IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 12, December 2017

An Alternative Approach to Fingerprint Hash Code Generation based on Modified Filtering Techniques

[1] Krishna Prasad K, [2] P.S. Aithal
[1] Research Scholar, College of Computer and Information Science, Srinivas University,Mangaluru-575001, India
[2] Vice Chancellor, College of Computer and Information Science, Srinivas University,Mangaluru-575001, India

Abstract: ***Fingerprint unique Hash code and template protection are the new technologies in biometric identification and verification system. Fingerprint hashing is the new technique which combines biometrics and cryptography. The modern study reveals that fingerprint is not so secured like secured passwords which consist of alphanumeric characters, number and special characters. Fingerprint Hash code acts as a key, which can uniquely identify every person. So it can be replaceable with user-id or username and can work along with text-based or picture based or pattern based passwords. In this paper, a fingerprint Hash code is generated using a novel Contrast Adjustment algorithm, modified segmentation algorithm, and Gabor filtering. The Hash code is generated from the extracted features of the grayscale fingerprint image using MD5 Algorithm. Fingerprint Hash code is not used for full security or authentication purpose but it can be combined with other security elements like password or OTP in order to enhance security. This study makes use of fingerprint Hash code as a unique key for human identification purpose.***

Keywords- ***Fingerprint Hash code, Gabor Filtering, Contrast Adjustment algorithm, Segmentation, MD5 Algorithm***

I. INTRODUCTION

Automatic Fingerprint Identification System (AFIS) contain the use of automatically and reliably enhance the image, reduce the noise and extract the minutiae features from the biometric images of the fingerprint. The performance of a minutiae extraction principle relies heavily on the pleasant quality of the input biometric image. The automatic fingerprint identification system consists of preprocessing, enhancement, segmentation, thinning, feature extraction, post-processing, minutiae orientation and alignment as its different stages or subprocess [1-9].

Contrast adjustment methods are extensively used for image processing to attain wider dynamic range and which is considered as preprocessing stage, especially in Automatic recognition system based on different types of images like a fingerprint, face, iris etc. When brightness is too high all the pixels of the image turn into lighter, conversely when the brightness is too low all the pixels of the image turn into darker. When the intensity is a too high, lighter area of the image becomes lighter and darker area of the image becomes An essential and important step in order to obtain high quality and performance rate at all types of image is through accurate segmentation. Fingerprint segmentation is the one of the main process involved in fingerprint pre-processing and it refers to the process of dividing or separating the image into two disjoint regions as the foreground and background. The foreground also called as Region of Interest (ROI) because only the region which contains ridge and valley structure is used for processing, while the background contains noisy and irrelevant content and that will be discarded in later enhancement or orientation or classification process.

Some of the most common types of segmentation algorithms are, TV-L1 based Adaptive Total Variation Model [11], TV-l2 based Directional Total Variation Model [12], Method based on a combination of ridge orientation and ridge frequency characteristics using orientation tensor approach [13], Orientation field is combined with the statistical characteristics of the gray to form new method [14], Ridge orientation Method based on Ridge Temple using correlation with a sinusoid [15], and the coherence, the mean, the variance as three pixel features methods.

One of the important challenges in biometric identification or verification system is keeping the biometric data or template safe and secure. A Hash function is usually transformed functions, which converts or transform data or features from one form to another. Always transform function should be the one-way function or another way it should not be invertible. Symmetric Hash functions for biometric fingerprints are some hash function, which is independent of the order in which input is presented to the system or invariant to translation and rotation [16]. In literature, few methods are already proposed by different researchers for building cancellable biometric template. In this regard, there are mainly two techniques, out of which one is error correcting code and another one is noninvertible transformation.

In this paper initially τ- Tuning Based Filtering Algorithm is used to improve grayscale fingerprint image contrast. The later image is converted into binary image and image is segmented based on Surfeit clipping based segmentation algorithm. From the segmented binary image, fingerprint features are extracted using Gabor filtering techniques, using angular and frequency variations. Finally, these features are converted into Hash code using the MD5 hash algorithm. To implement this algorithm MATLAB2015a is used by considering input from FVC ongoing 2002 benchmark dataset. The remaining part of the paper is organized as follows. Section II describes relative to research on the contrast adjustment, segmentation, and Gabor Filtering. Section III describes Research objective and Methodology. Section IV describes the tuning based Contrast Adjustment algorithm. Section V describes Surfeit Based segmentation algorithm. Section VI describes features extraction using Gabor filter. Section VII describes Results and Discussion. Section 8 concludes the paper.

II. RELATED RESEARCH

Equalization through Histogram (HE) is a very famous approach for image contrast adjustment or enhancement in image processing. In general, the histogram equalization distributes pixel values consistently and produces an outcome in a superior image with the linear increasing histogram. Some useful applications of HE enhancement consist of scientific image processing, speech recognition, fingerprint identification and texture synthesis, which might be typically employed with histogram adjustment [17-20].

Different techniques of making use of histogram equalization are determined in the literature. Global histogram equalization or GHE (Gonzalez & Woods, 2002) [21] makes use of the entire information of the input image to map into new distinct intensity levels of the image. Although this Global technique is suitable for ordinary or general enhancement, it fails to consider with the local brightness capabilities of the entered image. The gray ranges with very excessive frequencies (wide variety of occurrences) dominate over the opposite gray levels having decrease frequencies in an image. In any such situation, GHE remaps the gray levels in a way that the contrast stretching turns into confined in some dominating gray levels having large image histogram components, and it causes sizable contrast loss for other small ones.

Local histogram equalization (LHE) can overcome the problem encountered in GHE (Gonzalez & Woods, 2002) [21]. LHE uses a small window that slides on all pixel of the image sequentially and handiest the block of pixels that fall within this window are taken into consideration for HE and then gray level mapping for enhancement is carried out for the center pixel of that window. Therefore, it may make splendid use of local information also. But, LHE requires excessive computational cost and occasionally reasons over enhancement in some part of the image. Another shortfall of this approach is that it also enhances the noises inside the input image. To overcome the problem of high computational cost one more approach is to use the non-overlapping block for HE (Gonzalez & Woods, 2002; K. Krishna Prasad & Aithal P. S., 2017) [21 & 1]. But almost all times this method produces checkerboard effect.

In literature, there are many studies available, which mainly focuses on fingerprint image segmentation. Researchers, Mehtre, B. M., & Chatterjee, B. (1989) classified the image into blocks, which is administrative specific and the size was 16 × 16 pixels. Based on the gradient distribution, each block was classified. This method is best suited for simple fingerprint images which contain only background and foreground. Later Researchers Mehtre and Chatterjee (1989) [22] extended this work by leaving the grayscale variance, which will usually be lower than some threshold value. Researchers Ratha, N. K., Chen, S., & Jain, A. K. (1995) [23] proposed 16 × 16 blocks of classes and each one was developed based on the gray scale variance in the direction opposite to the orientation of ridges.

The authors Jain, Ratha, & Lakshmanan (1997) [24] concentrated for the detection of objects located in complex backgrounds. The given object is first applied to a bank of even-symmetric Gabor filters. The output image received from the Gabor filter is subjected to a sigmoid function transformation. The yield image of the Gabor filter is applied as an input to the clustering algorithm, which develops spatially compact clusters. Sun and Ai (1996) [25] pre-processed initially fingerprint image by converting it into a binary image with the help of dynamic threshold value (T). Moayer and Fu (1975) [26] used sampling squares, which are obtained from the subdivision of fingerprint images for the ultimate goal of feature extraction. They used dynamic threshold value (T) to convert the initial image to a binary image. In order to determine the local threshold value, researchers used neighbor pixels by group 5 × 5 pixels.

Naji, Ramli, Ali, R., Rahman, and Ali, M.L. (2002) [27] developed a segmentation algorithm, which computerized or automated the method of selecting a threshold value at the time of segmentation with the aid of histogram equalizer. Segmentation algorithm generally falls under two categories of machine learning techniques as supervised learning and unsupervised learning. Unsupervised learning uses threshold decided on detecting features to cluster the image. Supervised learning uses a simple linear classifier to classify features as a region of interest (ROI) or background and foreground. As a part of supervised methods, Alonso-Fernandez, Fierrez-Aguilar, & Ortega-Garcia, (2005) [28], used a Gabor filter to filter the

IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 12, December 2017

input image and to obtain a smooth image. The neural network can also be used in the segmentation process to reduce the noise or to enhance the image quality.

III. RESEARCH OBJECTIVE AND METHODOLOGY

The most of the research work in fingerprint identification system fails to provide template protection with main characteristics like revocability, diversity, non-invertible, and permanence. Most of the fingerprint identification algorithms or techniques are not able to match or recognize partial fingerprints. These two are the motivation for this study. The research gap identified in this study is listed below.

- When fingerprints are easily mimicable, what is the use of developing Hash code Translation and Rotation or orientation change invariant?
-
- Is it possible to compare and match fingerprint with only one Hash code stored in the database?
- Is there any possibility of using a fingerprint as identity-key or index-key with the aid of Hash code, without capturing through sensors every time, by considering the image captured at the beginning or onetime only (using a static image of the fingerprint)?

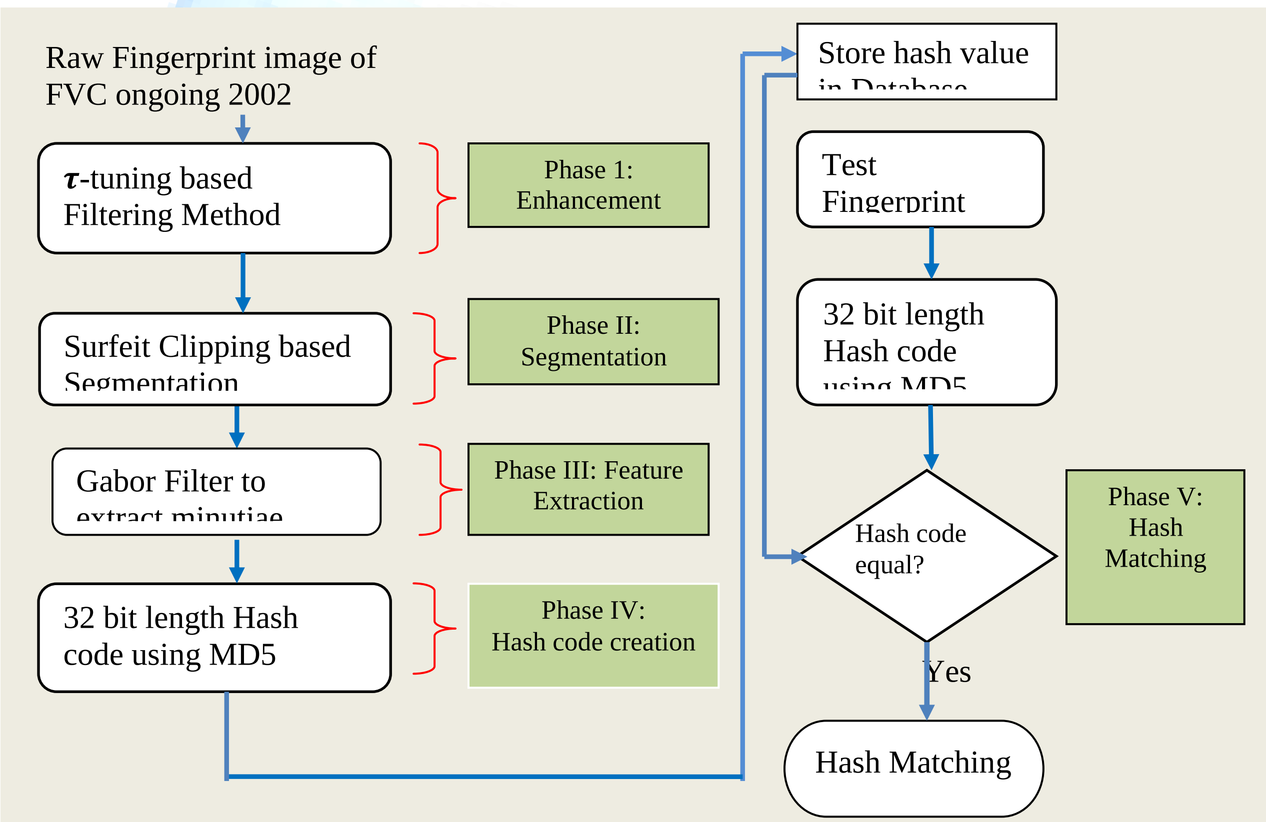


Fig. 1: Proposed Methodologies

The main objective of the research is given below.

- To Study a Fingerprint Hash code based on MD5 Hash Algorithm, using Gabor filter which includes modified filtering techniques, Contrast adjustment filtering, and Segmentation.

Here two-dimensional 64 × 64 sized Gabor Filter is used to extract features directly from segmented image without performing thinning process. The proposed work is implemented using MATLAB2015a. FVC ongoing 2002 benchmark dataset are used for training and test purpose. The methodology used in this study is shown in *Fig. 1.*

IV. TUNING BASED CONTRAST ADJUSTMENT ALGORITHM

The input for this algorithm is row image referred as I, and final output will be I_{filter} . Initially maximum intensity value of the image is found. We consider here 256 × 256 sized grayscale image. If the input fingerprint image is greater than this size then it will be converted into 256 × 256 sized grayscale image. The maximum intensity value in a 256 × 256 sized grayscale image is 255. The range of values is 0 to 255, which means that minimum value is 0 and maximum value is 255. Maximum intensity value of the image is represented as max (I). Each pixel intensity value is compared with max (I).

If pixel value is equal to max (I), then that pixel is assigned to ρ_{max} . The ρ_{max} is individual count of maximum intensity value. The total count of ρ_{max} is represented using lower case delta symbol δ_{max} and is calculated as follows [10].

$$\delta_{max} = \frac{\sum \rho_{max}}{R\times C} \qquad (1)$$

In (1) R, and C, are total number of rows and columns respectively. $\sum \rho_{max}$, indicates all pixels, whose intensity value is equal to maximum intensity value of the grayscale fingerprint image (I). Next minimum intensity values of the grayscale image are found and are referred as min (I). If pixel value is equal to min (I), then that pixel is assigned to ρ_{min} . The ρ_{min} is individual count of minimum intensity value of the image. The total count of ρ_{min} is represented using lower case delta symbol δ_{min} and is calculated as follows.

$$\delta_{min} = \frac{\sum \rho_{min}}{R\times C} \qquad (2)$$

As like (1) in (2), R, and C, is total number of rows and columns respectively. $\sum \rho_{min}$, indicates all pixels, whose intensity value is equal to intensity value of the grayscale fingerprint image (I).

Each row of the intensity matrix of the image is considered as a window and is represented as δ_w, which is expressed as

$$\delta_w = \delta_{max} \left(\frac{\delta(l)-\delta_{min}}{\delta_{max}-\delta_{min}}\right)^{\epsilon} \qquad (3)$$

In (3) ϵ value is 0.5, which is a constant. $\delta(l)$ is low or minimum value of each row. The difference value of $\delta(l) - \delta_{min}$ is divided by $\delta_{max} - \delta_{min}$. The quotient is multiplied by δ_{max} .

∂_w which is almost equal to Histogram equalization, cumulative density function, of window l is represented using 'tho' or partial derivative symbol and is defined as

$$\partial_w = \sum_{l=0}^{l_{max}} \frac{\delta_w(l)}{\sum \delta_w} \qquad (4)$$

In (4) $\sum \delta_w$ represents summation value of all window l or summation of δ_w. $\sum \delta_w$ is calculated as follows

$$\sum \delta_w = \sum_{l=0}^{l_{max}} \delta_w(l) \qquad (5)$$

The final output of this proposed algorithm (I_{filter}) is obtained using following equation

$$I_{filter} = \left(l_{max} \times \left(\frac{I(i,j)}{l_{max}}\right)\right)^{\tau} \qquad (6)$$

In (6), tau (τ) is an important value, which filters or maps input pixel intensity value to new intensity value in the output image and is defined as

$$\tau = round(1 - \max(\partial_w(:))) \qquad (7)$$

The (7) is rounded to 6 decimal points to get higher precision or accuracy.

The output of the robust tuning based algorithm (proposed method), I_{filter} is converted from grayscale 256 × 256 uint8 to double type for the purpose of grayscale image adjustment. The 256 × 256 double image consists of only two intensity values as 0 and 1. 0 represents dark and 1 represents bright or 0 dark black and 1 bright white. Here in image enhancement we focus more on Robust τ- Tuning Based Filtering Algorithm and in concluding part of the enhancement we just convert the output of this phase to just double type with an ultimate goal to achieve grayscale image adjustment.

Tuning Based Filtering Algorithm [10]

Input: Raw Image; I
Output: Filtered Output Image, I_{filter}
Step-1: for i=1 to R \\ R→ Row size of input image
Step-2: for j=1 to C \\ C→ Column size of input image
Step-3: if I(i,j)= =max(I)
Step-4: $\rho_{max} = I(i,j)$; end if;end for
Step-5: $\delta_{max} = \frac{\sum \rho_{max}}{R\times C}$
Step-6: for i=1 to R \\ R→ Row size of input image

Step-7: for j=1 to C \\ C→ Column size of input \\ image

Step-8: if I(i,j) = = min(I)

Step-9: $\rho_{min} = I(i,j)$; end if;end for

Step-10: $\delta_{min} = \frac{\sum \rho_{min}}{R \times C}$

Step-11: $\delta_w = \delta_{max} \left(\frac{\delta(l) - \delta_{min}}{\delta_{max} - \delta_{min}}\right)^{\epsilon}$ \\ ϵ→Constant; $\epsilon = 0.2$

Step-12: $\partial_w = \sum_{l=0}^{l_{max}} \frac{\delta_w(l)}{\sum \delta_w}$

Step-13: $\sum \delta_w = \sum_{l=0}^{l_{max}} \delta_w(l)$

Step-14: $\tau = round(1 - \max(\partial_w(:)))$ \\round to 6 decimal \\points

Step-15: $I_{filter} = (l_{max} \times (\frac{I(i,j)}{l_{max}}))^{\tau}$

V. SURFEIT BASED SEGMENTATION ALGORITHM

This algorithm considers Enhanced fingerprint image and produces a good quality segmented image. Let $I_{enhanced}$ be the enhanced image using robust τ-tuning based filtering method. The enhanced image is converted to binary image and stored as I_{binary} . Initially to find the edges of the I_{binary} image efficiently canny edge detection method is used. Canny edge detection finds the edges of the image through different processes, which includes, smoothing, locating gradients, non-maximum suppression, double thresholding, and edge tracking by using hysteresis. Smoothing of the image is done with the help of convolution, which blurs the image to get rid of noise. Canny edge detection uses double thresholding in order to find edges of the image. The result of the canny edge detection method is stored as I_{canny} . Next, the edge detected image, I_{canny} is converted into low resolution image by converting 256 × 256 sized grayscale image to 128 × 128 sized grayscale image.

$$I_{LRE} = I_{binary}(i \times 2, j \times 2)$$

The low resolution image is represented as I_{LRE}. In the next phase I_{LRE} image is padded with zeros using pad array and usually for simplicity in this method we use pad array size is eight and is referred as P. For $I_{LRE,}$ eight zeros are added in row and column respectively, and it enhanced to 144 × 144 sized grayscale image, which is denoted as I_{Parray} .

The I_{Parray} is clipped into 15×15 sized image and processed. The clipped image is stored in temp1. The entire 225 pixels of temp1 are reshaped as 1×225 matrix and denoted as temp2. The covariance of the matrix of the image, temp2 is calculated and if it is less than the threshold then the pixel of the I_{binary} (256 × 256 size) image is considered as not a part of ROI or foreground. Covariance of a matrix is calculated by considering row as observations and columns as random variables. Every pixel of the I_{binary} image is traced like this and marked as either foreground or background of the image based on covariance value. If it is greater than the threshold value then the pixel is considered to be foreground, means which is real part of the fingerprint image. Each time when padarray is considered, this takes into account one pixel out of 128×128 low resolution image and two pixels out of 256 ×128 sized image.

As the algorithm name suggests surfeit, means maximum, we discard maximum background part of the image by checking whether all the pixels of the each column intensity value sum becomes 256. If the column sum is 256 means all the pixels of that particular column contains intensity value 1. This signifies that this column contains background of the image. If any one column intensity value sum leads to value less than 256, which signifies that the particular column contains part of the foreground or ROI of the image. Then we skip the iteration and count considering starting of the column pixel for output of the segmented image from just previous to that column number. Same process we repeat from the last column to first column in reverse direction and stop moving backward until we get a column number sum of intensity value less than 256 for the purpose of finding last column number, which contains at least one pixel of foreground pixel. This means that from the last column to till this position image contains only background part of the image. The above-mentioned method repeated for rows also. So that it eliminates background or white blank area in left edge, right edge, top edge and bottom edge regions.

Surfeit Clipping based Segmentation Algorithm [6]

Input: binary image, I_{binary}

Output: Segmented Image, $I_{segment}$

Step-1: Read I_{binary} image

Step-2: Apply canny edge formation to the I_{binary} and store it in a variable I_{canny}

Step-3: for i=1 to floor(R/2)

Step-4: for j=1 to floor (C/2)

Step-5: $I_{LRE} = I_{binary}\ (i \times 2, j \times 2)$; end for loop; I_{LRE} →Low Resolution Edge Image

Step-6: $[R_{LRE}, C_{LRE}] = size(I_{LRE})$

Step-7: P=floor (max (15, 15)/2+1);

Step-8:I_{Parray} = padarray(I_{LRE}, $[P\ P]$)

Step-9: for i=P+1 to R_{LRE}+P

Step-10: for j=P+1 to C_{LRE}+P

Step-11: temp1=I_{Parray} $\left(\left(i - floor\left(\frac{15}{2}\right) : i + floor\left(\frac{15}{2}\right)\right), \left(j - floor\left(\frac{15}{2}\right) : j + floor\left(\frac{15}{2}\right)\right)\right)$

Step-12: temp2=reshape (temp1, 1, 225)

Step-13: $V_1 = covariance(temp2)$

Step-14: if $V_1 < T$ \\ T→Threshold

\\value=0.101

Step-15: $I_{binary}\ \big((i-P)*2-2+1:(i-P)*2, (j-P)*2-2+1:(i-P)*2\big) = 1$

Step-16: end if; end for;

Step-17: $N_R = ColumnSize\ of\ I_{binary}$

Step-18: for i=1 to N_R

Step-19: $C_{sum} = \sum I_{binary}\ (:, i)$

Step-20: Check if $C_{sum} = N_R$

Step-21: Position$_1$ $= i$; end if

Step-22: for i=1 to N_R

Step-23: $C_{sum} = \sum I_{binary}\ (:, N_R + 1 - i)$

Step-24: Check if $C_{sum} = N_R$

Step-25: Position$_2$ $= \frac{(N_R{}^2 - i^2)}{N_R + i}$; end if; end step-22 for

Step-26: $N_C = RowSize\ of\ I_{binary}$

Step-27: for i=1 to N_R

Step-28: $R_{sum} = \sum I_{binary}\ (:, i)$

Step-29: Check if $R_{sum} = N_C$

Step-30: Position$_3$ $= i$; end if

Step-31: for i=1 to N_C

Step-32: $R_{sum} = \sum I_{binary}\ (N_C + 1 - i, :)$

Step-33: Check if $R_{sum} = N_C$

Step-34: Position$_4$ $= \frac{(N_C{}^2 - i^2)}{N_C + i}$; end if; end step-27 for

Step-35: $I_{segment} = I_{binary}$ (Position$_3$ to Position$_4$: , Position$_1$ to Position$_2$)

VI. FEATURE EXTRACTION USING GABOR FILTER

After segmentation, we extract the features. This is mentioned in this study as Methodology. In this study first we convert the $I_{segment}$ image to double intensity image. Four floating point numbers are created using following statement

f=[1/3.2,1/3.4,1/3.6,1/3.8]*2*pi

Next gray thresh value of the Grayimage is calculated. Gray thresh is a threshold value between 0 and 1, and which always return a fraction value between 0 and 1. The above this value is treated as 1 and below this value is treated as 0, while converting Grayimage to binary image. In the binary image, I_{binary} the value 1 is considered as background of the image and 0 is foreground or ROI of the fingerprint image. All the pixels, which are having value 1 is extracted using index position which is having value 0 in, I_{binary} image. The starting and ending positions of the pixel which is having value 0 is calculated using imin, jmin, imax, and jmax respectively. i and j represents row and column of the I_{binary} image. These variables are used to extract ROI of the image from the I_{binary} image. To extract minutiae details here 64 × 64 sized Gabor filter is used with 4 different frequencies. The equation for Gabor filter is given by

G (i, j) = exp(-.5*((xPrime/Sx)^2+

Prime/Sy)^2))*cos(2*pi*f(1,fre)*xPrime) (8)

In (8) xPrime = x * cos(theta) + y * sin(theta), yPrime = y * cos(theta) - x * sin(theta), theta=(pi*i)/8, i, can take value from 1 to 4.

Then convolution of the image p1 and imaginary part of G is found. p1 is a 64 × 64 sized double image obtained from I_{binary} . Again the convolution of the image p1 and Real part of G is also found, these two are stored in variables Imgabout and Regabout. Finally mean, standard deviation, and variance mean of Regabout is calculated. The above entire process is repeated for 4 different values of f. Total 12 real values are obtained and which is given as an input for hash function. The hash function is explained in 4.5.

Algorithm for extracting features directly from segmented image

Input: Segmented image, $I_{segment}$

Output: Extracted features

Step-1: Convert segmented image to double

I_{double} = double ($I_{segment}$)

Step-2: Initialize three constants Sx and Sy with value 3 and L with value 4.

Sx=3, Sy=3, L=4

Step-3: Initialize frequency for Gabor filter

f= [1/3.2, 1/3.4, 1/3.6, 1/3.8] * 2 * pi

// [where pi = 22/7]

Step-4: Initialize a matrix p1 for Gabor filter as p1 with initialize value 1 with size 64 × 64

$p1_{\,64\times 64} = 1$

Step-5: Find a grey thresh (threshold value) for I_{double}

level = graythresh (I_{double})

Step-6: Convert double image to binary image using graythresh value

I_{binary} = binary_image (I_{double} , level)

Step-7: [i, j] = find (zero index position of I_{binary} in row and column matrix)

Step-8: Find the minimum and maximum position value for i and j.

imin = min(i), imax = max(i),

jmin = min(j), jmax = max(j)

Step-9: Create a new binary image, $I_{binary\ 1}$ which contains only value zero from I_{binary}

$I_{binary\ 1}$= I_{binary} (imin to imax, jmin to jmax)

Step-10: Initialize a constant variable rate

rate = 64/max (size($I_{binary\ 1}$))

Step-11: Resize the $I_{binary\ 1}$ as

Resize $I_{binary\ 1}$ (i × rate, j × rate) // where i and j are row and column dimension

Step-12: Find the new size of I_{binary}

[i, j] = size ($I_{binary\ 1}$)

Step-13: Round off the value of i, and j

i1 = round ((64-i) / 2)

j1 = round ((64-j) / 2)

Step-14: Reassign the value of p1

p1 = (i1 + 1 to i1 + i, j1 + 1 to j1 + j)

Step-15: Convert p1 to double from binary

p1= double(p1)

Step-16: for fre = 1 to 4

Step-17: for i= 1 to 4

Step-18: Initialize the theta value as theta = (pi × i) / 8;

Step-19: for x= round to nearest integer value (-Sx) to round to nearest integer value (Sx)

Step-20: for y= round to nearest integer value (-Sy) to round to nearest integer value (Sy)

Step-21: Rotate with respect to theta

xPrime = x * cos(theta) + y * sin(theta)

yPrime = y * cos(theta) - x * sin(theta)

Step-22: Use Gabor filter by varying frequency and angle

G(x, y) = exp(-.5*((xPrime/Sx)^2+

(yPrime/Sy)^2))*cos(2*pi*f(1,fre)*xPrime)

// ^ represents power

Step-23: end of for loop of Step-20

Step-24: end of for loop of Step-19

Step-25: Do convolution of P1 and imaginary part of G from central part

Imgabout = conv2 (p1, double (imag (G)), ‘same’)

// ‘same’ does the convolution

// from the central part

Step-26: Do convolution of P1 and real part of G from central part

Regabout = conv2 (p1, double (real (G)), ‘same’)

// ‘same’ does the convolution from the central part

Step-27: Find the mean of Imgabout and Regabout

imfea1(i) = mean (Imgabout)

imfea2(i) = mean (Regabout)

Step-28: Find Standard Deviation of Imgabout and Regabout

stfea1(i) = standard_deviation (Imgabout)

srfea2(i) = standard_deviation (Regabout)

Step-29: Find the mean of variance of Regabout

medfea2(i)=mean(var(Regabout))

Step-30: End of for loops of Step-17 and Step-16

Step-31: features = [imfea2, srfea2, medfea2]

VII. RESULTS AND DISCUSSIONS

Extracting features directly from segmented image based on Gabour filter uses four different values for frequency and theta, which is shown in Table 4.3. These four frequencies and Angle value helps to generate a matrix of size, 7×7 containing total 49 real values due to Gabor filtering process. Each row of the TABLE I results in 3 positive or negative real numbers due to mean, standard deviation, and mean of variance calculations. Before calculating these three statistical calculation convolutions process was conducted on Gabor filter matrix of size 64×64 (p1) and real part of the imaginary number of Gabor value (G).

TABLE I FREQUENCY AND THETA VALUE USED IN GABOR FILTER TO EXTRACT FEATURES

Sr. No	Frequency value	Angle (theta) value
1	1.9635	0.3927
2	1.8480	0.7854
3	1.7453	0.1781
4	1.6535	1.5708

With the aid of Gabor filter process, each fingerprint image produces total of twelve (12) double precision values. These large double precision values ensure that each fingerprint sample produces different hash values through MD5 Hash functions.

TABLE II shows the trainhash1 table values for the benchmark dataset FVC ongoing 2002 DB1_B, dataset image 101_1 to 101_8 using Gabor filtering. Each user will be having 8 fingerprint images.

TABLE II HASH VALUES FOR IMAGE DB1_B 101.TIF

id	**Hash**
1	d579254fa5831c03e60e18729fbc710b
1	27024ce2cdd3dbdfa8d689adb7abc36c
1	23a8d9d5cb9b4eb62dab62bb4a67e30c
1	51398c3f65804111d281ba3e9f62e084
1	3db3a5b0777e10d97b49851c4e71fe49

IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 12, December 2017

1	d7d4a52858c98fb16e37d9613242ca36
1	d2901a0ae4e697d2ca2b9122e7bd2a1e
1	9f164d2dfaa2dff871208789d9056098

Fig.2 Shows output of the tuning based contrast adjustment algorithm. In τ- Tuning Based Filtering Algorithm, dark pixels are highest dark and bright pixels are either highest or near to high bright values. This algorithm is best suited for grayscale fingerprint image, especially 256 × 256 sized. As like histogram equalization one of the pixel intensity values dominates over other. So this algorithm is not best for natural image because it may cause wash out appearance. But this is very good and robust for grayscale fingerprint image. In fingerprint image usually black colour represents ridges and white color represents valley. The dominating intensity value usually falls in upper boundary region of intensity range, which makes the image brighter

The Surfeit clipping based segmentation is analyzed by considering FVC ongoing 2002 DB1_B datasets. A sample fingerprint image named as 102_1.tif from FVC ongoing 2002 dataset is considered in Fig. 3.

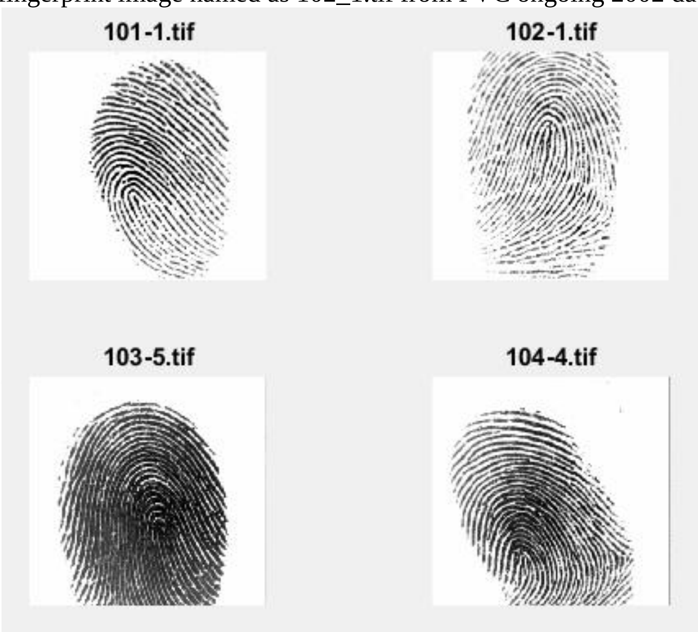


Fig.2. Sample original fingerprint images of FVC ongoing 2002 DB1_B dataset

The speed refers the time taken by the system to enroll as well as authenticate or reject. In technology term this can be referred as Elapsed time or time utilized by the new algorithm or model in to enroll and match. Elapsed time is calculated on following configuration system, and which are given in TABLE III

TABLE III CONFIGURATION OF SYSTEM FOR FINDING ELAPSED TIME

Sr. No.	Parameters	System Configuration
1	Model	Compaq 435
2	Processor	AMD E-350 processor 1.60 GHz
3	Installed Memory	3 GB (2 GB usable)
4	System Type	32-bit operating System
5	Operating System	Windows 7 Starter
6	Software	MATLAB 2015a 32-bit

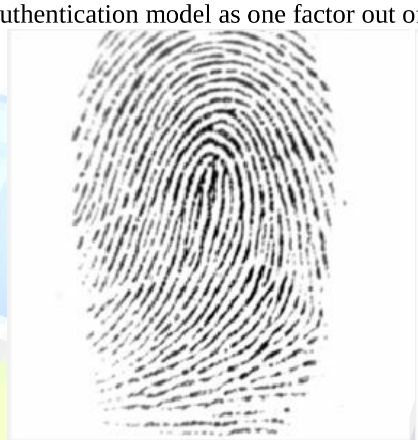
TABLE IV shows execution time of the training phase for Hash generation using Gabor filter. Gabor filter is used to extract features from the segmented image. Execution time of the testing phase is same for all four methods, which are about 0.6 seconds and 0.44 seconds more than training phase.

TABLE IV ELAPSED TIME OF THE TRAINING PHASE

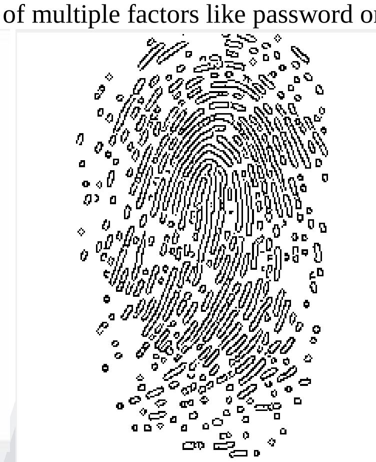
Method Name	**Image name**	**Execution Tine (in seconds) using System-I**	**Average**
Method-2	101_1	9.433374	6.783357
	101_5	5.912853	
	102_2	5.997807	
	103_3	5.789369	

CONCLUSION

In this research study, a new approach for fingerprint Hashcode generation developed based on MD5 Algorithm, which makes use of Tuning based Contrast Adjustment Algorithm and Surfeit based segmentation algorithm (Modified algorithm). The generated Hash code is rotation and translation invariant and can be used as an identity or index key and also can be used along with multifactor Authentication model as one factor out of multiple factors like password or OTP.

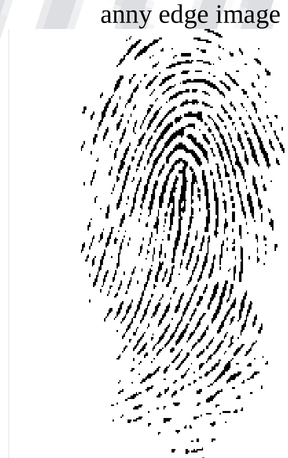


(a) riginal image



(b) anny edge image

(c) ow Resolution Image



(d) egmented image with margins

(e)

egmented image by clipping left right top and bottom border

Fig. 3 Outputs of different phases of Surfeit based Segmentation Algorithm

REFERENCES

[1] Krishna Prasad, K. & Aithal, P.S. (2017). A Novel Method to Contrast Dominating Gray Levels during Image contrast Adjustment using Modified Histogram Equalization. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 1(2), 27-39. DOI: http://dx.doi.org/10.5281/zenodo.896653

[2] Krishna Prasad, K. & Aithal, P.S. (2017). A Critical Study on Fingerprint Image Sensing and Acquisition Technology. *International Journal of Case Studies in Business, IT and Education (IJCSBE)*, 1(2), 86-92. DOI: http://dx.doi.org/

[3] Krishna Prasad, K. & Aithal, P.S. (2017). A Conceptual Study on Image Enhancement Techniques for Fingerprint Images. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 1(1), 63-72. DOI: http://dx.doi.org/10.5281/zenodo.831678

[4] Krishna Prasad, K. & Aithal, P.S. (2017). Literature Review on Fingerprint Level 1 and Level 2 Features Enhancement to Improve Quality of Image. *International Journal of Management, Technology, and Social Sciences (IJMTS)*, 2(2), 8-19. DOI: http://dx.doi.org/10.5281/zenodo.835608

[5] Krishna Prasad, K. & Aithal, P.S. (2017). Fingerprint Image Segmentation: A Review of State of the Art Techniques. *International Journal of Management, Technology, and Social Sciences (IJMTS)*, 2(2), 28-39. DOI: http://dx.doi.org/10.5281/zenodo.848191

[6] Krishna Prasad, K. & Aithal, P.S. (2017).Two Dimensional Clipping Based Segmentation Algorithm for Grayscale Fingerprint Images. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 1(2), 51-65. DOI: http://dx.doi.org/10.5281/zenodo.1037627.

[7] Krishna Prasad, K. & Aithal, P.S. (2017). A conceptual Study on Fingerprint Thinning Process based on Edge Prediction. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 1(2), 98-111. DOI: http://dx.doi.org/10.5281/zenodo.1067110

[8] Krishna Prasad, K. & Aithal, P.S. (2017). A Study on Fingerprint Hash Code Generation Using Euclidean Distance for Identifying a User. International Journal of Management, Technology, and Social Sciences (IJMTS), 2(2), 116-126. DOI : http://doi.org/10.5281/zenodo.1133545

[9] K. Krishna Prasad & P. S. Aithal (2018). A Study on Multifactor Authentication Model Using Fingerprint Hash Code, Password and OTP, International Journal of Advanced Trends in Engineering and Technology, 3(1), 1-11.

[10] K.Krishna Prasad & P.S. Aithal (2018). A Novel Tuning Based Contrast Adjustment Algorithm for Grayscale Fingerprint Image.
In press.

[11] Zhang, J., Lai, R., & Kuo, C. C. J. (2012). Latent fingerprint detection and segmentation with a directional total variation model. In Proceedings - International Conference on Image Processing, ICIP (pp. 1145–1148).

[12] Zhang, J., Lai, R., & Kuo, C. C. J. (2012). Latent fingerprint segmentation with adaptive total variation model. In Proceedings - 2012 5th IAPR International Conference on Biometrics, ICB 2012 (pp. 189–195).

[13] Choi, H., Boaventura, M., Boaventura, I. A. G., & Jain, A. K. (2012). Automatic segmentation of latent fingerprints. 2012 IEEE Fifth International Conference on Biometrics: Theory, Applications, and Systems (BTAS), 303–310.
[14] Xue, J., & Li, H. (2012, July). Fingerprint image segmentation based on a combined method. In Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012 IEEE International Conference on (pp. 207-208). IEEE.
[15] Short, N. J., Hsiao, M. S., Abbott, A. L., & Fox, E. A. (2011). Latent fingerprint segmentation using ridge template correlation. 4th International Conference on Imaging for Crime Detection and Prevention 2011 (ICDP 2011), P28–P28.

[16] Tulyakov, S., Farooq, F., Mansukhani, P., & Govindaraju, V. (2007). Symmetric hash functions for secure fingerprint biometric systems. *Pattern Recognition Letters*, *28*(16), 2427-2436.
[17] Wahab, A., Chin, S., & Tan, E. (1998). Novel approach to automated fingerprint recognition. *IEE Proceedings - Vision, Image and Signal Processing*. DOI: https://doi.org/10.1049/ip-vis:19981809.
[18] Pizer, S. M. (2003). The Medical Image Display and Analysis Group at the University of North Carolina: Reminiscences and philosophy. *Medical Imaging, IEEE Transactions on*, 22(1), 2–10. DOI: https://doi.org/10.1109/TMI.2003.809707.
[19] Pei, S. C., Zeng, Y. C., & Chang, C. H. (2004). Virtual restoration of ancient Chinese paintings using color contrast enhancement and Lacuna texture synthesis. *IEEE Transactions on Image Processing*, 13(3), 416–429. DOI: https://doi.org/10.1109/TIP.2003.821347
[20] De La Torre, Á., Peinado, A. M., Segura, J. C., Pérez-Córdoba, J. L., Benítez, M. C., & Rubio, A. J. (2005). Histogram equalization of speech representation for robust speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(3), 355–366. DOI: https://doi.org/10.1109/TSA.2005.845805.

[21] Gonzalez, R. C., Woods, R. W. (2002). *Digital Image Processing*. *Education*. DOI: https://doi.org/10.1049/ep.1978.0474.
[22] Mehtre, B. M., & Chatterjee, B. (1989). Segmentation of fingerprint images-a composite method. Pattern Recognition, 22(4), 381-385.

[23] Ratha, N. K., Chen, S., & Jain, A. K. (1995). Adaptive flow orientation-based feature extraction in fingerprint images. *Pattern Recognition*, *28*(11), 1657-1672.
[24] Jain, A. K., Ratha, N. K., & Lakshmanan, S. (1997). Object detection using Gabor filters. *Pattern recognition*, *30*(2), 295-309.
[25] Sun, X. and Ai, Z. (1996) Automatic feature extraction and recognition of fingerprint images, Proceeding of ICSP'96, Beijing, Pp.1086-1089.
[26] Moayer, B., & Fu, K. S. (1975). A syntactic approach to fingerprint pattern recognition. Pattern Recognition, 7(1–2), 1–23. https://doi.org/10.1016/0031-3203(75)90011-4.
[27] Naji, A.W., Ramli, A.R., Ali, R., Rahman, S.A., and Ali, M.L. (2002). A segmentation algorithm based on histogram equalizer for fingerprint classification system, Second International Conference on Electrical and Computer Engineering ICECE 2002, Dhaka, Bangladesh, Pp. 390-393.

[28] Alonso-Fernandez, F., Fierrez-Aguilar, J., & Ortega-Garcia, J. (2005, September). An enhanced gabor filter-based segmentation algorithm for fingerprint recognition systems. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on* (pp. 239-244). IEEE.