International Journal of Advanced Technology in Engineering and Science www.ijates.com
Volume No.02, Issue No. 11, November 2014 ISSN (online): 2348 – 7550

PALMPRINT RECOGNITION SYSTEM USING BACK PROPOGATION NEURAL NETWORK BY APPLYING THE CONCEPT OF LIBRARY

Madhav Prasad Namdev[1], Manoj Kumar Sah[2], Varun Bansal[3]

[123]Asst. Prof, Department of Computer Science and Engineering

Shobhit University Gangoh(India)

ABSTRACT *The research on Palm prints recognition over the past few years. Palm print recognition comprises of palm print acquisition, preprocessing, feature extraction, enrollment and matching.*

The main purpose of this research paper is to go through the basics of palm print recognition, its features and analysis so that we could understand and find the efficient algorithm in near future Palm print images contain rich unique features for reliable human identification, which makes it a very competitive topic in biometric research.

Keywords : Palm Print Acquisition, Recognition, Matching, Distance Metrics, Feature Extraction, Palm Print Recognition Algorithms.

1. INTRODUCTION

Palm is the inner surface of the hand between the wrist and fingers. Palm area contains large number of features such as principle lines, wrinkles, minutiae, datum point features and texture [1]. Palm print recognition system employs high or low resolution images. Most of the system uses the low resolution image [2]. The palm print image is captured using a palm print scanner. Preprocessing has two parts, image alignment and region of interest (ROI) selection. ROI selection is the cropping of palm print image from the hand image. Feature extraction stage obtains proposed features from the preprocessed palm prints. At the last matching compares the captured image features with the stored templates. Methods belonging to low resolution images (75 or 150dpi); where only principal lines, wrinkles, and texture are evident [3]. Some of them use different edge detection methods to extract palm lines, and match them directly or after some feature transformations. Other approaches first extract some features like Gabor filter or wavelets, then use a subspace projection like principal component analysis or linear discriminant analysis to reduce their dimensionality and adopt distance measures or classifiers to compare the reduced features. Methods belonging to the high resolution images (500dpi), where, in addition to principal lines and wrinkles, more discriminant features such as ridges, singular points and minutiae can be extracted.

IJATES

A biometric system is a personal identification system which plays a significant role in daily life. There are two approaches of the personal identification: the first method is token-based such as a passport, a physical key and an ID card, and the second method is based on Knowledge such as a password. However, these approaches have some limitations [4]. In token-based, "token" can be stolen or lost easily and in a knowledge-base in a certain degree knowledge can be forgotten or guessed.

These are some security parameter that is associated with biometrics.

1)Biometrics is based on the uniqueness of human features and we are considering similarity is null.

2) Because biometrics is associated with individual so it can't be shared with other.

3) Biometrics properties cannot be loss, until the serious accident.

4) It can't be copied

Today, the area of personal identification is exploiting computer-aided systems as a safer and more robust method and biometrics is one of the most reliable features that can be used in computer-aided personal recognition The biometric personal identification systems concern with identifying persons by either physiological characteristics such as fingerprints, palm print, iris, face or by using some aspects such as the signature or the voice [5]. Fingerprint-based personal identification has drawn considerable attention over the last 25 years [6]. However, workers and old people may not provide clear fingerprints because of their problematic skin caused by physical work. Recently, voice, face, and iris-based verifications have been studied extensively [7]. Inconvenience with using the traditional methods caused a rapid increase in the application of biometrics. Palm print recognition system is a promising technology which received considerable interest. Among various biometric identifications technologies palm print recognition system has been successful due to its simplicity, feature extraction, matching feature, small size, high precision, real time computation, and the resolution of used images. Palm prints provide a number of privileges over other biometric features, making them an appropriate choice for identification application.

Palm print identification has emerged as one of the popular and promising biometric modalities for forensic and commercial applications [8]. Palm print features are considered promising in identify people. There are two types of Palm print features with reference to the field at which palm print systems are used. The first type of features are the principal lines and wrinkles which could be extracted from low resolution images (<100 dpi) and it is used for identification in the commercial applications. The second type of features are the singular point, ridges and minutiae point which could be extracted from high resolution images (>100dpi) and it is used for forensic applications such as law enforcement application [4]. Both high and low resolution image Features in palm print are shown in Figure 1 [8].

IJATES

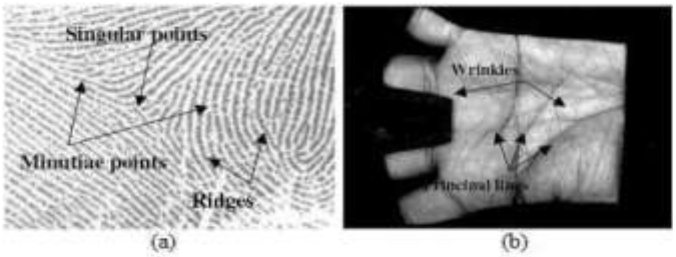

Figure 1.1: Palm print Features

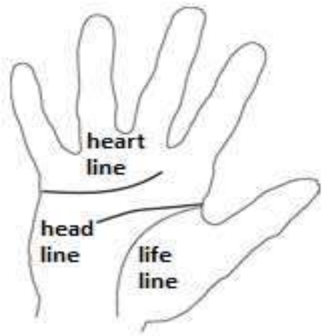

Figure 1.2: Principal lines of a palm

The inner surface of the palm normally contains three flexion creases, secondary creases and ridges. The flexion creases are also called principal lines and the secondary creases are called wrinkles. The flexion and the major secondary creases are formed between the 3rd and 5th months of pregnancy and superficial lines appear after we born. Although the three major flexions are genetically dependent, most of other creases are not . Even identical twins have different palm prints . These non-genetically deterministic and complex patterns are very useful in personal identification. Human beings were interested in palm lines for fortune telling long time ago. images. High resolution images are suitable for forensic applications such as criminal detection . Low resolution images are more suitable for civil and commercial applications such as access control. Generally speaking, high resolution refers to 400 dpi or more and low resolution refers to 150 dpi or less. Researchers can extract ridges, singular points and minutia points as features from high resolution images while in low resolution images they generally extract principal lines, wrinkles and texture. Initially palm print research focused on high-resolution images [5-6] but now almost all research is on low resolution images for civil and commercial applications. This is also the focus of this thesis. The design of a biometric system takes account of five objectives: cost, user acceptance and environment constraints, accuracy, computation speed and security. Reducing accuracy can increase speed. Typical examples are hierarchical approaches. Reducing user acceptance can improve accuracy. For instance, users are required to provide more samples for training. Increasing cost can enhance security. We can embed more sensors to collect different signals for liveness detection. In some applications, environmental constraints such as memory usage, power consumption, size of templates and size of devices have to be fulfilled. A biometric system installed in PDA (personal digital assistant) requires low power and memory consumption but these requirements may not be vital for biometric access control systems. A practical biometric system should balance all these aspects.

LITERATURE WORK

2. PALMPRINT SCANNERS

There are various ways to capture palm print image. Researchers utilize CCD-based scanners, digital scanners, video camera and tripod to collect palm print images. Fig.2 shows a CCD-based scanner developed by Hong Kong Polytechnic University [9]. Rafal Kozik and Michal Chores were made a

IJATES

special tripod to capture palm images as shown in fig.3 [9]. Its shape and proportion minimize errors caused by camera movements and rotation. A CCD-based scanner captures high resolution images and aligns palms accurately because it has pegs for guiding the placement of hand.

Figure 2: A CCD-based palm print scanner

3. PRE-PROCESSING

Preprocessing is used to correct distortions, align different palm prints, and to crop the region of interest for feature extraction.

There are five steps

1. Binarizing the palm image
2. Boundary tracking
3. Key point's detection
4. Establishing a coordination system and
5. Extracting the central part.

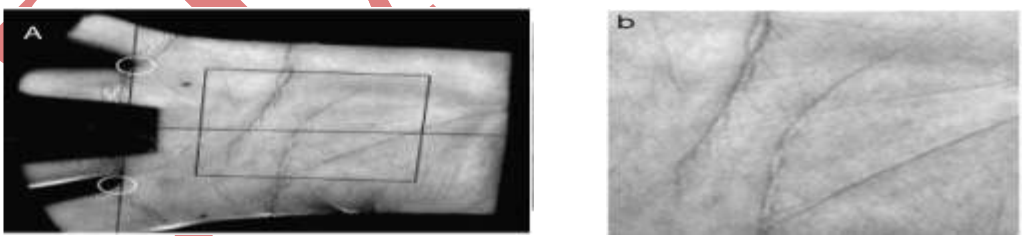

Figure 3: (a) key points and coordinate system, (b) ROI extraction

The first two steps are common for any preprocessing algorithm. Third step has different variations such as tangent based, finger based to detect the key points between fingers. The tangent based method considers the edges of two fingers holes on binary image which are to be traced and the common tangent of two fingers holes is found to be axis. The middle point of the two tangent points is defined as the key points for establishing coordinate system [10].

IJATES

The central part of palm image is then segmented. Most of the algorithms segment square regions for feature extraction but some of algorithm extracts circular and half elliptical regions. The square region is easier than others.

4. FEATURE EXTRACTION AND MATCHING

The aim of this section is to recognize a correct person to authenticate and to prevent multiple people from using the same identity. In identification, the system recognizes an individual by searching the templates of all users in the database for matching. Research on feature extraction and matching algorithms are classified as follows: Line based, subspace based, Statistical based and coding based [11].

4.1 LINE BASED APPROACH This approach develops edge detectors and makes use of the magnitude of the palm lines. The magnitudes of the palm lines are projected in x and y coordinates forming histograms. After this, the first and second order derivatives of the palm images are calculated. The first order derivative is used to identify the edge points and corresponding directions. The second order derivative is used to identify the magnitude of lines. Then the Euclidian distance is used for matching.

4.2 SUB SPACE BASED APPROACH This approach makes use of Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Independent Component Analysis (IDA). The spatial coefficients are considered as the features used for matching. This approach does not need any prior knowledge of the palm prints.

4.3 STATISTICAL APPROACH These are of two types, local and global. The local approach transforms the image in another domain. This transformed image is then divided into several regions such as mean and variance of each region. The local features include moments, center of gravity and density. The global approach is applied on the whole palm print image. This is the only difference between the local and global approach. The local approach is applied on the segments of the palm print image whereas the global approach is applied on the whole image [12,13].

4.4 CODING APPROACH This approach uses a single Gabor filter to extract the local phase information of palm print. This extracted phase information is used by the palm print recognition systems to reduce the registered data size and to deal with non-linear distortion between palm print images. This approach has very low memory requirement and fast matching speed

5. PROPOSED WORK

5.1 FILTERING

This sections considers filters and their types, emphasis will be on non-linear ordered statistics filters like the median filter and the adaptive median filter. In image processing, a filter is a device or process

IJATES

that removes some unwanted component or feature from a image. Filtering is a class of image processing, the defining feature of filters being the complete or partial suppression of some aspect of the image. Most often, this means removing some frequencies and reduce background noise. However, filters do not exclusively act in the frequency domain; especially in the field of image processing many other targets for filtering exist. There are many different bases of classifying filters and these overlap in many different ways; there is no simple hierarchical classification. We will focus our attention to non-linear filters since our work is geared towards is a filtering our impulse noise from a 2-dimensional grayscale image. A nonlinear filter is a image-processing device whose output is not a linear function of its input. Terminology concerning the filtering problem may refer to the time domain (state space) showing of the image or to the frequency domain representation of the image. Nonlinear filters locate and remove data that is recognised as noise. The algorithm is 'nonlinear' because it looks at each data point and decides if that data is noise or valid image. If the point is noise, it is simply removed and replaced by an estimate based on surrounding data points, and parts of the data that are not considered noise are not modified at all. Linear filters, such as those used in band pass, high pass, and low pass, lack such a decision capability and therefore modify all data.

Image filtering is used to:

- ✓ Remove noise
- ✓ Highlight contours
- ✓ Sharpen contrast
- ✓ Detect edges

Image enhancement is a very first step in digital image processing. It is of two types

a) Spatial Domain
b) Frequency Domain

Filtering operations that are performed directly on the pixels of an image are referred as Spatial Filtering. The process of spatial filtering consists simply of moving the filter mask from point to point in an image. At each point(x, y), the response of the filter at that point is calculated using a predefined relationship. Spatial Filters can be distinguished as Smoothing linear filters. The response of smoothing, linear spatial filter is simply the average of the pixels contained in the neighbourhood of the filter mask. These filters sometimes are called averaging filters. They are also referred to as low pass filters.

5.1.1 MEDIAN FILTER

We have seen that smoothing (low pass) filters reduce noise. However, the underlying assumption is that the neighbouring pixels represent additional samples of the same value as the reference pixel, i.e. they represent the same feature. At edges, this is clearly not true, and blurring of features results. We have used convolution techniques to implement weighting kernels as a neighbourhood function, which represented a linear process. There are also nonlinear neighbourhood operations that can be performed for the purpose of noise reduction that can do a better job of preserving edges than simple smoothing filters. One such method is known as median filtering. Median filtering is a nonlinear method used to remove noise from images. It is widely used as it is very effective at removing noise while preserving edges. It is particularly effective at removing 'salt and pepper' type noise. The median filter works by

IJATES

moving through the image pixel by pixel, replacing each value with the median value of neighbouring pixels. The pattern of neighbours is called the "window", which slides, pixel by pixel over the entire image2pixel, image. The median is calculated by first sorting all the pixel values from the window into numerical order, and then replacing the pixel being considered with the middle (median) pixel value. In particular, compared to the smoothing filters examined thus far, median filters offer three advantages:

- ✓ No reduction in contrast across steps, since output values available consist only of those present in the neighborhood (no averages).
- ✓ Median filtering does not shift boundaries, as can happen with conventional smoothing filters (a contrast dependent problem).
- ✓ Since the median is less sensitive than the mean to extreme values (outliers), those extreme values are more effectively removed.

The median is, in a sense, a more robust “average” than the mean, as it is not affected by outliers(extreme values). Since the output pixel value is one of the neighbouring values, new “unrealistic” values are not created near edges. Since edges are minimally degraded, median filters can be applied repeatedly, if necessary.

_	_	_
_	origin	_
_	_	_

Figure 5.1.1: Pixel window and Origin

In image processing, several filtering algorithms belong to a category called windowing operators. Windowing operators use a window, or neighbourhood of pixels, to calculate their output [22]. For example, windowing operator may perform an operation like finding the average of all pixels in the neighbourhood of a pixel. The pixel around which the window is found is called the *origin*. Figure 4.1, below, shows a 3 by 3 pixel window and the corresponding origin.

All the Median Filters are rank order filter, which are usually common filtering algorithm in image processing systems. Median Filters are nonlinear filter, so while it is easy to develop, it is difficult to understand its properties. It offers several useful effects, such as smoothing and noise removal [23].

Median filters are very useful in salt -and-pepper noise filtering. Since the rank order filter uses no arithmetic, a mathematical description is difficult to represent efficiently [24].

5.1.1.1 STANDARD MEDIAN FILTER

The best-known order-statistics filter is the median filter, which as its name implies, replaces the value of a pixel by the median of the gray levels in the neighbourhood of that pixel:

$f(x,y)= \text{median}_{(s,t)}\{g(s,t)\}$

The original value of the pixel is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise. The median filter is a non-linear ordered statistic digital filtering technique which is normally used to reduce noise drastically in an image. It is one of the best windowing operators out of the many windowing operators like the mean filter, min and max filter and the mode filter. The median filter filters each pixel in the image in turn and its nearby neighbours are used to decide whether or not it is representative of its surroundings. Normally, instead of replacing the pixel value with the mean of neighbouring pixel values, median filter replaces it with the median of those values. That is, the values from the surrounding neighbourhood are first sorted into numerical order, and then the value of the pixel in question is replaced with the middle (median) pixel value. The neighbourhood is referred to as the *window*. The window can have various shapes cantered on the target pixel.

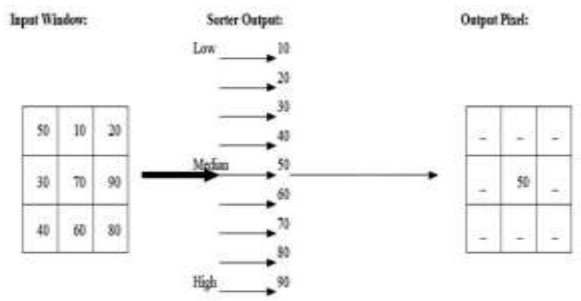

Figure 5.1.1.1: A graphical depiction of the median filter operation

The square is a typical shape chosen for windows defined for 2D images. It should be noted that under normal circumstances the median filter, is performed using a window containing an odd number of pixels. If the neighbourhood under consideration consists of an even number of pixels, the median value selected as the output is the average of the two middle pixel values. The figure below illustrates an example of how the median filter calculation is performed in the window

This filter works by analyzing the neighbourhood of pixels around an origin pixel like in the diagram above, for every valid pixel in an image. For this case, a 3×3 window, of pixels is used to calculate the output. For every pixel in the image, the window of neighbouring pixels is found. As shown in the example above, the pixel values in the window are sorted in ascending order and the median value is chosen, in this case the median value is 50. Next, the pixel in the output image corresponding to the origin pixel in the input image is replaced with the value specified by the filter order. The value in the origin which is 70 is replaced by 50. One of the advantages of median filter over the other rank order filters especially the mean filter, is that the median value is a more robust average than the mean value; the median value will not be affected significantly by one very unrepresentative pixel in neighbourhood. The median value of the surrounding pixels is most likely to be the value of one of the pixels in the neighbourhood within the window. Thus the median filter is least likely to create new unrealistic pixel values especially when the filter is working in transition zones. For this reason, the median filtering technique is much better than the mean filtering technique in terms of preserving sharp edges [21].

5.1.1.2 ADAPTIVE MEDIAN FILTER

This filter is used for:

1) Remove impulse noise and reduce the distortion along the boundaries.
2) It removes salt and pepper noise
3) Smoothness ,non-impulsive noise

The median filter discussed performs well as long as the spatial density of the impulse noise is not large (as a rule of thumb, P_a and P_b less than 20%). It is shown in this section that adaptive median filtering can handle impulse noise with probabilities even larger than these. An additional benefit of the adaptive median filter is that it seeks to preserve detail while smoothing non-impulse noise, something that the "standard" median filter does not do. As in all the nonlinear ordered statistics filters in literature, the adaptive median filter also works in a rectangular window area unlike those filters, however, the adaptive median filter , S_{xy} during filter operation, depending on certain conditions listed in this section. Keep in mind that the output of the filter is a single value used to replace the value of the pixel at (x, y), the particular point on which the window is cantered at a given time.

5.1.1.2.1: WORKING OF ADAPTIVE MEDIAN FILTER

Consider the following notations:

Z_{min}= minimum gray-level value in R_{xy}

Z_{max}= maximum gray level value in R_{xy}

Z_{med} =median gray level value in R_{xy}

Z_{xy}= gray level at coordinates(x,y)

R_{xy}= maximum allowed size of window

IJATES

The adaptive median filtering works in two levels i.e; Level A and Level B

Level A: $A_1= Z_{med}- Z_{min}$

$A_2= Z_{med}- Z_{max}$

if $A_1>0$ and $A_2< 0$, Goto level B

else increases the size of window

if window size<= R_{xy}, Repeat level A

else the output is Z_{med}

Level B: $B_1= Z_{xy}- Z_{min}$

$B_2= Z_{xy}+ Z_{max}$

if $B_1>0$ and $B_2<0$, the output is Z_{xy}

else, output is Z_{med}

The key to understanding the mechanics of this algorithm is to keep in mind that it has three main purposes: to remove salt-and-pepper (impulse) noise, to provide smoothing of other noise that may not be impulsive, and to reduce distortion, such as excessive thinning or thickening of object boundaries. The values of Z_{max} and Z_{min} are considered statistically by the algorithm to be "impulse like" noise components even if these are not the lowest and highest possible pixel values in the image. With these observations in mind, we see that the purpose of level ***A*** is to determine if the median filter output Z_{med} , is an impulse (black or white) or not. If the condition, min max $Z_{min}< Z_{med}< Z_{max}$ holds, then Z_{med} cannot be all impulse for the reason mentioned in the previous paragraph. In this case, we go to level *B* and test to see if the point in the centre of the window Zxy , is itself an impulse (recall that Zxy is the point being processed). If the condition *B*1and *B*2 0 is true, then min max $Z_{min}< Z_{med}< Z_{max}$ and *Zxy* cannot be an impulse for the same reason that Zmed was not. In this case, the algorithm outputs the unchanged pixel value Zxy [34]. By not changing these "intermediate-level "points, distortion is reduced in the image. If the condition *B*10 AND *B*2 0is false, then either $Z_{min}= Z_{xy}$ or $Z_{max} = Z_{xy}$. In either case, the value of the pixel is an extreme value and the algorithm outputs the median value Z_{med} , which we know from level A that is not an impulse. The last step is what the standard median filter does. The problem is that the standard median filter replaces every point in the image by the median of the corresponding neighbourhood. This causes unnecessary loss of detail. Continuing with the explanation, suppose that level A does find an impulse (i.e. it fails the test that would cause it to branch to level B).The algorithm then increases the size of the window and repeats level A. This looping continues until the algorithm either finds a median value that is not an impulse (and branches to level B), or the maximum window size is reached. If the maximum window size is reached, the algorithm returns the value of Z_{xy} . Note that there is no guarantee that this value is not an impulse. The smaller the noise probabilities P_a and/or P_b are, or the larger S_{max} is allowed to be, the less likely it

IJATES

is that a premature exit condition will occur. This is plausib1e. As the density of the impulses increases, it stands to reason that we would need a larger window to "clean up" the noise spikes. Every time the algorithm outputs a value, the window S_{max} is moved to the next location in the image. The algorithm then is reinitialized and applied to the pixels in the new location. As indicated in the median filtering algorithm, the median value can be updated iteratively using only the new pixels, thus reducing computational overhead [25, 26].

5.2 USE OF LIBRARY FUNCTION

Designing the library, one of the main ideas was to make it flexible, reusable, and easy to use and understand. Instead of combining several neural network entities into a single class and making a mess, which leads to loosing flexibility and clarity in the code and design, all entities were split into distinct classes, making them easier to understand and reuse. Some neural networks libraries tend to combine the entity of neuron's network together with the learning algorithm, what makes it hard to develop another learning algorithm which can be applied to the same neural network architecture. Some other libraries and applications do not extract such entities, like neurons, layers of neurons, or a network of layers, but implement the entire neuron network architecture in a single class. In some cases, it is arguable what is better, because there may be such unusual neural network architectures, where it is hard to split the network into layers and neurons. In some other cases, networks do not tend to multi-layer architecture, so it may be useless to have an additional entity like layer. But in most cases, it is favourable to split all these entities into distinct classes, what leads not only to easier understanding, but also allows reusing of all these components and building new neural networks architectures from smaller generic pieces.

The library contains six main entities:

- Neuron - a base abstract class for all neurons, which encapsulates such common entities like a neuron's weight, output value, and input value. Other neuron classes inherit from the base class to extend it with additional properties and specialize it.
- Layer - represents a collection of neurons. This is a base abstract class, which encapsulates common functionality for all neuron's layers.
- Network - represents a neural network, what is a collection of neuron's layers. This is a base abstract class, which provides common functionality of a generic neural network. To implement a specific neural network architecture, it is required to inherit the class, extending it with specific functionalities of any neural network architecture.
- I Activation Function - activation function's interface. Activation functions are used in activation neurons - the type of neuron, where the weighted sum of its inputs is calculated and then the value is passed as input to the activation function, and the output value becomes the output value of the neuron.
- I Unsupervised Learning - interface for unsupervised learning algorithms - the type of learning algorithms where a system is provided with sample inputs only during the learning phase, but not with

IJATES

the desired outputs. The aim of the system is to organize itself in such a way to find correlation and similarities between data samples.

- Supervised Learning - interface for supervised learning algorithms - the type of learning algorithms where a system is provided with sample inputs, with desired output values during the learning phase. The aim of the system is to generalize learning data, and learn to provide the correct output value when it is presented with the input value only.

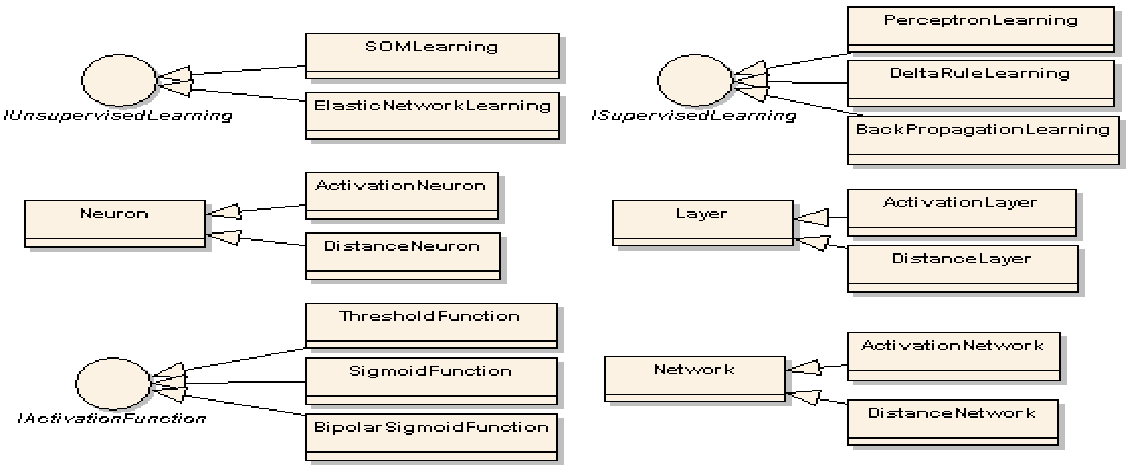

Figure 5.2: Libraries Entity

The library provides the following neural network architectures:

- Activation Network - the neural network where each neuron computes its output as the activation function's output, and the argument is a weighted sum of its inputs combined with the threshold value. The network may consist of a single layer, or of multiple layers. Trained with supervised learning algorithms, the network allows to solve such tasks as approximation, prediction, classification, and recognition.
- Distance Network - the neural network where each neuron computes its output as a distance between its weight values and input values. The network consists of a single layer, and may be used as a base for such networks like Kohonen Self Organizing Map, Elastic Network, and Hamming Network.

Different learning algorithms are used to train different neural networks, and are used to solve different problems:

- Perceptron Learning - the algorithm may be considered as the first neural network learning algorithm, and its history starts from 1957. The algorithm may be used with a one-layer activation network, where each neuron has a threshold activation function. The range of its applications are rather small and limited the with classification of linearly separable data.
- Delta Rule Learning - the algorithm is a next step after the perceptron learning algorithm. It utilizes the activation function's derivative, and may be applicable to single-layer activation networks only, where

each neuron has a continuous activation function instead of a threshold activation function. The most popular continuous activation function is the unipolar and bipolar sigmoid function. Because the algorithm may be applied to one-layer networks only, it is limited to some classification and recognition tasks mostly.

- Back Propagation Learning - this is one of the most popular and known algorithms for multi-layer neural network learning. Initially, it was described in 1974, and from that time, it was extensively studied and applied to a broad range of different tasks. Because the algorithm is able to train multi-layer neural networks, the range of its applications is very great, and includes such tasks as approximation, prediction, object recognition, etc.
- SOM Learning - this algorithm was developed by Kohonen, and may be considered as one of the most famous unsupervised learning algorithms for clusterization problems. It treats neural network as a 2D map of nodes, where each node may represent a separate class. The algorithm organizes a network in such a way, that it becomes possible to find the correlation and similarities between data samples.
- Elastic Network Learning- the algorithm is similar to the idea of the SOM learning algorithm, but it treats network neurons not as a 2D map of nodes, but as a ring. During the learning procedure, the ring gets some shape, which represents a solution. One of the most common demonstrations of this learning algorithm is the Traveling Salesman Problem (TSP).

5.3 BACK – PROPAGATION NEURAL NETWORK

A typical back-propagation network [27] with Multi-layer, feed-forward supervised learning is as shown in the Figure (4.3). Here learning process in Back-propagation requires pairs of input (and target vectors. The output vector ‟o„ is compared with target vector ‟t „. In case of difference of ‟o„ and „t„ vectors, the weights are adjusted to minimize the difference. Initially random weights and thresholds are assigned to the network. These weights are updated every iteration in order to minimize the mean square error between the output vector and the target vector [28].

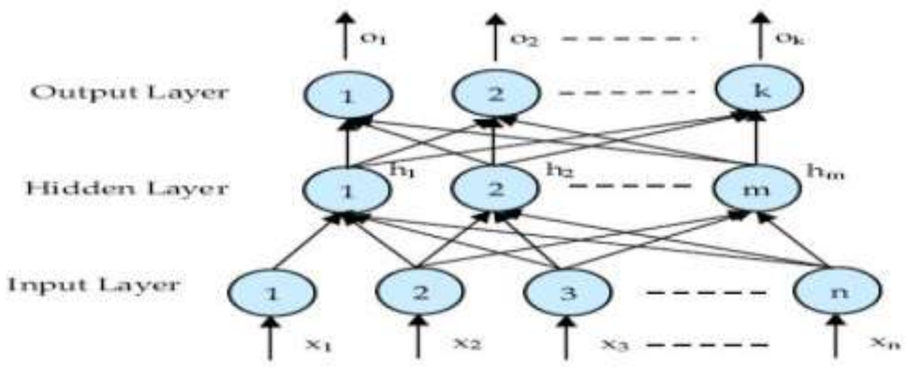

Figure 5.3: Basic block of Back propagation neural networks

Input for hidden layer is given by

$$net_m = \sum_{z=1}^{n} x_z w_{mz}$$

Where m is index of input layer, z is a set of integer numbers, x are entire pair input and w is weight. The units of output vector of hidden layer after passing through the sigmoid function as activation function are given by

$$h_m = \frac{1}{1+\exp(-net_m)}$$

Where m is index of hidden layer output vector and is exponential function.

In same manner, input for output layer is given by

$$net_k = \sum_{z=1}^{m} h_z w_{kz}$$

Where k is index of output layer,h is output vector of hidden layer and the units of output vector of output layer are given by

$$h_k = \frac{1}{1+\exp(-net_k)}$$

For updating the weights, we need to calculate the error. This can be done by

$$E = \frac{1}{2}\sum_{i=1}^{k}(o_i - t_i)^2$$

o_i and t_i represents the real output and target output at neuron i in the output layer, respectively. If the error is less than a predefined limit, training process will stop; otherwise weights need to be updated. For weights between hidden layer and output layer, the change in weights is given by

$$\Delta w_{ij} = \alpha \delta_i h_j$$

Where α is a training rate coefficient that is restricted to the range [0.01,1.0],h_j is the output of neuron j in the hidden layer δ, and can be obtained by

$$\delta_i = (t_i - o_i)o_i(l - o_i)$$

Similarly, the change of the weights between hidden layer and output layer is given by

IJATES

$$\Delta w_{ij} = \beta \delta_{Hi} x_j$$

Where β is a training rate coefficient that is restricted to the range [0.001,1.0], x_j is the output of neuron j in the input layer, and δ_{Hi} can be obtained by

$$\delta_{Hi} = x_i(l - x_i) \sum_{j=1}^{k} \delta_i w_{ij}$$

x_i the output at neuron i in the input layer, and summation term represents the weighted sum of all δ_i values corresponding to neurons in output layer that obtained in equation. After calculating the weight change in all layers, the weights can be simply updated by

$$w_{ij}(new) = w_{ij}(old) + \Delta w_{ij}$$

This process is repeated, until the error reaches a minimum value[29].

For the efficient operation of the back-propagation neural network it is necessary for the appropriate selection of the parameters used for training. The initial weight will influence whether the net reaches a global or local minima of the error and if so how rapidly it converges. To get the best result, the initial weights are set to random numbers between -1 and 1[27, 28].

Training a Net; the motivation for applying back-propagation net is to achieve a balance between memorization and generalization; it is not necessarily advantageous to continue training until the error reaches a minimum value.

The weight adjustments are based on the training patterns. As along as the error for validation decreases training continues. Whenever the error begins to increase, the net is starting to memorize the training patterns. At this point, training is terminated. If the activation function can vary with the function, then it can be seen that an input, m output function requires at most 2n+1 hidden units. If more number of hidden layers are present, then the calculation for the δ_s are repeated for each additional hidden layer present, summing all the δ_s units present in the previous layer that is fed into the current layer for which δ_s is being calculated.

In back-propagation neural network, the weight change is in a direction that is a combination of current gradient and the previous gradient. A small learning rate is used to avoid major disruption of the direction of learning when very unusual pair of training patterns is presented.

5.4 TRAINING ALGORITHM

The training routine implemented in the following basic algorithm

1. Form network according to the specified topology parameters
2. Initialize weights with random values within the specified weight_bias value
3. load trainer set files (both input image and desired output text)
4. analyze input image and map all detected symbols into linear arrays
5. read desired output text from file and convert each character to a binary Unicode value to store separately
6. for each character :

a. calculate the output of the feed forward network

b. compare with the desired output corresponding to the symbol and compute error

c. back propagate error across each link to adjust the weights

7. move to the next character and repeat step 6 until all characters are visited
8. compute the average error of all characters
9. repeat steps 6 and 8 until the specified number of epochs

a. Is error threshold reached? If so abort iteration

b. If not continue iteration

5.5 TESTING

The testing phase of the implementation is simple and straightforward. Since the program is coded into modular parts the same routines that were used to load, analyze and compute network parameters of input vectors in the training phase can be reused in the testing phase as well.

The basic steps in testing input images for characters can be summarized as follows:

Algotithm:

- Load image file.
- Analyze image for character lines.
- For each character line detect consecutive character symbols.
- Analyze and process symbol image to map into an input vector.
- Feed input vector to network and compute output.
- Convert the Unicode binary output to the corresponding character and render to a text box .

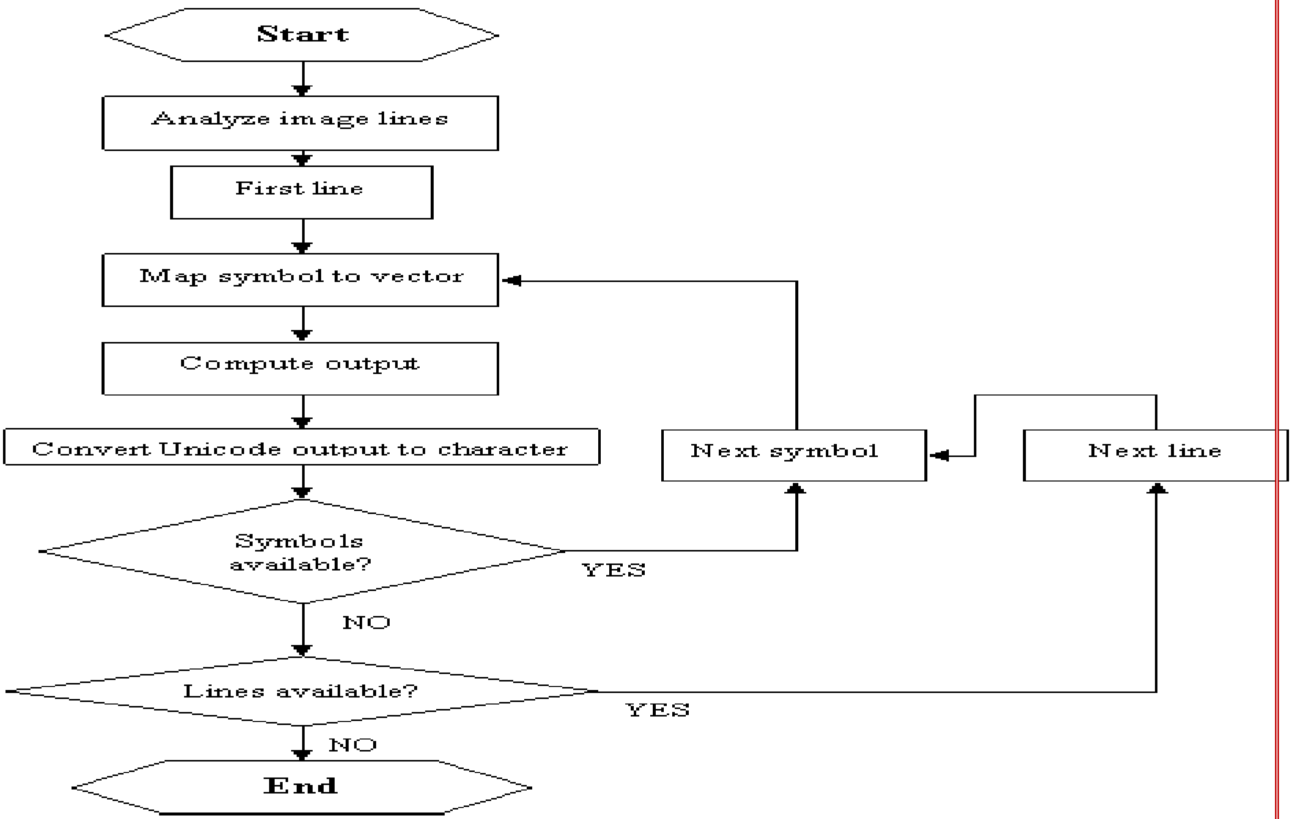

Figure 5.5: Flow chart of Algorithm implimentation

7. RESULTS

S No.	No. of Neurons	No. of Layers	Max Error	Performance
1	10	1	1.1	95.56
2	20	2	1.1	97.64
3	30	3	1.1	98.44

7. CONCLUSION

This paper proposed a best approach to identify the individuals based on their palm prints. Image processing operations are applied to extract the features of palm. Finally, neural network toolbox used for recognition training and verification the resulted vectors. The system shows effectiveness of results with accuracy around 98.44%. This work would detect a user is a member of a system or not. If he/she is a valid user of the system, then he/she is identified and the output is "Yes" If the user could not be identified by the system, it output is "No" Implementation of the program would result in a much secure and accurate system. The result shows that, image pre-processed plus the classifier here give the good accuracy results.

For future work, the improved median filter is developed and implemented to handle the problems faced by the standard median filter and to improve the mode of operation of the adaptive median filter to achieve effective results. In recent times vector machines are gaining a lot of importance for doing classification and prediction in the field of machine learning. Vector machine has been found place in bio informatics fertinity. For future scope some more bio-metric matrices with support vector machines will be used for recognition purposes.

7. REFERENCES

[1] D. Zhang, Wai-Kin Kong, J. You and Michael Wong, "online palmprint identification", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, pp. 1041-1050, Sept. 2003.

[2] A. Jain, R. Bolle and S. Pankanti (eds.), Biometrics: Personal Identification in Networked Society, Boston, Mass: Kluwer Academic Publishers, 1999.

[3] A. Morales, M. Ferrer and A. Kumar, "Improved Palmprint Authentication using Contactless Imaging", Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS), pp. 1-6, Sep. 2010.

[4]A. Kong, D. Zhang and M. Kamel, "A Survey of Palmprint Recognition", Published in: Journal of Pattern Recognition, vol. 42, pp. 1408-1418, July. 2009.

[5] N. Duta, A.K. Jain and K.V. Mardia, "Matching of palmprints", *Pattern Recognition Letters*, vol. 23, no. 4, pp. 477-485, 2002

[6] W. Shu and D. Zhang, "Automated personal identification by palmprint", *Optical Engineering*, vol. 38, no. 8, pp. 2359-2362, 1998

[7]David Zhang et al., Online Palmprint Identification, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 25, NO. 9, SEPTEMBER 2003, 1041-1050.

[8]Haruki Ota et al., Implementation of Remote System Using Touchless Palmprint Recognition Algorithm, Mobile Multimedia Security, ACM, 2010, 33-41.

[9] H B Kekre et al., Palmprint Recognition Using Kekre's Wavelet's Energy Entropy Based Feature Vector, International Conference & Workshop on Emerging Trends in Technology 2011, ACM, 39-45, 2011.

[10] Jiwen Lu, Erhu Zhang, Xiaobin Kang, YanxueXue, "Palmprint recognition using wavelet decomposition and 2D principal component analysis", International conference on Communications, Circuits and Systems Proceedings, vol. 3, pp. 2133-2136, June. 2006.

IJATES

[11] M. Sharkas, I. El-Rube and M.A. Mostafa, ”The Contourlet Transform with the Principal Component Analysis for Palmprint Recognition”, International conference on Computational Intelligence, Communication Systems and Networks (CICSyN), pp. 262-267, July. 2010.

[12] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, “Image quality assessment: From error to structural similarity”, *IEEE Trans. Image Processing*, vol. 13, pp. 600-612, 2004.

[13] J. Doi and M. Yamanaka, “Personal authentication using feature points on finger and palmar creases” *in Proceedings of 32nd Applied Imagery Patten Recognition Workshop*, pp. 282-287, 2003.

[14] C.C. Han, “A hand-based personal authentication using a coarse-to-fine strategy”, *Image and Vision Computing*, vol. 22, no. 11, pp. 909-918, 2004. [10] C.C. Han, H.L. Cheng, C.L. Lin and K.C. Fan, “Personal authentication.

[16] Chin-Chuan Han, “A hand-based personal authentication using a coarse-to-fine strategy”, Image Vision Computing, vol. 22, pp. 909-918, May 2004.

[17] J. Doublet, M. Revenu, and O. lepetit, “Robust Gray Scale Distribution Estimation for Contactless Palmprint Recognition", First IEEE international conference on biometrics: Theory, Applications, and Systems (BTAS), pp. 1-6, Sep. 2007.

[18] [A. Kumar and D. Zhang, “Integrating shape and texture for hand verification”, First IEEE symposium on Muli-Agent security and Survivabily, pp. 222-225, Dec. 2004.

[19] C. Poon, D. C. M. Wong and H. C. shen, “A New Method in Locating and Segmenting Palmprint into Region-of-Interest”, 17th International conference in Proceedings of the Pattern Recognition (ICPR), pp.533-536, 2004.

[20] T. Connie, A.T.B. Jin, M.G.K. Ong and D.N.C. Ling, “An automated palmprint recognition system”, *Image and Vision Computing*, vol. 23, no. 5, pp. 501-515, 2005. D. Zhang, W.K. Kong, J. You and M. Wong, “On-line palmprint identification”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1041-1050, 2003.

[21] R.C.Gonzalez and R.E. Wood, Digital Image Processing, Prentice-Hall, India, Second Edition, 2007.

[22] Hussain, Z.: ―Digital Image Processing – Practical Applications of Parallel Processing Techniques,‖ Ellis Horwood, West Sussex, UK, 1991.

[23] J.-S. Lee, ―Digital image enhancement and noise filtering by use of local statistics,‖ IEEE Trans. Patt. Anal. Mach. Intell., vol. 2, pp. 165-168,1980. 29.

IJATES

[24]M. A. Schulze and J. A. Pearce, ―Some properties of the two-dimensional pseudomedian filter,‖ in Nonlinear Image Processing II, E. R. Dougherty, G. R. Arce, and C. G. Boncelet, Jr., Editors, Proc. SPIE, vol. 1451, pp. 48- 57, 1991.

[25] Bernard Widrow and Samuel D. Steavns, ―Adaptive Signal Processing‖, Pearson Edition, 2000.

[26] D. Dhanasekaran and K. B. Bagan, ―High Speed Pipelined Architecture for Adaptive Median Filter‖, European Journal of Scientific Research, Vol. 29, No. 4, pp. 454-460, 2009

[27] S.Lawrence, C.L.Giles, A.C.Tsoi, and A.d.Back, "Face Recognition: A Convolutional Neural Network Approach ", (1993) "IEEE Transactions of Neural Networks. vol.8, no.1, pp.98-113.

[28] P.Latha, Dr.L.Ganesan and Dr.S.Annadurai, "Face Recognition using Neural Networks", Signal Processing: An International Journal (SPIJ) Volume (3) : Issue (4).

[29] W.S. Zheng, J.H. Lai, S.Z. Li, "1D-LDA vs. 2DLDA: When is vector-based linear discriminant analysis better than matrix-based?," Pattern Recognition. vol. 41, pp. 2156-2172, July 2008.

IJATES