3 -pi/4];

%% Termination Orientation

for ind=1:length(CentroidTermX)

Klocal=K(CentroidTermY(ind)-2:CentroidTermY(ind)+2,CentroidTermX(ind)-2:CentroidTermX(ind)+2);

Klocal(2:end-1,2:end-1)=0;

[i,j]=find(Klocal);

OrientationTerm(ind,1)=Table(i,j);

end

dxTerm=sin(OrientationTerm)\*5;

dyTerm=cos(OrientationTerm)\*5;

figure

imshow(K)

set(gcf,'position',[1 1 600 600]);

hold on

plot(CentroidTermX,CentroidTermY,'ro','linewidth',2)

plot([CentroidTermX CentroidTermX+dyTerm]',...

[CentroidTermY CentroidTermY-dxTerm]','r','linewidth',2)

%% Bifurcation Orientation

for ind=1:length(CentroidBifX)

Klocal=K(CentroidBifY(ind)-2:CentroidBifY(ind)+2,CentroidBifX(ind)-2:CentroidBifX(ind)+2);

Klocal(2:end-1,2:end-1)=0;

[i,j]=find(Klocal);

if length(i)~=3

CentroidBifY(ind)=NaN;

CentroidBifX(ind)=NaN;

OrientationBif(ind)=NaN;

else

for k=1:3

OrientationBif(ind,k)=Table(i(k),j(k));

dxBif(ind,k)=sin(OrientationBif(ind,k))\*5;

dyBif(ind,k)=cos(OrientationBif(ind,k))\*5;

end

end

end

plot(CentroidBifX,CentroidBifY,'go','linewidth',2)

OrientationLinesX=[CentroidBifX CentroidBifX+dyBif(:,1);CentroidBifX CentroidBifX+dyBif(:,2);CentroidBifX CentroidBifX+dyBif(:,3)]';

OrientationLinesY=[CentroidBifY CentroidBifY-dxBif(:,1);CentroidBifY CentroidBifY-dxBif(:,2);CentroidBifY CentroidBifY-dxBif(:,3)]';

plot(OrientationLinesX,OrientationLinesY,'g','linewidth',2)

%% Save in a text file

% In this step, we are going to save the minutia in a file

MinutiaTerm=[CentroidTermX,CentroidTermY,OrientationTerm];

MinutiaBif=[CentroidBifX,CentroidBifY,OrientationBif];

saveMinutia('John Doe',MinutiaTerm,MinutiaBif);

%% Minutia Match

% Given two set of minutia of two fingerprint images, the minutia match

% algorithm determines whether the two minutia sets are from the same

% finger or not.

% two steps:

% 1. Alignment stage

% 2. Match stage

**SUPPORT VECTOR MACHINE**

X=[

-1.490869068 0.523933793 0.350575028 0.616360247

-1.325174925 0.814948206 -0.219020159 0.729246878

-0.6961484 0.925124916 -1.018176858 0.789200343

-1.414270396 0.322307683 0.158834277 0.933128436

-1.434106855 0.220302415 0.325704989 0.888099451

-0.366952511 0.93566954 -1.240430906 0.671713877

0.927929622 0.327290327 -1.416292134 0.161072185

-1.312621929 0.594492814 -0.217887457 0.936016572

-0.101028573 0.688949473 -1.376161069 0.788240169

0.443250554 0.538062262 -1.498756236 0.51744342

0.448278317 0.366469947 -1.487383151 0.672634887

-0.014149245 0.695952705 -1.40931683 0.727513371

-1.338436152 0.54283696 -0.148747594 0.944346787

1.116341969 0.189120708 -1.310719775 0.005257099

-1.05491507 0.455023487 -0.566129513 1.166021096

-0.97327637 0.168784435 -0.519882355 1.32437429

];

Y=[

1

1

1

1

1

1

1

1

-1

-1

-1

-1

-1

-1

-1

-1

];

gam=5;

sig2=0.2;

type='classification';

[alpha,b]= trainlssvm({X,Y,type,gam,sig2,'RBF\_kernel'});

[alpha,b]= trainlssvm({X,Y,type,gam,sig2,'RBF\_kernel','original'});

[alpha,b]= trainlssvm({X,Y,type,gam,sig2,'RBF\_kernel','preprocess'});

Xt=[

-1.05491507 0.455023487 -0.566129513 1.166021096

-0.97327637 0.168784435 -0.519882355 1.32437429

0.85514563 0.458790632 -1.429956765 0.116020503

0.833106188 0.454668194 -1.440765313 0.152990931

];

Ytest=simlssvm({X,Y,type,gam,sig2,'RBF\_kernel'},{alpha,b},Xt)

plotlssvm({X,Y,type,gam,sig2,'RBF\_kernel'},{alpha,b});

% [M N]= perfcurve([1; 1; 1; -1; -1;-1],Ytest,1,'XCrit','FP');

% plot(M,N)

[area, se, deltab, oneMinusSpec, sens, TN, TP, FN, FP] = roc(Ytest, [-1; -1; 1; 1])