Palm Vein Authentication using Image Classification Technique

Mr. Saagar Mukhopadhyay
M. Tech. Student
Department of Computer Science & Engineering
University of Calcutta, India

Prof. Samir Kumar Bandyopadhyay
Assistant Professor
Department of Computer Science & Engineering
University of Calcutta, India

Abstract

This paper presents an optimized palm vein authentication algorithm which will match the similar type of vascular pattern from the database given i.e. CASIA-Palmprint V1 dataset. . The palm vein authentication technology offers a high level of accuracy. Palm vein authentication uses the vascular patterns of an individual's palm as personal identification data. If we compare with a finger or the back of a hand, a palm has a broader and more complicated vascular pattern and thus contains a wealth of differentiating features for personal identification. The importance of biometrics in the current field of Security has been depicted in this work. Also, we have also outlined opinions about the utility of Biometric authentication systems. We have processed the raw image from the dataset before implementing authentication algorithm. After getting the suitable image after pre- processing, we have used local binary pattern (LBP) for feature extraction purpose & then using a machine learning algorithm, with support vector machine (SVM), we tried to match the vascular vein pattern for authentication. Result of the matching algorithm is not only optimized as per the proposed approach but also quite efficient.
Keywords: Palm Vein, Feature Extraction and Identification

I. Introduction

In the prevalent network society, in which anyone can easily access information anytime and anywhere, people are also faced with the risk that anyone can easily access information about other people anytime and anywhere. Passwords as a personal identification numbers and cards have been used before for personal identification. However, cards can be stolen, and password and numbers can be guessed or forgotten. To solve these problems biometric authentication technology is used. As the personal authentication technology which can distinguish between registered legitimate users and imposters is now attracting attention. Biometrics is automated methods of recognizing a person based on a physiological or behavioral characteristic. An example of or behavioral characteristic are face recognition, fingerprints, hand geometry, signature verification, iris, retinal, finger/hand/palm vein recognition, ear recognition, and voice recognition. Furthermore, academic and industry tried to develop a device that can catch the vascular patterns under the skin.

Palm vein recognition is a biometric authentication method based on the unique patterns of veins in the palms of people's hands. Palm vein recognition systems, like many other biometric technologies, capture an image of a target, acquire and process image data and compare it to a stored record for that individual. The world first "contactless vein authentication" technology developed by Fujitsu, and the leading edge technology known as "palm vein authentication". Fujitsu has developed a palm vein authentication technology that uses vascular patterns as personal identification data. Vein recognition technology is secure because the authentication data exists inside the body and so it is very difficult to forge and also highly accurate. This technology has many applications like in banking, hospitals, government offices, in passport issuing, libraries, personal computer, etc. Business growth will be achieved with these solutions by reducing the size of the palm vein sensor and shortening the authentication time. Palm vein authentication has high level of accurate because it is located inside the body and does not change over the life and cannot be stolen. Fundaments of biometrics are that they are things about a person.

- Measurable characteristics - things that can be counted numbered or otherwise quantified.
- Physiological characteristics - like height, eye colour, fingerprint, DNA etc.
- Behavioural characteristics - such as the way a person moves, walks, types of person.

The contactless palm vein authentication technology consists of image sensing and software technology. The palm vein is one of the most reliable physiological characteristics that can be used to distinguish between individuals. Palm vein technology works by identifying the vein patterns in an individual's palm. Also, the system is contactless and hygienic for use in public areas. It is more powerful than other biometric authentication techniques.

II. Related works

Biometrics authentication is a growing and controversial field in which civil liberties groups express concern over privacy and identity issues. Today, biometric laws and regulations are in process and biometric industry standards are being tested. Automatic recognition based on "who you are" as opposed to "what you know" (PIN) or "what you have" (ID card). Recognition of a person by his body & then linking that body to an externally established identity forms a very powerful tool for identity management Biometric Recognition. Below figure shows the different type of biometric authentication. Canadian airports started using iris scan in 2005 to screen pilots and airport workers. Pilots were initially worried about the possibility that repeated scans would negatively affect their vision but the technology has improved to the point where that is no longer an issue. Canada Customs uses an iris scan system called CANPASS-Air for low-risk travelers at Pearson airport.

Finger vein authentication, a new biometric method, utilizes the vein patterns inside one's fingers for personal identification. Vein patterns are different for each finger and for each person, and as they are hidden underneath the skin's surface, forgery is extremely difficult. These unique aspects of finger vein pattern recognition set it apart from previous forms of biometrics and have led to its adoption by the major Japanese financial institutions as their newest security technology [2-3].

Some researchers have shown the theoretical foundation and difficulties of hand vein recognition, at first. Then, the threshold segmentation method and thinning method of hand vein image are deeply studied and a new threshold segmentation method and an improved conditional thinning method are proposed. The method of hand vein image feature extraction based on end points and crossing points is studied initially, and the matching method based on distances is used to match vein images [4-6].

Another researchers proposed a biometric technique using hand-dorsa, extracting vein structures. For conventional algorithm, it is necessary to use high-quality images, which demand high-priced collection devices [7-9].

The proposed method makes using low-cost devices possible. The results shown that they could extract the vein networks as successfully as using high-quality images. Different biometric systems are shown in figure 1.

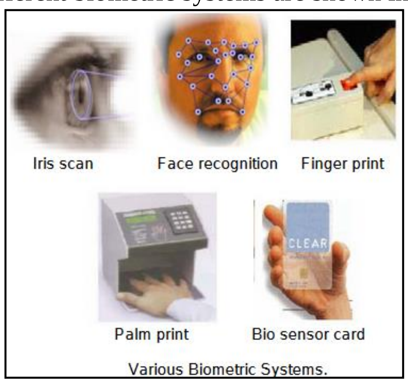

Fig. 1: Different Biometric Systems

III. PROPOSED WORK

The aim of the paper is to present an optimized palm vein authentication algorithm which will match the similar type of vascular pattern from the database given i.e. CASIA-Palmprint V1 dataset. . The dataset contained 312 unique palm images with left/right hand images of male/female persons & 16 images in each set with each 15 samples are of same type. The palm vein authentication technology offers a high level of accuracy. Palm vein authentication uses the vascular patterns of an individual's palm as personal identification data. If we compare with a finger or the back of a hand, a palm has a broader and more complicated vascular pattern and thus contains a wealth of differentiating features for personal identification. So, we have to process the raw image from the dataset taking a set sample data images before implementing authentication algorithm on it. After getting the processed images, we need to use a machine learning algorithm for authentication purpose from where we have to find the accuracy of the authentication system & that way we will know how much efficiency we will achieve from our result & also in future how much we need to improve the algorithm to get the better result.

JPEG stands for Joint Photographic Experts Group. It is a standard method of compressing photographic images. We also call JPEG the file format which employs this compression. The file extensions for this format are .JPEG, .JFIF, .JPG, OR .JPE although .JPG is the most common on all platforms.

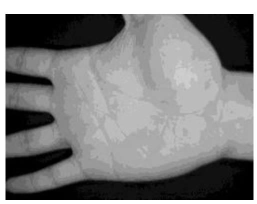

Fig. 2: Input Image [1]

JPEG/JFIF is mostly used for storing and transmitting photographs on the World Wide Web, but not as well suited for line drawings and other textual or iconic graphics because its compression method performs badly on these types of images. The MIME media type for JPEG is image/jpeg, except in older Internet Explorer versions, which provides a MIME type of image/pjpeg when uploading JPEG images. JPEG files usually have a filename extension of .jpg or .jpeg. JPEG/JFIF supports a maximum image size of 65,535×65,535 pixels, hence up to 4 giga-pixels for an aspect ratio of 1:1. The input image is shown in figure 2.

ROI processing defined as in an Image Processing Toolbox. A region of interest (ROI) is a portion of an image that we want to filter or perform some other operation on. We define an ROI by creating a binary mask, which is a binary image that is the same size as the image we want to process.

ROI is a selected subset of samples within a dataset identified for a particular purpose. The concept of a ROI is commonly used in many application areas. For example, in medical imaging, the boundaries of a tumour may be defined on an image or in a volume, for the purpose of measuring its size. ROI can be defined by creating a *binary mask*, where the binary image will be of same size as the image we want to process. In the mask image, the pixels that define the ROI are set to 1 and all other pixels set to 0. We can define more than one ROI in an image. The regions can be geographic in nature, such as polygons that encompass contiguous pixels, or defined by a range of intensities. In the latter case, the pixels are not necessarily contiguous. Much like healthcare reform introduces a holistic approach to patient care, a new hospital ROI reform mindset will allow marketers to provide a broader approach to measuring return on investment.

Here, in the paper we are working on palm vein pattern for authentication purpose. So, we need the main palm lines for further processing i.e. for matching algorithm. We don't need any other part of the palm images for processing, so we need to remove the less concerned part from the palm images. We will work with the fixed mask size i.e. 192x192 sized images for further processing, so that we won't have any problem regarding height x width of the images for matching algorithm. So, we fixed the mask size as specified then we have used Gaussian blur (also known as Gaussian smoothing) which gives the result of blurring an image by a Gaussian function.

It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. After that, we have used thresholding for binarizing the image from gray scale. Once we get the binary image, we have detected the boundary lines of the image, and then we locate the centroid of image from coordinate points i.e. from x & y points of the image. As we are getting the result of boundary value, we have taken the image inside the boundary & check the coordinate distance from each point of the image. After getting the distance, keep the coordinate sorted by taking three points at a time. We, then rotate the image as needed to get top & bottom coordinate point & with a fixed x- offset value, extract the desired image region as output. This is a pre-defined function for region of interest; we have modified it as per our need to get the desired region of interest of the palm images for further processing. The result of the region of interest of palm images provides the main palm lines which we are concerned with for matching purpose but we still need to process the image for better result from the authentication algorithm.

Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness & used for image segmentation and data extraction. Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in 1D signal is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

Canny, Sobel and prewitt proposed approaches which involve in convolution of images with specific kernels to detect edges in an image. We used Sobel edge detector in the dataset. Sobel operator, sometimes called the Sobel–Feldman operator or Sobel filter, is used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasizing edges. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel–Feldman operator is either the corresponding gradient vector or the norm of this vector. The Sobel–Feldman operator is based on convolving the image with a small, separable, and integer-valued filter in the horizontal and vertical directions and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation that it produces is relatively crude, in particular for high-frequency variations in the image.

The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives – one for horizontal changes, and one for vertical. We defined A as the source image, and G_x and G_y are two images which at each point contain the horizontal and vertical derivative approximations respectively, the computations are as follows,

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by,

$$|G| = \sqrt{Gx^2 + Gy^2}$$

Typically, an approximate magnitude is computed using,

$$|G| = |Gx| + |Gy|$$

This is much faster to compute. We used the Sobel mask for edge detection to detect main lines of palm images in our project.

Filtering is a technique for modifying or enhancing an image. For example, we can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement. Filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel. Digital images are prone to a variety of types of noise. Noise is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene. There are several ways that noise can be introduced into an image, depending on how the image is created.

After edge detection, the resultant image had some small connected components as noise, so we had to remove the small components from the processed image for further processing. So, we used a function which removes small connected lines from a binary image i.e. all connected components (objects) that have less than a fixed pixel value, producing another binary image as output which is filtered. The default connectivity is 8 for two dimensions, 26 for three dimensions, and more for higher dimensions. This operation is known as an area opening. We used 2D function here with intensity value 10 with default 8 connected objects to remove noise from the image & then we have stored the filtered image for further operation.

Feature extraction, a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector. This approach is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as image matching and retrieval. In machine learning, pattern recognition and in image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction involves reducing the amount of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power; also it may cause a classification algorithm to over fit to training samples and generalize poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

Feature extraction describes the relevant shape information contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure. In pattern recognition and in image processing, feature extraction is a special form of dimensionality reduction. The main goal of feature extraction is to obtain the most relevant information from the original data and represent that information in a lower dimensionality space. Feature extraction is done after the pre-processing phase in character recognition system. The primary task of pattern recognition is to take an input pattern and correctly assign it as one of the possible output classes. Common feature extraction techniques include Histogram of Oriented Gradients (HOG), Speeded up Robust Features (SURF), Local Binary Patterns (LBP), Haar wavelets, and colour histograms. We used Local Binary Patterns (LBP) as feature extraction technique on the pre- processed palm print images what we have stored after filtering.

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. Due to its discriminative power and computational simplicity, LBP texture operator has become a popular approach in various applications. It can be seen as a unifying approach to the traditionally divergent statistical and structural models of texture analysis. Perhaps the most important property of the LBP operator in real-world applications is its robustness to monotonic gray-scale changes caused, for example, by illumination variations. Another important property is its computational simplicity, which makes it possible to analyze images in challenging real-time settings. We have used general LBP algorithm for feature extraction in our project work before training the images of given dataset.

The Local Binary Patterns algorithm has its roots in 2D texture analysis. The basic idea is to summarize the local structure in an image by comparing each pixel with its neighborhood. For an example, take a pixel as centre and threshold its neighbors against. If the intensity of the centre pixel is greater-equal its neighbor, then denote it with 1 and 0 if not. We'll end up with a binary number for each pixel, just like 11001111. With 8 surrounding pixels we'll end up with 2^8 possible combinations, which are called Local Binary Patterns or sometimes abbreviated as LBP codes. The first LBP operator actually used a fixed 3 x 3 neighborhood just like this,

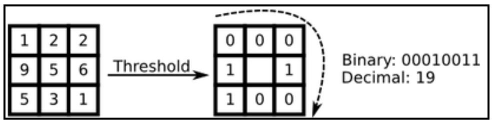

Fig. 3: Local Binary Patterns algorithm

Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be i.e. the process or action of proving or showing something to be true, genuine, or valid. Logically, authentication precedes authorization. Here, authentication is accompanied by an ability to "recall" or "recognize" a given pattern. In this project, the pattern was the path followed by the user (imagine a very linear path for straight up/down/left/right gestures while a curvy one for anything involving circular movements). Pattern matching was essential here and even before the given pattern could be matched to the computer's knowledge base, its knowledge base had to be constructed. This is where Machine Learning comes. We have used a popular image classification technique called Support Vector Machine as per our system needs. We trained it over hundreds of samples, so that the system was capable was of detecting and understanding subtle differences in movements for the same gesture performed by different users.

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can change when exposed to new data. The process of machine learning is similar to that of data mining. Both systems search through data to look for patterns. However, instead of extracting data for human comprehension -- as is the case in data mining applications -- machine learning uses that data to detect patterns in data and adjust program actions accordingly. Machine learning algorithms are often categorized as being supervised or unsupervised in image classification. Supervised algorithms can apply what has been learned in the past to new data. Unsupervised algorithms can draw inferences from datasets. Image classification refers to the task of extracting information classes from a multiband raster image. The resulting raster from image classification can be used to create thematic maps. The goal of machine learning is to optimize differentiable parameters so that a certain loss/cost function is minimized. Machine learning can be used in both image processing and computer vision but it has found more use in computer vision than in image processing. In machine learning the loss function can have a physical meaning in which case the features learnt can be quite informative but this is not necessarily the case for all situations.

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. A support vector machine builds a hyper plane or set of hyper planes in a high- or infinite dimensional space, used for classification. Good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class (functional margin), generally larger the margin lower the generalization error of the classifier. SVM uses Nonparametric with binary classifier approach and can handle more input data very efficiently. Performance and Accuracy depends upon the hyper plane selection and kernel parameter.

It uses a technique called the kernel trick to transform the data and then based on these transformations it finds an optimal boundary between the possible outputs. Simply put, it does some extremely complex data transformations, and then figures out how to separate the data based on the labels or outputs we've defined. The kernel trick is used to get the features for the mapping to identify the pattern in case of non- linear dataset. So, it needed to process the dataset with the non-linear functions (curved boundaries are needed). But there are infinitely many of these cases can arrive to identify the pattern from the given dataset. From that, SVM uses a function as a solution, is called kernel trick. A kernel function maps pairs of data points onto their inner products (i.e., they work like distance functions).A feature space based on a kernel function has one dimension for every pair of data points. Mathematical minimization can then be used to find the max-margin hyperplane in the feature-space. The effect is to identify a non-linear (curved) boundary in the original data space. A mapping example is shown in figure 3.

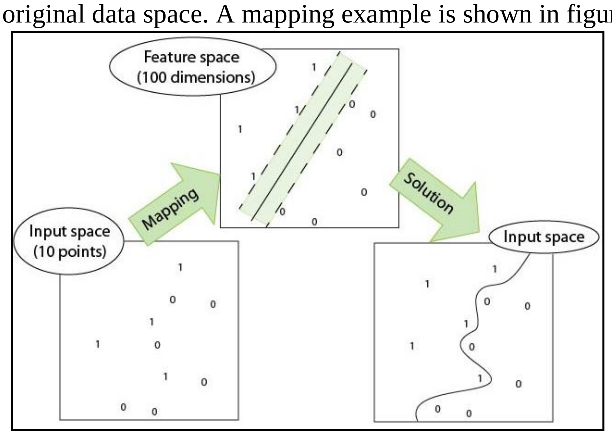

Fig. 4: Mapping Process

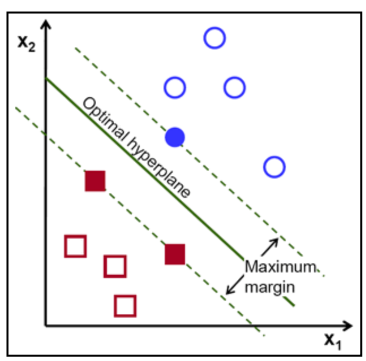

Fig. 5: Separating hyper plane maximizes the margin of the training data

In using a kernel function, we are moving from the original data space to a space that has one dimension for every pair of original points. Manipulating points in the feature space then has the effect of `stretching' or `compressing' areas of the data space. This can be a way of `pulling' differently classified data points apart, or `pushing' same-class points together. This is the unique trick using SVM. This kernel functions have been getting a lot of attention. But their practical value remains unclear at this stage. Derivation of weights for a separating hyperplane may still be best done using iterative error-correction.

Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labelled training data (supervised learning), the algorithm outputs an optimal hyper plane which categorizes new marginal distance. The operation of the SVM algorithm is based on finding the hyper plane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of margin within SVM's theory. The optimum separation hyperplane

(OSH) is the linear classifier with the maximum margin for a given finite set of learning patterns. The OSH computation with a linear support vector machine is presented in this section. Therefore, the optimal separating hyper plane *maximizes* the margin of the training data as shown in figure 4.

In the paper, we have three data sets: training, validation and testing. We train the classifier using 'training set', tune the parameters using 'validation set' and then test the performance of your classifier on unseen 'test set'. An important point to note is that during training the classifier only the training and/or validation set is available. The test set must not be used during training the classifier. The test set will only be available during testing the classifier. There is no 'one' way of choosing the size of training/testing set and people apply heuristics such as 10% testing and 90% training. However, doing so can bias the classification results and the results may not be generalizable. A well accepted method is N-Fold cross validation, in which you randomize the dataset and create N (almost) equal size partitions. Then choose N^{th} partition for testing and N-1 partitions for training the classifier. Within the training set you can further employ another K-fold cross validation to create a validation set and find the best parameters. And repeat these process N times to get an average of the metric. Since we want to get rid of classifier 'bias' we repeat this above process M times (by randomizing data and splitting into N fold) and take average of the metric. Cross-validation is almost unbiased, but it can also be misused if training and validation set comes from different populations and knowledge from training set is used in the test set.

For training data, to study the process, observe the behavior of the random process for some time, collecting a large number of sample (x_1,y_1), (x_2,y_2),…, (x_N,y_N). In the example where it's needed to consider, each sample would consist of a phrase x containing the words surrounding *in*, together with the translation y of *in* which the process produced. For now we can imagine that these training samples have been generated by a human expert who was presented with a number of random phrases containing *in* and asked to choose a good translation for each. The training set is used to fit the models; the validation set is used to estimate prediction error for model selection; the test set is used for assessment of the generalization error of the final chosen model. Ideally, the test set should be kept in a "vault," and be brought out only at the end of the data analysis.

Cross-validation, sometimes called rotation estimation, is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (or first seen data) against which the model is tested (testing dataset).

The goal of cross validation is to define a dataset to "test" the model in the training phase (i.e., the validation dataset), in order to limit problems like over fitting, give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem), etc. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. One of the main reasons for using cross-validation instead of using the conventional validation (e.g. partitioning the data set into two sets of 70% for training and 30% for test) is that there is not enough data available to partition it into separate training and test sets without losing significant modelling or testing capability. In these cases, a fair way to properly estimate model prediction performance is to use cross-validation as a powerful general technique. Cross-validation combines (averages) measures of fit (prediction error) to derive a more accurate estimate of model prediction performance.

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once. For classification problems, one typically uses stratified k-fold cross-validation, in which the folds are selected so that each fold contains roughly the same proportions of class labels. In repeated cross-validation, the cross-validation procedure is repeated n times, yielding n random partitions of the original sample. The n results are again averaged (or otherwise combined) to produce a single estimation. Figure 5 shows K- fold cross validation with k=4 to show how it really works for validating the datasets.

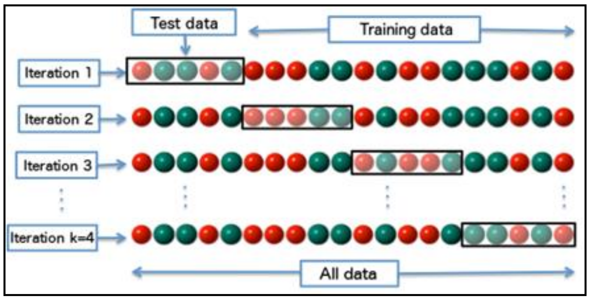

Fig. 6: Shows K- fold cross validation with k=4

In many areas of information science, finding predictive relationships from data is a very important task. Initial discovery of relationships is usually done with a training set while a test set and validation set are used for evaluating whether the discovered relationships hold. More formally, a training set is a set of data used to discover potentially predictive relationships. A test set is a set of data used to assess the strength and utility of a predictive relationship. Test and training sets are used in intelligent systems, machine learning, genetic programming and statistics.

There are some statistical measure like Sensitivity and specificity to check the performance of the binary classification test, also known in statistics as classification function, where some measuring elements are,

- True positive (TP) = correctly identified
- False positive (FP)= incorrectly identified
- True negative(TN) = correctly rejected
- False negative(FN) = incorrectly rejected

Sensitivity (also called the true positive rate, the recall, or probability of detection in some fields) measures the proportion of positives that are correctly. It tells us how likely the test is come back positive in someone who has the characteristic. This is calculated as TP/ (TP+FN).

Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified. It tells us how likely the test is to come back negative in someone who does not have the characteristic. This is calculated as TN/ (TN+FP). Low sensitivity & high specificity defines optimized & efficient result with lower type / error rate.

Finally, Accuracy of the test result is calculated as (TP+TN)/ (TP+FP+TN+FN).

Here, we have chosen a linear binary classifier where we have considered class label 0 for left palm images & class label 1 for right palm images as decision making both for training & testing the dataset over linear support vector machine. For cross validation, we have taken few different CV fold values to check the range of our implemented algorithm & for validating the images which means, it runs up to N times as per the CV fold value taken to evaluate the mean average from the given dataset. In our dataset, we have 15 set of same type of palm images consisting 16 palm images in each set. So, we have taken around 192 images for training & testing purpose. In between, we have used 128 around images for training the data images & 64 sample images for testing the dataset to identify the matching accuracy. The result is shown in a table manner in next section where we have achieved accuracy percentage of our implementation for training & testing purpose with different CV fold values, also the accuracy of sensitivity & specificity value from where we get to know about biometric identification i.e. how correctly our authentication algorithm performed. The following process is used for performing the entire task.

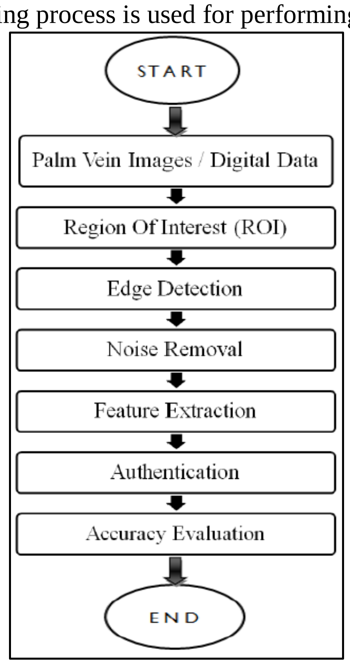

Fig. 7: Basic Flowchart for performing Entire Task

A. Algorithm

In the paper the entire process is implemented in two module i.e. pre- processing & post- processing of images. So, below we have depicted the algorithms.

1) Pre-Processing Steps

- Start
- Input data images
- Call find ROI to get the ROI images
- Filtering the images

- Edge detected using Sobel operator
- Remove small components with area opening from all connected components that have fewer than pixels value 10
- Store the pre-processed images for further processing
- End

2) Find ROI
- Start
- Input raw images
- Smoothing the images with Gaussian filter
- Convert the images into binary
- Detect the boundary of the images
- Locate the centroid of the images from x & y coordinate
- Find the distance from each coordinate taking 3 axis points at a time
- Rotate the images with x coordinate off set value 150 to get top & bottom point of the images
- Extract 192x192 as fixed mask size from the images as region of interest
- End

3) Training Steps
- Start
- Input pre- processed images for training
- Use LBP feature extraction to normalize the data
- Reshape the images as per fixed mask size & train images
- Validating data with k-fold with Cross Validation
- Check the data up to k-th time for the train images
- After validating the images using SVM model with linear kernel function test the images with train images
- End

4) Testing Steps
- Start
- Input pre- processed images for testing
- Use LBP feature extraction to normalize the data
- Reshape the images as per fixed mask size & train images for testing
- Validating data with k- fold with Cross Validation
- Check the data up to k-th time for the test images
- After validating the images using SVM model with linear kernel function test the images with previously train images
- Show the training accuracy
- Show the testing accuracy
- Show the accuracy of Sensitivity
- Show the accuracy of Specificity
- End

IV. RESULTS

The processed image results from the input dataset & accuracy assessment result is shown below which we have evaluated from the program code as an output from our implementation. Pre-processed images are shown in figure 7.

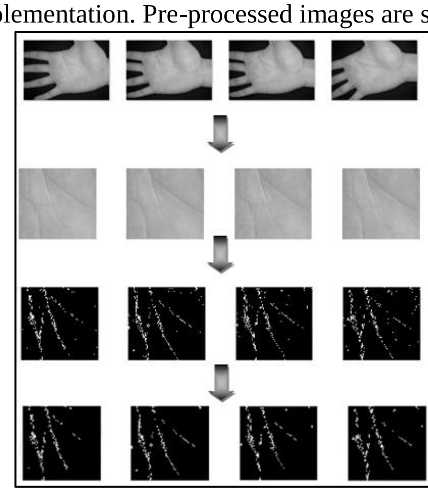

Fig. 8: Results are shown based on Proposed Algorithm

After post-processing, one sample accuracy assessment result what we have seen with different CV fold values as an output from our implementation is shown below.

```
Command Window
>> testing

Accuracy_of_training =

    0.9375


Accuracy_of_testing =

    0.9063


Sensitivity_of_testing =

    0.8387


Specificity_of_testing =

    0.9697

fx >>
```

Fig. 9: Implementation

The different test accuracy result with multiple CV fold values are given in table below what we have achieved from our test result. This result may change with multiple instances over different iteration.

Table – 1
Different Test Accuracy Result

CV with Different K- Fold Values	*Accuracy Result(in percentage)*			
	Training	*Testing*	*Sensitivity*	*Specificity*
K=7	*92.19*	*89.06*	*80.65*	*93.85*
K=8	*93.75*	*90.43*	*83.87*	*96.97*
K=9	*95.31*	*92.19*	*87.10*	*96.97*
K=10	*96.09*	*93.75*	*90.32*	*96.97*

So, the maximum identification/authentication accuracy is 96.09% from training instance & 93.75% from the testing instance with the given dataset from our implementation after verifying the algorithm multiple times as per our proposed work.

V. FUTURE SCOPE

After the implementation of our proposed work we have identified the palm images for authentication as required. The identification/matching algorithm can be optimized further more. So, we will try to perform the below modification & implementation for better efficiency in future.

1) As, we have used only LBP feature extraction. So, we will use different feature extraction algorithms to check how well it can perform.
2) We have used SVM as a linear image classification technique for authentication. So, in future we will use other classification approach like Artificial Neural Network, Fuzzy logic etc.
3) Also, we will try to implement the algorithm which can compare between False Acceptance Rate (FAR) and False Rejection Rate (FRR) in our biometric authentication work to check the efficiency of our implemented algorithm.

VI. CONCLUSIONS

This paper implemented a biometric authentication system i.e. palm vein authentication. This technology is highly secure because it uses information contained within the body and is also highly accurate because the pattern of veins in the palm is complex and unique to each individual. Moreover, its contactless feature gives it a hygienic advantage over other biometric authentication technologies. Palm vein matching systems rely on the uniqueness of vein patterns in each person, which differ more markedly from one individual to another than the patterns of fingerprints and iris scanning. Palm recognition is considered a strong form of the inherence biometric authentication factor and is currently used or being considered for an increasing number of user identification and authentication applications, including online and onsite authentication, automobile security, employee time and attendance tracking, computer and network authentication, healthcare identification, end point security and ATM machines. We have processed the authentication system as proposed in our project which gives an optimized output as a matching result from the given database & this is quite a very efficient system for authentication.

REFERENCES

[1] Dataset CASIA-PalmprintV1 image database, http://biometrics.idealtest.org/
[2] Z. Wei, X. Qiu, Z. Sun, and T. Tan, "Counterfeit iris detection based texture analysis", Proc. Int. Conf. Pattern Recognition, Tampa, 2008.
[3] Shigeru Sasaki, and Akira Wakabayashi, "Business Expansion of Palm Vein Pattern Authentication Technology", Fujitsu Sci. Tech. J, 2005.
[4] Y. Zhou and A. Kumar, "Human identification using palm-vein images," IEEE Trans. Inf. Forensics Security, 2011.

[5] Ajay Kumar, David C. M. Wong, Helen C. Shen, Anil K. Jain, "Personal Verification using Palm print and Hand Geometry Biometric", Springer-Verlag Berlin Heidelberg, 2003.
[6] Daniel Hartung, Martin Aastrup Olsen, Haiyun Xu, Christoph Busch, "Spectral Minutiae for Vein Pattern Recognition", Biometrics (IJCB) IEEE Conference, 2011.
[7] Debnath Bhattacharyya, Poulami Das, Tai-hoon Kim, Samir Kumar Bandyopadhyay ,"Vascular Pattern Analysis towards Pervasive Palm Vein Authentication", Journal of Universal Computer Science, vol. 15, 2009.
[8] Gin-Der Wu and Zhen-Wei Zhu "An enhanced discriminability recurrent fuzzy neural network for temporal classification problems", Science Direct, 2013.
[9] Goh Kah Ong Michael, Tee Connie, Andrew Teoh Beng Jin," Design and Implementation of a Contactless Palm Print and Palm Vein Sensor", Control Automation Robotics & Vision (ICARCV) IEEE Conference, 2010.