Real-Time Finger-Vein Recognition System

S. Nivas[1], P. Prakash[2]

[1]Head of Department, Maharaja Co-Educational Arts and Science College, Erode- India

[2]Research Scholar, Maharaja Co-Educational Arts and Science College, Erode- India

E-mail- prkshmca@gmail.com

ABSTRACT

As the need for personal authentication increases, many people are turning to biometric authentication as an alternative to traditional security devices. Concurrently, users and vendors of biometric authentication systems are searching for methods to establish system performance. However, most existing biometric systems have high complexity in time or space or both. In this paper, we propose a novel Finger-Vein recognition algorithm for authentication. It consists of two phases. Enrolment phase and verification phase. Both stages start with finger-vein image pre-processing, which includes detection of the region of interest (ROI), image segmentation, alignment, and enhancement. For the enrolment stage, after the pre-processing and the feature extraction step, the finger-vein template database is built. For the verification stage, the input finger-vein image is matched with the corresponding template after its features are extracted. Feature Selection process is based on SURF algorithm.

Keywords

(Finger-Vein recognition is Enrolment phase and verification phase both stages vein image pre-processing after the feature extraction step, the finger-vein template database is built verification stage; the input finger-vein image is matched with the corresponding template after its features are extracted)

INTRODUCTION:

What is a Biometric?

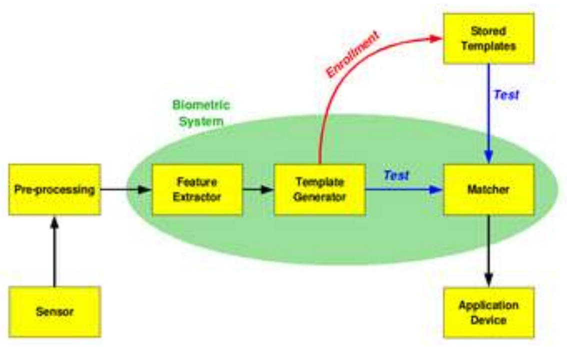
Biometrics(or biometricauthentication) refers to the identification of humans by their characteristics or traits. Biometrics is used in computer science as a form of identification and access control. It is also used to identify individuals in groups that are under surveillance.

Biometric identifiers are the distinctive, measurable haracteristics used to label and describeindividuals. Biometric identifiers are often categorized as physiological versus behavioral characteristics. Physiological characteristics are related to the shape of the body. Examples include, but are not limited to fingerprint, face recognition, DNA, Palm print, hand geometry, iris recognition, retina and odour/scent. Behavioral haracteristics are related to the pattern of behavior of a person, including but not limited to: typing rhythm, gait, and voice. Some researchers have coined the term behaviometrics to describe the latter class of biometrics.

More traditional means of access control include token-based identification systems, such as a driver's license or passport, and knowledge-based identification systems, suchaspassword or personal identification number. Since biometric identifiers are unique to individuals, they are more reliable in verifying identity than token and knowledge-based methods; however, the collection of biometric identifiers raises privacy concerns about the ultimate use of this information.

Many different aspects of human physiology, chemistry or behavior can be used for biometric authentication. The selection of a particular biometric for use in a specific application involves a weighting of several factors. Jain *et al.* (1999) identified seven such factors to be used when assessing the suitability of any trait for use in biometric authentication. **Universality** means that every person using a system should possess the trait. **Uniqueness** means the trait should be sufficiently different for individuals in the relevant population such that they can be distinguished from one another. **Permanence** relates to the manner in which a trait varies over time. More specifically, a trait with 'good' permanence will be reasonably invariant over time with respect to the specific matching algorithm. **Measurability** (collectability) relates to the ease of acquisition or measurement of the trait. In addition, acquired data should be in a form that permits subsequent processing and extraction of the relevant feature sets. **Performance** relates to the accuracy, speed, and robustness of technology used (see performance section for more details).**Acceptability** relates to how well individuals in the relevant population accept the technology such that they are willing to have their biometric trait captured and assessed. **Circumvention** relates to the ease with which a trait might be imitated using an artifact or substitute.

No single biometric will meet all the requirements of every possible application.



The basic block diagram of a biome in the following two modes.[3] In verification mode the system performs a one-to-one comparison of a captured biometric with a specific template stored in a biometric database in order to verify the individual is the person they claim to be. Three steps involved

in person verification. In the first step, reference models for all the users are generated and stored in the model database. In the second step, some samples are matched with reference models to generate the genuine and impostor scores and calculate the threshold. Third step is the testing step. This process may use a smart card, username or ID number (e.g. PIN) to indicate which template should be used for comparison. 'Positive recognition' is a common use of verification mode, "where the aim is to prevent multiple people from using same identity".

In Identification mode the system performs a one-to-many comparison against a biometric database in attempt to establish the identity of an unknown individual. The system will succeed in identifying the individual if the comparison of the biometric sample to a template in the database falls within a previously set threshold. Identification mode can be used either for 'positive recognition' (so that the user does not have to provide any information about the template to be used) or for 'negative recognition' of the person "where the system establishes whether the person is who she (implicitly or explicitly) denies to be".The latter function can only be achieved through biometrics since other methods of personal recognition such as passwords, PINs or keys are ineffective.

The first time an individual uses a biometric system is called *enrollment*. During the enrollment, biometric information from an individual is captured and stored. In subsequent uses, biometric information is detected and compared with the information stored at the time of enrollment. Note that it is crucial that storage and retrieval of such systems themselves be secure if the biometric system is to be robust. The first block (sensor) is the interface between the real world and the system; it has to acquire all the necessary data. Most of the times it is an image acquisition system, but it can change according to the characteristics desired. The second block performs all the necessary pre-processing: it has to remove artifacts from the sensor, to enhance the input (e.g. removing background noise), to use some kind of normalization, etc. In the third block necessary features are extracted. This step is an important step as the correct features need to be extracted in the optimal way. A vector of numbers or an image with particular properties is used to create a *template*. A template is a synthesis of the relevant characteristics extracted from the source. Elements of the biometric measurement that are not used in the comparison algorithm are discarded in the template to reduce the filesize and to protect the identity of the enrollee.

If enrollment is being performed, the template is simply stored somewhere (on a card or within a database or both). If a matching phase is being performed, the obtained template is passed to a matcher that compares it with other existing templates, estimating the distance between them using any algorithm (e.g. Hamming distance). The matching program will analyze the template with the input. This will then be output for any specified use or purpose (e.g. entrance in a restricted area). Selection of biometrics in any practical application depending upon the characteristic measurements and user requirements. We should consider Performance, Acceptability, Circumvention, Robustness, Population coverage, Size, Identity theft deterrence in selecting a particular biometric. Selection of biometric based on user requirement considers Sensor availability, Device availability, Computational time and reliability, Cost, Sensor area and power consumption.

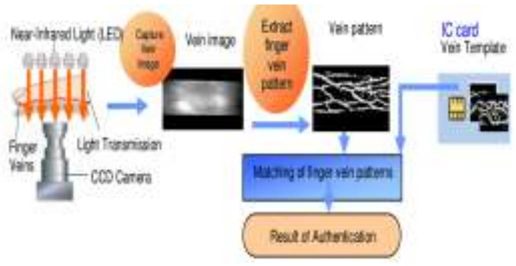
3.1SYSTEMIMPLEMENTATION:

3.1.1 Finger Vein Recognition:

Finger vein recognition is a method of biometric authentication that uses pattern-recognition techniques based on images of human finger vein patterns beneath the skin's surface. Finger vein recognition is one of many forms of biometrics used to identify individuals and verify their identity.

To obtain the pattern for the database record, an individual inserts a finger into an attester terminal containing a near-infrared LED (light- emitting diode) light and a monochrome CCD (charge-coupled device) camera. The hemoglobin in the blood absorbs near-infrared LED light, which makes the vein system appear as a dark pattern of lines. The camera records the image and the raw data is digitized, certified and sent to a database of registered images. For authentication purposes, the finger is scanned as before and the data is sent to the database of registered images for comparison. The authentication process takes less than two seconds.

Blood vessel patterns are unique to each individual, as are other biometric data such as fingerprints or the patterns of the iris. Unlike some biometric systems, blood vessel patterns are almost impossible to counterfeit because they are located beneath the skin's surface. Biometric systems based on fingerprints can be fooled with a dummy finger fitted with a copied fingerprint; voice and facial characteristic-based systems can be fooled by recordings and high-resolution images. The finger vein ID system is much harder to fool because it can only authenticate the finger of a living person.



3.2 FEATURES USED:

3.2.1 SURF Features:

SURF (Speeded Up Robust Features) is a robust local feature detector, first presented by Herbert Bay et al. in 2006, that can be used in computer vision tasks like object recognition or 3D reconstruction. It is partly inspired by the SIFT descriptor. The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT. SURF is based on sums of 2D Haar wavelet responses and makes an efficient use of integral images.

It uses an integer approximation to the determinant of Hessian blob detector, which can be computed extremely quickly with an integral image (3 integer operations). For features, it uses the sum of the Haar wavelet response around the point of interest. Again, these can be computed with the aid of the integral image.

An application of the algorithm is patented in the US. The task of finding correspondences between two images of the same scene or object is part of many computer vision applications. Camera calibration, 3D reconstruction, image registration, and object recognition are just a few. The search for discrete image correspondences – the goal of this work – can be divided into three main steps. First, 'interest points' are selected at distinctive locations in the image, such as corners, blobs, and T-junctions. The most valuable property of an interest point detector is its repeatability, i.e. whether it reliably finds the same interest points under different viewing conditions. Next, the neighbourhood of every interest point is represented by a feature vector. This descriptor has to be distinctive and, at the same time, robust to noise, detection errors, and geometric and photometric deformations. Finally, the descriptor vectors are matched between different images. The matching is often based on a distance between the vectors, e.g. the Mahalanobis or Euclidean distance. The dimension of the descriptor has a direct impact on the time this takes, and a lower number of dimensions is therefore desirable.

A wide variety of detectors and descriptors have already been proposed in the literature. Also, detailed comparisons and evaluations on benchmarking datasets have been performed. While constructing our fast detector and descriptor, we built on the insights gained from this previous work in order to get a feel for what are the aspects contributing to performance. In our experiments on benchmark image sets as well as on a real object recognition application, the resulting detector and descriptor are not only faster, but also more distinctive and equally repeatable.

When working with local features, a first issue that needs to be settled is the required level of invariance. Clearly, this depends on the expected geometric and photometric deformations, which in turn are determined by the possible changes in viewing conditions. Here, we focus on scale and image rotation invariant detectors and descriptors. These seem to over a good compromise between feature complexity and robustness to commonly occurring deformations. Skew, anisotropic scaling, and perspective effects are assumed to be second-order effects that are covered to some degree by the overall robustness of the descriptor.

As also claimed by Lowe, the additional complexity of full affine-invariant features often has a negative impact on their robustness and does not pay off, unless really large viewpoint changes are to be expected. In some cases, even rotation invariance can be left out, resulting in a scale-invariant only version of our descriptor, which we refer to as 'upright SURF' (U-SURF). Indeed, in quite a few applications, like mobile robot navigation or visual tourist guiding, the camera often only rotates about the vertical axis. The benefit of avoiding the overkill of rotation invariance in such cases is not only increased speed, but also increased discriminative power. Concerning the photometric deformations, we assume a simple linear model with a scale factor and offset. Notice that our detector and descriptor don't use colour.

3.2.2 Lacunarity:

Lacunarity, from the Latin lacuna meaning "gap" or "lake", is a specialized term in geometry referring to a measure of how patterns, especially fractals, fill space, where patterns having more or larger gaps generally have higher lacunarity. Beyond being an intuitive measure of gappiness, lacunarity can quantify additional features of patterns such as "rotational invariance" and more generally, heterogeneity. This is illustrated in Figure 1 showing three fractal patterns. When rotated 90°, the first two fairly homogeneous patterns do not appear to change, but the third more heterogeneous figure does change and has correspondingly higher lacunarity.

3.2.3 Measuring Lacunarity:

In many patterns or data sets, lacunarity is not readily perceivable or quantifiable, so computer-aided methods have been developed to calculate it. As a measurable quantity, lacunarity is often denoted in scientific literature by the Greek letters Λ or λ but it is important to note that there is no single standard and several different methods exist to assess and interpret lacunarity.

3.2.4 Box Counting Lacunarity

One well-known method of determining lacunarity for patterns extracted from digital images uses box counting, the same essential algorithm typically used for some types of fractal analysis.[1][4] Similar to looking at a slide through a microscope with changing levels of magnification, box counting algorithms look at a digital image from many levels of resolution to examine how certain features change with the size of the element used to inspect the image. Basically, the arrangement of pixels is measured using traditionally square (i.e., box-shaped) elements from an arbitrary set of E sizes, conventionally denoted ϵs. For each ϵ, the box is placed successively over the entire image, and each time it is laid down, the number of pixels that fall within the box is recorded.[note 1] In standard box counting, the box for each ϵ in E is placed as though it were part of a grid overlaid on the image so that the box does not overlap itself, but in sliding box algorithms the box is slid over the image so that it overlaps itself and the "Sliding Box Lacunarity" or SLac is calculated.[3][6] Figure 2 illustrates both types of box counting.

3.2.5 Relationship to the Fractal Dimension

Lacunarity analyses using the types of values discussed above have shown that data sets extracted from dense fractals, from patterns that change little when rotated, or from patterns that are homogeneous, have low lacunarity, but as these features increase,[clarification needed] so generally does lacunarity. In some instances, it has been demonstrated that fractal dimensions and values of lacunarity were correlated,[1] but more recent research has shown that this relationship does not hold for all types of patterns and measures of lacunarity. Indeed, as Mandelbrot originally proposed, lacunarity has been shown to be useful in discerning amongst patterns (e.g., fractals, textures, etc.) that share or have similar fractal dimensions in a variety of scientific fields including neuroscience.

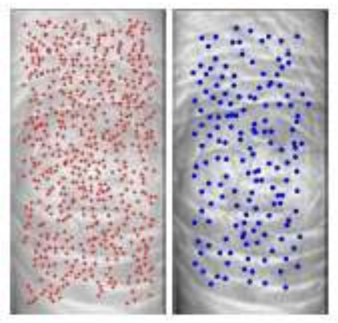


Fig 3.1 Surf feature pints to detect the finger vein propertied

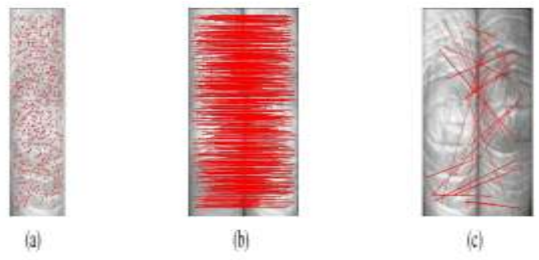


Fig 3.2 Shows the matched feature Points of the surf Algorithm

3.3 HARRIS CORNER

Corners are image locations that have large intensity changesin more than one directions.

Shifting a window in any direction should give a large change in intensity detect feature points, also called keypoints match feature points in different images The nearest neighbor is defined as the keypointwith minimum Euclidean distance for the invariant descriptor

Our a_ne invariant interest point detector is an a_ne-adapted version ofthe Harris detector. The a_ne adaptation is based on the second moment matrixand local extrema over scale of normalized derivatives. Locations of interest points are detected by the a_ne-adapted Harris detector. For initial- ization, approximate localizations and scales of interest points are extracted by the multi-scale Harris detector. For each point we apply an iterative procedure which modi_es position as well as scale and shape of the point neighbourhood. This allows to converge toward a stable point that is invariant to a_netrans- formations. This detector is the main contribution of the paper. Furthermore, we have developed a repeatability criterion which takes into account the point position as well as the shape of the neighbourhood. A quantitative comparison with existing detectors shows a signi_cant improvement of our method in the presence of large a_ne transformations. Results for wide baseline matching and recognition based on our a_ne invariant points are excellent in the presence of signi_cant changes in viewing angle and scale and clearly demonstrate their invariance.

3.3.1 Filtration Process

The presence of the fragments in an image is determined bythe combined use of a similarity measure and a detection threshold. Using a sliding window over the image, we measure the presence of the fragment in the window with normalized cross-correlation, a common method used in computer vision to measure visual similarity, and compare the score to a threshold.

3.4 MSER ((Maximally Stable Extremal Regions) Detection:

For region detection invariance transformations that should be considered are illumination changes, translation, rotation, scale and full affine transform (i.e. a region should correspond to the same pre-image for different viewpoints. Viewpoint changes can be locally approximated by affine transform if assuming locally planar objects and orthographic camera, that is perspective effects ignored)

3.4.1Analyzing Minimal regions:

Here we detect anchor points (f.e. using Harris detector for corners). Anchor points detected at multiple scales are local extremas of intensity – explore image around rays from each anchor point. Go along every ray starting from this point until an extremum of function f is reached.

MSER is a method for blob detection in images. The MSER algorithm extracts from an image a number of co-variant regions, called MSERs: an MSER is a stable connected component of some gray-level sets of the image .

MSER is based on the idea of taking regions which stay nearly the same through a wide range of thresholds. All the pixels below a given threshold are white and all those above or equal are black. If we are shown a sequence of thresholded images It with frame t corresponding to threshold t, we would see first a black image, then white spots corresponding to local intensity minima will appear then grow larger. These white spots will eventually merge, until the whole image is white. The set of all connected components in the sequence is the set of all extremal regions. Optionally, elliptical frames are attached to the MSERs by fitting ellipses to the regions. Those regions descriptors are kept as features.Sweep threshold of intensity from black to white, performing a simple luminance thresholding of the imageExtract connected components ("Extremal Regions") Find a threshold when an extremal region is "Maximally Stable", i.e. local minimum of the relative growth of its square. Due to the discrete nature of the image, the region below above may be coincident with the actual region, in which case the region is still deemed maximal. Approximate a region with an ellipse (this step is optional)Keep those regions descriptors as features

4. ALGORITHM:

4.2.1 Feature extraction:

The method of feature extraction is described in this section, s the intensity of the pixel s the intensity of the pixel Rf is the set of pixels within the finger's outline, and Tr is s the locus space. Suppose the pixel in the lower left in the image to be (0,0), the positive direction of the x-axis to be rightward in the image, the positive direction of the y-axis to be upward within the image, and Tr(x, y)to be initialized to 0.

Step 1: Determination of the start point for line tracking and the moving-direction attribute

Step 2: Detection of the direction of the dark line and movement of the tracking point

Step 3: Updating the number of times points in the locus space have been tracked.

Step 4: Repeated execution of step 1 to step 3 (N times).

Step 5: Acquisition of the finger-vein pattern from the locus space.

The details of each step are described below.

Step 1: Determination of the start point for line tracking and the moving-direction attribute.

The start point for line tracking is (xs, ys), a pair of uniform random numbers selected from Rf. That is, the initial value of the current tracking point (xc, yc) is (xs, ys). After that, the moving-direction attribute Dlr, Dud is determined. Dlr, Dud are the parameters that prevent the tracking point from following a path with excessive curvature. Dlrand Dud are independently determined as follows:

$$D_{lr} = \begin{cases} (1,0) & (\text{if } R_{nd}(2) < 1). \\ (-1,0) & (\text{otherwise}); \end{cases}$$

$$D_{ud} = \begin{cases} (0,1) & (\text{if } R_{nd}(2) < 1). \\ (0,-1) & (\text{otherwise}), \end{cases}$$

whereRnd(n) is a uniform random number between 0 and n.

Step 2-1: Initialization of the locus-position table TcThe positions that the tracking point moves to are storedin the locus-position table, Tc. The table is initialized in thisstep.

Step 2-2: Determination of the set of pixels Ncto which the current tracking point can move

A pixel to which the current tracking point (xc, yc) moves must be within the finger region, have not been a previous (xc, yc) within the current round of tracking, and be one of the neighboring pixels of (xc, yc). Therefore, Ncis determined as follows:

$$N_c = \overline{T_c} \cap R_f \cap N_r(x_c, y_c),$$

whereNr(xc, yc) is the set of neighboring pixels of (xc, yc), selected as follows:

$$N_r(x_c, y_c) = \begin{cases} N_3(D_{lr})(x_c, y_c) & (\text{if } R_{nd}(100) < p_{lr}); \\ N_3(D_{ud})(x_c, y_c) & (\text{if } p_{lr}+1 \le R_{nd}(100) < p_{lr}+p_{ud}); \\ N_8(x_c, y_c) & (\text{if } p_{lr}+p_{ud}+1 \le R_{nd}(100)), \end{cases}$$

where N8(x, y) is the set of eight neighboring pixels of a pixel (xc, yc) and N3(D)(x, y) is the set of three neighboring pixels of (xc, yc) whose direction is determined by the moving-direction attribute D (defined as (Dx,Dy)). N3(D)(x, y) can be described as follows:

$$\begin{aligned} N_3(\boldsymbol{D})(x,y) = \{ & (D_x+x,\ D_y+y), \\ & (D_x-D_y+x,\ D_y-D_x+y), \\ & (D_x+D_y+x,\ D_y+D_x+y)\}. \end{aligned}$$

Parameters plrand pudin Eq. 4 are the probability of selecting the three neighboring pixels in the horizontal or vertical direction, respectively, as Nr(sc, yc). The veins in a finger tend to run in the direction of the finger's length. Therefore, if we increase the probability thatN3(Dlr)(xc, yc) is selected as Nr(xc, yc), we obtain a faithful representation of the pattern of finger veins. In preliminary experiments, excellent results are produced when plr= 50 and pud= 25.

Step 2-3: Detection of the dark-line direction near the current tracking point

To determine the pixel to which the current tracking point (xc, yc) shouldmove, the following equation, referred to as the line-evaluation function, is calculated. This reflects the depth of the valleys in the cross-sectional profiles around the current tracking point

$$V_l = \max_{(x_i,y_i)\in N_c} \Big\{ F(x_c+r\cos\theta_i-\frac{W}{2}\sin\theta_i,\ y_c+r\sin\theta_i+\frac{W}{2}\cos\theta_i) + F(x_c+r\cos\theta_i+\frac{W}{2}\sin\theta_i,\ y_c+r\sin\theta_i-\frac{W}{2}\cos\theta_i) - 2F(x_c+r\cos\theta_i,\ y_c+r\sin\theta_i)\Big\},$$

whereWis the width of the profiles, r is the distance between (xc, yc) and the cross section, and θiis the angle between the line segments (xc, yc) − (xc + 1, yc) and (xc, yc) − (xi, yi).

In this paper, in consideration of a thickness of the veins that are visible in the captured images, these parameters are set at W = 11 and r = 1.

Step 2-4: Registration of the locus in the locus-position table Tcand moving of the tracking point

The current tracking point (xc, yc) is added to the locuspositiontable Tc. After that, if Vlis positive, (xc, yc) is thenupdated to (xi, yi) where Vlis maximum.

Step 2-5: Repeated execution of steps 2-2 to 2-4

If Vlis positive, go to step 2-2; if Vlis negative or zero, leave step 2 and go to step 3, since (xc, yc) is not on the dark line.

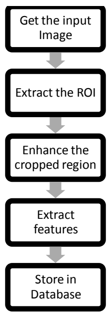
Step 3: Updating the number of times points in the locus space have been tracked Values of elements in the locus space Tr(x, y) are incremented ∀(x, y) ∈Tc.

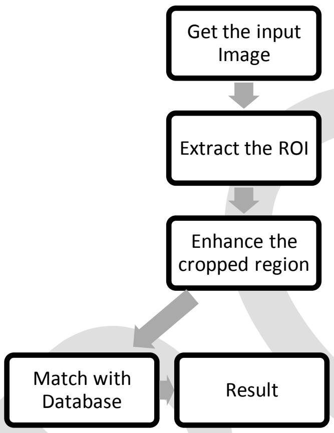
Step 4: Repeated execution of steps 1 to 3 (N times) Steps 1 to 3 are thus executed N times. If the number of repetitions N is too small, insufficient feature extraction is performed. If, on the other hand, N is too big, computational costs are needlessly increased. Through an experiment, we

determined that N = 3000 is the lower limit for sufficient feature extraction.

Step 5:Acquisition of the pattern of veins from the locus space The total number of times the pixel (x, y) has been the current tracking point in the repetitive line tracking operation is stored in the locus space, Tr(x, y). Therefore, the fingervein pattern is obtained as chains of high values of Tr(x, y).

5. FLOW CAHRT





6. SCREEN SHOT:

6.1 Enrolment Phase:

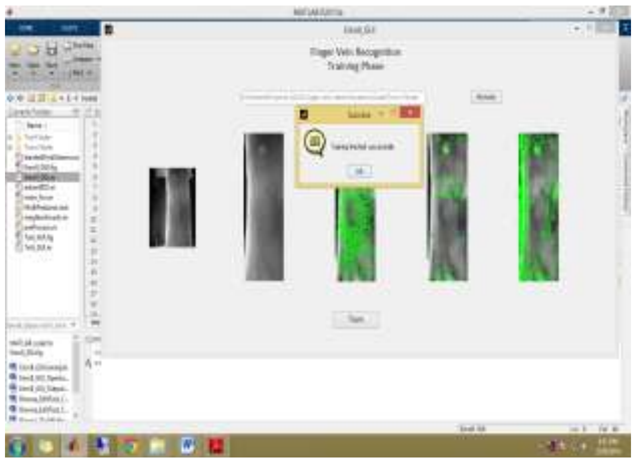
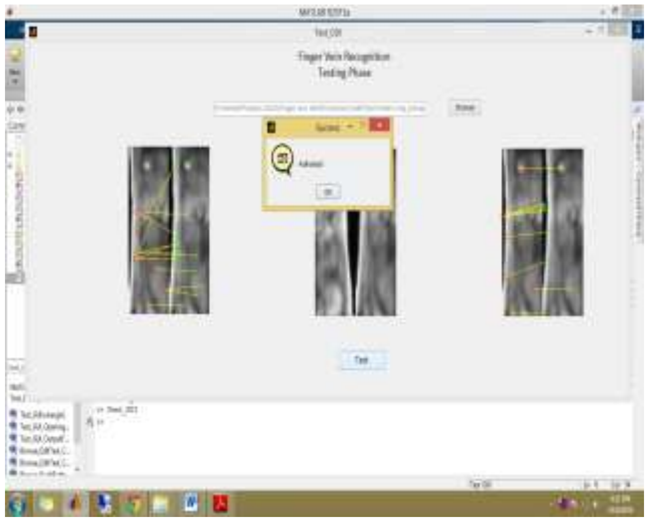
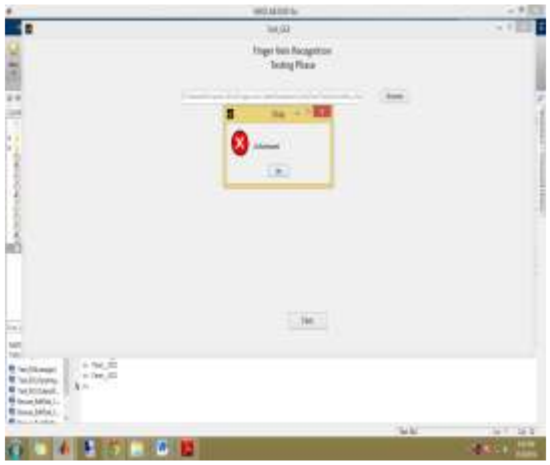


Fig (1) Enrolment phase

6.2 Testing Phase:





Fig(2.1)Authorizedvein,2.2 Unauthorized person

7. CONCLUSION

The present study proposed an end-to-end finger-vein recognition system based on the blanket dimension. The proposedsystem includes a device for capturing finger-vein images, amethod for ROI segmentation, and a novel method combiningblanket dimension features and lacunarity features and surf feature for recognition algorithm An approach to correct the non-uniform brightnessand to improve the contrast is proposed. During recognition, the corresponding featuresare matched using nearest-neighbourhood-ratio method with SURF matching scores are fused using weighted sum rule toobtain fused matching score. It is observed that the system performs with CRRof atleast 98:62%.

8. FUTURE WORK:

The Full proposed work of enhancement has been predicting the feature points with more accuracy that can be implemented using SIFT algorithm in future and we can classify the each part of the stages using neural networks which results the high percentage rate of predicting the outcomes of the system in future.

REFERENCES:

[1] A. K. Jain, S. Pankanti, S. Prabhakar, H. Lin, and A. Ross, "Biometrics:a grand challenge", Proceedings of the 17th International Conference onPattern Recognition (ICPR), vol. 2, pp. 935-942, 2004.

[2] P. Corcoran and A. Cucos, "Techniques for securing multimedia content in consumer electronic appliances using biometric signatures," IEEE Transactionson Consumer Electronics, vol 51, no. 2, pp. 545-551, May 2005.

[3] Y. Kim, J. Yoo, and K. Choi, "A motion and similarity-based fake detection method for biometric face recognition systems," IEEE Transactions onConsumer Electronics, vol.57, no.2, pp.756-762, May 2011.

[4] D. Wang , J. Li, and G. Memik, "User identification based on fingerveinpatterns for consumer electronics devices", IEEE Transactions onConsumer Electronics, vol. 56, no. 2, pp. 799-804, 2010.

[5] H. Lee, S. Lee, T. Kim, and HyokyungBahn, "Secure user identification for consumer electronics devices," IEEE Transactions on ConsumerElectronics, vol.54, no.4, pp.1798-1802, Nov. 2008.

[6] D. Mulyono and S. J. Horng, "A study of finger vein biometric for personal identification", Proceedings of the International SymposiumBiometrics and Security Technologies, pp. 134-141, 2008.

[7]. Anil K. Jain, Patrick Flynn, and Arun A. Ross. Handbook of Biometrics. Springer-Verlag, USA, 2007