SIGNATURE VERIFICATION AND REDUCTION OF FAR AND FRR USING HUNGARIAN ALGORITHM

[1]A.P.ATHANE and [2]PRASHANT D. GANGAWANE
[1&2] Assistant Professor , Department of Textiles , D.K.T.E.Society's Textile & Engineering Institute, Ichalkaranji , Dist.: Kolhapur, Maharashtra, India. Pin-416 115.

Abstract :
The main goal of signature verification system is to detect forgeries and simultaneously reduce rejection of genuine signature. This paper deals with off-line signature verification and presents simple and effective verification method depends on raw binary pixel intensities and avoids complex set of features. . This method looks at the off-line signature verification problem as graph matching problem, using Hungarian algorithm. Genuine and forgery signatures produced by different subject are used to test this method. This method is useful to detect random and simple forgery signatures except skilled forgery. Threshold selection is based on statistical parameters like cost factor (Cost) which is derived from standard cost matrix and co-relation factor (K). According to that FAR(false acceptance rate), FRR(false rejection rate) is calculated. Vital parameters such as FAR and FRR is considerably reduced. This method provides high accuracy with optimum speed. Final results are displayed using after signature verification.

***Index Terms*- off-line signature verification, cost matrix, cost factor, co-relation factor, FAR, FRR, GUI, Hungarian algorithm.**

I. INTRODUCTION

Signature verification is an important research area in the field of personal authentication. Signature is just a special case of handwriting and often is just a symbol. Signatures are most legal and common means for individual’s identity verification. The importance of signature verification arises from the fact that it has long been accepted in government, legal and commercial transaction.

Signature verification is the decision about whether the signature is genuine or forgery. In this decision phase the forgery images can be classified in three groups i) random ii) simple iii) skilled. Random forgeries are formed without any knowledge of signers name and signature’s shape. Simple forgeries are produced knowing the name of the signer but without having an example of signer’s signature. Skilled forgeries are produced by people attempting to imitate as closely as possible.

We can generally distinguish between two different categories of verification systems. On-line, for which the signature signal is captured once the writing process is over and thus the dynamic information is available. And off-line, for which the signature is captured once the writing process is over and only a static image is available. The precision of signature verification can be expressed by two types of errors. The percentage of genuine signature rejected as forgery which is called FRR. The percentage of forgery signature accepted as genuine is called FAR. The objective of signature verification is to discriminate between two classes; the original and the forgery, which are related to intra and inter personal variability. The variation among signatures of the same person is called intra personal variation. The variation between originals and forgeries is called Inter Personal Variation.

In this paper we concentrated on problem of off-line signature verification. Off-line signatures are scanned from paper documents, where they were written in conventional way. Off-line signature analysis can be carried out with a scanned image of signature using a standard camera or scanner. The signature verification using off-line technique is relatively more difficult, since the volume of information available is less. The static information derived in off-line signature verification system may be global, structural, geometric or stastical. Off-line signature verification compares two signatures after they have been put, using scanned images of two signatures as input. Off-line method, needs to apply complex image processing techniques to segment and analyze signature shape for the feature extraction.

The objective of the work is to discriminate between two classes; the original and forgery. This method is useful in separating random as well as simple forgery from original signature by comparison with original signature. In this method features are based on geometric properties (graph matching problem). Hungarian algorithm is used to solve the standard cost matrix to calculate the value of cost factor (Cost). The algorithm used have given improved results compared to previously proposed algorithm based on geometric center with higher efficiency. Using co-relation factor (K) for comparison, vital parameters FAR and FRR are considerably reduced.

This paper is organized in the following sections. Section II describes signature verification using Hungarian algorithm. Results are presented in section III. Section IV provides conclusion and references.

II. SIGNATURE VERIFICATION USING HUNGARIAN ALGORITHM.

Collection of signatures and Preprocessing

Signatures were collected on A4 size sheets. These sheets were later scanned and signatures were stored in labeled folders, (or database) being numbered. The next stage was the preprocessing of these signatures collected. The purpose of preprocessing phase is to make signature ready for feature extraction. The code will normalize the signature image. The code does not bring every signature to a similar state, but it is accurate. The preprocessing stage includes four steps: Resizing, thinning, rotating, cropping.

Resizing: Load the image file and store it as variable. Input Image is either RGB, grayscale or binary image. Resize the image to pixel size 32 x 64.

Thinning: After resizing the image in to required pixel size. RGB or color map image is converted to grayscale. Conversion from RGB image to black and white image. For signature verification, color of the ink has no significance at all. Instead the form of the two signatures must compared. Hence all the scanned images were converted to black and white images, where white is represented by 1 and black by 0. Hence the signature part of the image was represented by 0 and blank part (without any signature) by 1. This conversion also makes future conversion easier. Then convert the image to double precision for more accurate results. Finally, image is converted to binary image in the form of black and white pixels. Morphological operations such as thinning were performed on binary image. It removes pixels so that an image without holes shrinks to minimally connected stroke and object with holes shrinks to a connected ring half way between each hole and the outer boundary. This operation is useful to remove the pepper noise in the image, also to remove stray dots arising in the image due to improper scanning

Rotating: Extracting the black pixels and inverting the image from black on white background, to white on black background. Now the signature part coded by 1, while blank spaces are coded by 0. This makes logical comparisons a lot easier. Later moving the signature to the origin and find out new curve of the signature. Minimum Eigen value as well as Eigen vector were calculated. Then the signature was rotated and passed the new co-ordinates. After that the signature was moved to its original position. Because of this we can reduce intra personal variations.

Eigen values and Eigen vector

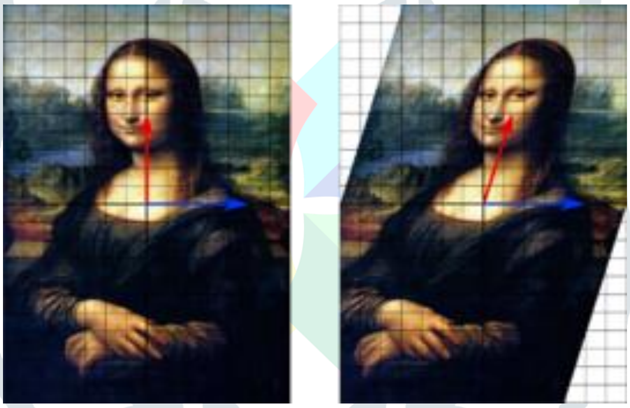


Fig. 2.1 Eigen values and Eigen vector

In this shear mapping the red arrow (vertical direction) changes direction but the blue arrow (horizontal direction) does not. Therefore the blue arrow is an eigenvector, with eigenvalue 1 as its length is unchanged.

The eigenvectors of a square matrix are the non-zero vectors that, after being multiplied by the matrix, remain parallel to the original vector. For each eigenvector, the corresponding eigenvalue is the factor by which the eigenvector is scaled when multiplied by the matrix. The prefix Eigen- is adopted from the German word "Eigen" for "self" in the sense of a characteristic description. The eigenvectors are sometimes also called characteristic vectors. Similarly, the eigenvalues are also known as characteristic values. The mathematical expression of this idea is as follows: if A is a square matrix, a non-zero vector v is an eigenvector of A if there is a scalar λ (lambda) such that

$$Av = \lambda v$$

The scalar λ (lambda) is said to be the eigenvalue of A corresponding to v. An Eigen space of A is the set of all eigenvectors with the same eigenvalue together with the zero vector. However, the zero vector is not an eigenvector. These ideas are often extended to more general situations, where scalars are elements of any field, vectors are elements of any vector space, and linear transformations may or may not be represented by matrix multiplication. For example, instead of real numbers, scalars may be complex numbers; instead of arrows, vectors may be functions or frequencies; instead of matrix multiplication, linear transformations may be operators such as the derivative from calculus. These are only a few of countless examples where eigenvectors and eigenvalues are important in such cases, the concept of direction loses its ordinary meaning, and is given an abstract definition. Even so, if that abstract direction is unchanged by a given linear transformation, the prefix "Eigen" is used, as in eigenfunction, eigenmode, eigenface, eigenstate, and eigenfrequency. Eigenvalues and eigenvectors have many applications in both pure and applied mathematics. They are used in matrix factorization, in quantum mechanics, and in many other areas.

Definition:

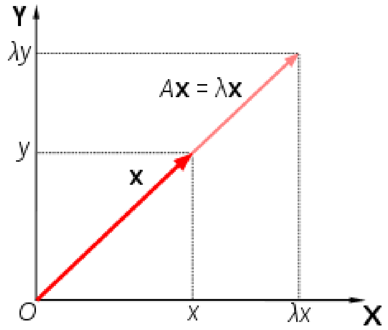


Fig. 2.2 Eigen vector and value

Matrix A acts by stretching the vector x, not changing its direction, so x is an eigenvector of A. Eigenvectors and eigenvalues depend on the concepts of vectors and linear transformations.

Cropping: After rotating the image the signature might go out of boundary (128x128) therefore we have to move the signature curve. We get cropped image after drawing the boundary box and getting the margins, thereby eliminating blank areas from the sides of the image. Thus each image consists only of the signature part.

Cost matrix reduction.

When new signatures comes for testing we have to calculate features of the signature. In this paper features are based on geometric properties (graph matching problem). Our method looks at the off-line signature verification as graph matching problem, for which we used the Hungarian method to solve. This is simple and effective method, depends on raw binary pixel intensities and avoids complex set of features. After signature collection they were scanned as binary images using scanner, resolution set to 300 dpi. Normalized signature S is set of pixels constitutes final image; denoted as set of vertices X which represents signature S. Normalized signature size at 32 x 64 pixels. Verification time is in 2 sec range.

Let S1 and S2, two off-line signature images [1] to be compared. X and Y represents sets of vertices (pixels) that represents S1 and S2. Cost matrix C which is m * n matrix, whose rows corresponds to vertices X and column corresponds to vertices Y. Every vertex x Є X and y Є Y has its X and Y co-ordinates (row and column number) in raster image. Co-ordinates are used to find distance between x and y after aligning the centers of area of sets X and Y. cxy of C, x Є X and y Є Y equals the b Euclidean distance between x and y . This is the cost of matching point x from signature S1 to point y from S2.. After calculating all entries of c, the formulated assignment problem is solved. Cost min. = sum of all entries of cxy corresponds to min. cost solution.

To verify that a test signature, S, belongs to a specific subject, it is compared against a predetermined number, p, of prototype genuine signatures of the same subject as follows:

1. The p prototype signatures are preprocessed to produce the sets of vertices Y1, Y2, . . . , Yp .
2. S is preprocessed as described in to produce the set of vertices X.
3. Let d = ∞, where d will measure the minimum distance between S and the prototype signatures of the considered subject.

For every set of vertices Yi, i = 1, 2, . . . , p, do

{Let f = min (|X|, |Yi|)/ max (|X|, |Yi|).

If f ≥ α, then

{Find the cost matrix C between X and Yi as described earlier.

Rotate the matrix C so that there are at least as many rows as columns. Let r = number of rows of C. Compute the minimum cost matching, cost min , of rows into columns considering the cost matrix C.

Let cost min = cost min /(f × r).

If cost min < d, then let d = cost min.

}

}

The factor f, measures the percentage of vertices sharing in any complete matching X into Y. Use of Hungarian matching algorithm for cost matrix which finds complete matching of row into columns. Algorithm were run many times using the following size of normalized boxes 32 x 64. All these results are stored in a data matrix which comprises the feature model.

Comparison

First the input test signature is appropriately processed as previously described, to render it suitable for comparison by software. The user, against whom the test signature will be checked for a match was chosen from suitable database. The final result, called the test result, gives an accurate representation of the difference between test and specimen signatures. In this method segment cost of both the images were calculated by taking length of vector as Y or X. Then unique length of the vector was calculated to build the cost matrix. Hungarian algorithm was used to solve the cost matrix. Where row elements represents one image and column elements represents second image. If a constant is added (or subtracted) to every element of any row (or column) of the cost matrix

[cij] in an assignment problem then an assignment which minimizes the total cost for the new matrix will also minimize the total cost matrix. When cost matrix is reduced there is optimum assignment of zeros to rows and columns. Ideal cost matrix that has the same optimal assignment as given matrix.

In this method for comparison between two random signatures a threshold was set, when the co-relation factor (K) >= 0.05 and cost factor (Cost) > 0.85 output result was displayed as match or it was displayed as Not-match.

Assignment Problem: Hungarian Algorithm and Linear Programming

a) Introduction to Assignment Problem: Let C be an n x n matrix representing the costs of each of n workers to perform any of n jobs, the assignment problem is to assign jobs to workers so as to minimize the total cost. Since each worker can perform only one job and each job can assigned to only one worker the assignment constitute an independent set of the matrix C.

Example 2.1

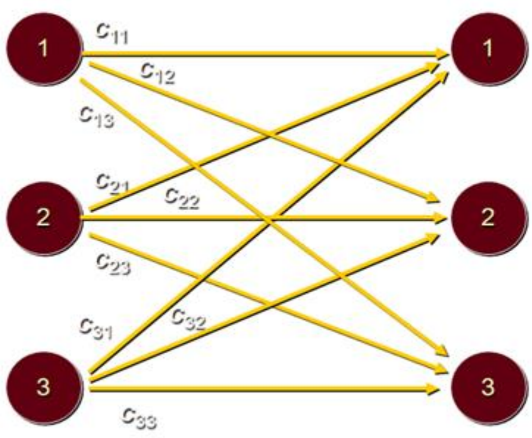
$$C(i,j) = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} p & q & r & s \end{array} \\ \begin{bmatrix} 1 & \mathbf{2} & 3 & 4 \\ 2 & 4 & 6 & \mathbf{8} \\ 3 & 6 & \mathbf{9} & 12 \\ \mathbf{4} & 8 & 12 & 16 \end{bmatrix} \end{array} \quad \text{----- (3.3)}$$

Workers = {a, b, c, d}, Jobs = {p, q, r, s}

An arbitrary assignment A = {(a, q), (b, s), (c, r), (d, p)}

Total cost = 23

An assignment problem seeks to minimise the total cost assignment of m workers to m jobs, given the cost of worker I performing job j is Cij.It assumes all workers are assigned and each job is performed. An assignment problem is a special case of a transportation problem in which all supplies and all demands are equal to 1.



workers job

Fig. 2.3 The network representation of an assignment problem

The network representation of an assignment problem with three workers and three jobs is shown in the figure. An assignment problem seeks to minimize the Linear programming (LP) problems can be solved on the computer using MATLAB and many other. There special classes of LP problems such as the assignment problems (AP). Efficient solutions methods exist to solve AP. AP can be formulated as LP and solved by general purpose LP codes. However there are many computer packages, which contain separate computer codes for these models which take advantage of problem network structure.

b) The Hungarian method: is a combinatorial optimization algorithm which solves the assignment problem in polynomial time. It was developed and published by Harold Kuhn in 1955, who gave the name 'Hungarian method" because the algorithm was largely based on the earlier works of two Hungarian mathematicians: Denes Konig and Jeno Egervary. Jemes Munkres reviewed the algorithm in 1957 and observed that it is (strongly) polynomial. In 2006 it was discovered that Carl Gustav had solved the assignment problem in the 19th century and published posthumously in 1890 in Latin. The Hungarian method solves minimization assignment problems m workers and m jobs. Special considerations can include: Number of workers does not equal number of jobs add dummy workers/jobs with 0 assignment costs as needed workers i cannot do job j – assign Cij = +M.

Maximization objective create an opportunity loss matrix subtracting all profits for each job from the maximum profit for that job before beginning the Hungarian method.

Step 1: For each row, subtract the minimum number in that row from all numbers in that row.

Step 2: For each column, subtract the minimum number in that column from all numbers in that columns. Step 3: Draw the minimum number of lines to cover all zeroes. If this number = m, STOP – an assignment can be made. for each row, subtract the minimum number in that row from all numbers in that row.

Step 4: Determine the minimum uncovered number (call it d).Subtract d from uncovered numbers. Add d to numbers covered by two lines. Numbers covered by one line is same then, GO TO STEP 3.

Finding the minimum numbers of lines and determining the optimal solutions.

Step 1: Find a row or column with only one unlined zero and circle it (If all rows/columns have two or more unlined zeros choose an arbitrary zero).

Step 2: If the circle is in a row with one zero, row draw a line through it column. If the circle is in a column with one zero, draw a line through its row. One approach, when all rows and columns have two or more zeros, is to draw a line through one with most zeros, breaking ties arbitrarily.

Step 3: Repeat step 2 until all circles are lined. If this minimum number of lines equals m, the circles provide the optimal assignment.

Creating Threshold:

In order to compare whether a given random input signature is match with a particular user, thresholds were defined for each parameter of testing viz. FAR, FRR. Signature one was taken from data base and compared with rest of the signatures. The co-efficient of co-relational between these two matrices was computed and stored in data vector. Also cost factor was calculated for verification of two signatures and stored.

a. False acceptance: suppose that a given signature has been signed by a particular person that is genuine. However, if the system rejects this signature as not belongs to that particular person, such cases of rejection are termed as false rejection. There were total 13 signatures of the same subject chosen from database. First signature was compared with rest of the signatures using the suitable criteria co-relation factor K > 0 cost factor

(Cost) >= 0.8 for testing of false acceptance. Finally false rejection ratio was calculated by using the formula

$$FRR = \left\{ \frac{(No.of\ originals\ rejected)}{(No.of\ originals\ tested)} \right\} * 100$$

b. False acceptance: suppose that a given signature does not belong to person A however , on comparing with signatures of person A, if the system accepts signature , then such case of acceptance are termed as false acceptance . Testing for false acceptance was done by randomly choosing 25 other signatures from database. After comparison of first signature with the rest of the signatures using the criteria co-relation factor (K) > 0.01 and cost factor (cost) >=0.8. Any acceptance made by system were termed as false acceptance signature for testing of false acceptance. Finally false acceptance ratio was calculated by using the formula

$$FAR = \left\{ \frac{(No.of\ forgery\ accepted)}{(No.of\ forgery\ tested)} \right\} * 100$$

Simulation and software development using MATLAB

In this project MATLAB language is used for algorithm development used in image processing. It is also used for simulation of FAR (false acceptance ratio) and FRR (false rejection ratio) using normalized box of size 32 x 64. It is also used for different software development and performing simulation in this project which provides better and accurate results compared to other software languages. The following information describes the importance of MATLAB language.

III. Results

This is the sophisticated method of off-line signature verification with visual display using GUI (graphical user interface) For comparison of two signatures. 21 random signature images were stored in database as S1 to S21. We can use pushbuttons-select image1, select image2 to select signatures for comparison and pushbutton-Match to display result using text box. As shown in figure 1. If there is matching between two selected signatures, the result is displayed as Match. As shown in figure 2 if there is not matching between two images, the result is displayed as Not-match.

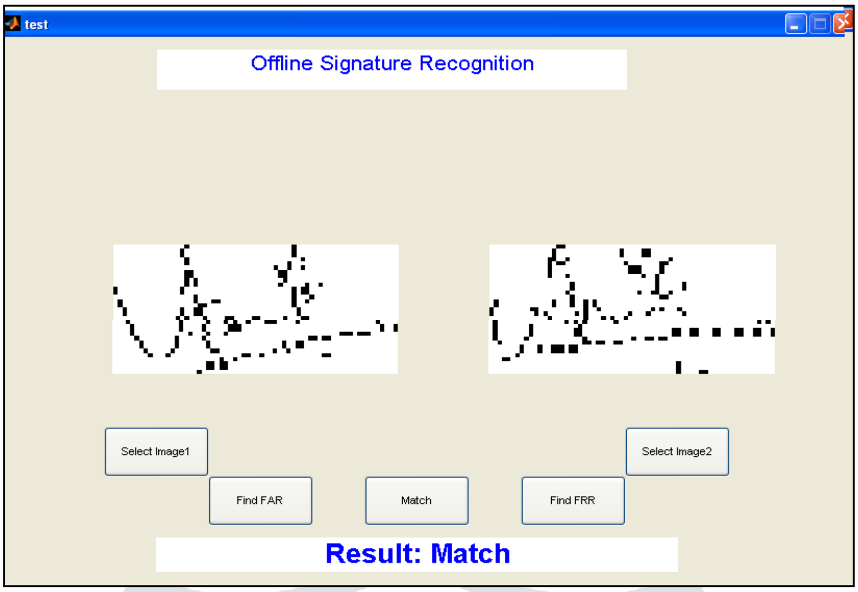


Fig. 3.1 Displays matching of two selected signatures from database

as shown in fig. 3.2 if there is not matching between two images , the result is displayed as Not-match. Results are as shown in Table 3.1

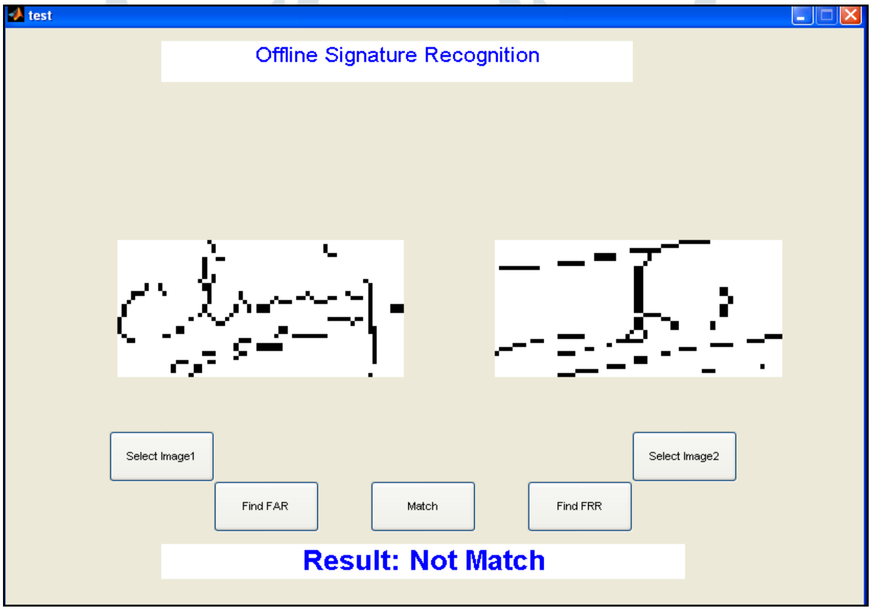


Fig. 3.2 Displays not matching of two selected signatures from database

Table 3.1 Result of signature comparison

Thresholds used for comparison if K >= 0.1 and Cost >= 0.8 result is Match otherwise Not match				
Compared signatures	**K (correlation factor)**	**T (total cost)**	**Cost (cost factor)**	**Result**
1 and 11	0.0152	-	-	Not match
2 and 17	-0.0124	-	-	Not match
6 and 6	1	200	0.8983	Match
7 and 17	0.0024	-	-	Not match

1 and 17	0.1068	164	0.9193	Match

Result of FAR(False Acceptance Ratio)To calculate value of %AvgFAR signature folders FAR1, FAR2 and FAR3 are used. 13 signatures are stored in each folder. First genuine signature is compared with remaining 12 random forgery signatures of same subject as well as different subjects.

Table 3.2 Result of FAR1

Thresholds used for comparison if K >= 0.1 and Cost >= 0.8 result is Falsely accepted otherwise Not accepted				
Signature 1 Compared with	**K (correlation factor)**	**T (total cost)**	**Cost (cost factor)**	**resultFAR1**
2	0.1452	288	0.8455	Falsely accepted
3	0.015	-	-	Not accepted
4	0.0377	-	-	Not accepted
5	0.1154	321	0.8616	Falsely accepted
6	0.0055	-	-	Not accepted
7	0.0018	-	-	Not accepted
8	0.0732	-	-	Not accepted
9	0.0418	-	-	Not accepted
10	0.0095	-	-	Not accepted
11	0.0282	-	-	Not accepted
12	0.0359	-	-	Not accepted
13	0.0049	-	-	Not accepted

Forgery signatures 2 and 5 in folder FAR1 are falsely accepted: **resultFAR1 = 2**

Table 3.3 Result of FAR2

Thresholds used for comparison if K >= 0.1 and Cost >= 0.8 result is Falsely accepted otherwise Not accepted				
Signature 1 Compared with	**K (correlation factor)**	**T (total cost)**	**Cost (cost factor)**	**resultFAR2**
2	0.1575	28	260	Falsely accepted
3	0.0991	-	-	Not accepted
4	-0.0174	-	-	Not accepted
5	0.1478	32	184	Falsely accepted
6	0.0280	-	-	Not accepted
7	0.0221	-	-	Not accepted
8	0.0684	-	-	Not accepted
9	0.0972	-	-	Not accepted
10	- 0.0023	-	-	Not accepted
11	0.0531	-	-	Not accepted
12	0.0227	-	-	Not accepted
13	0.0238	-	-	Not accepted

Forgery signatures 2 and 5 in folder FAR2 are falsely accepted: **resultFAR2 = 2**

Table 3.4 Result of FAR3

Thresholds used for comparison if K >= 0.1 and Cost >= 0.8 result is Falsely accepted otherwise Not accepted				
Signature 1 Compared with	**K (correlation factor)**	**T (total cost)**	**Cost (cost factor)**	**resultFAR3**
2	0.1152	233	0.8782	Falsely accepted
3	0.0497	-	-	Not accepted
4	0.0637	-	-	Not accepted
5	0.2541	262	0.8925	Falsely accepted
6	0.0168	-	-	Not accepted
7	-0.0073	-	-	Not accepted
8	-0.0084	-	-	Not accepted
9	0.716	-	-	Not accepted
10	0.0972	-	-	Not accepted
11	0.0545	-	-	Not accepted
12	-0.033	-	-	Not accepted
13	0.0467	-	-	Not accepted

Forgery signatures 2 and 5 in folder FAR3 are falsely accepted: **resultFAR3 = 2**

Final result of FAR for three folders: result AR =

2 2 2

Also %FAR for each folder is displayed as below.
FAR1 = 16.6667 FAR2 = 16.6667 FAR3 = 16.6667
And as shown in fig. 5.3 %AvgFAR value is displayed on GUI screen using text box. Pushbutton Find FAR is used to display the result of FAR.

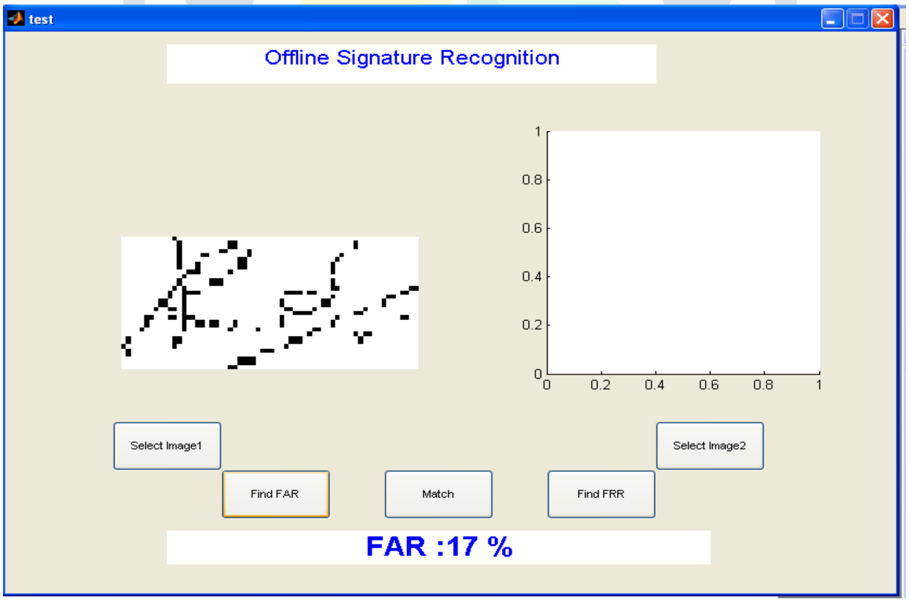


Fig. 3.3 Displays value of %AvgFAR.

Result of FRR(False Rejection Ratio):

To calculate value of %AvgFRR (genuine) signature folders FRR1, FRR2 and FRR3 are used. 13 signatures are stored in each folder. First genuine signature is compared with remaining 12 genuine signatures of the same subject and resultFRR for falsely rejected signatures is displayed for each folder.

Table 3.5 Result of FRR1

Thresholds used for comparison if K >= 0.1 and Cost >= 0.8 result is Falsely accepted otherwise Not accepted				
Signature 1 Compared with	**K (correlation factor)**	**T (total cost)**	**Cost (cost factor)**	**resultFRR1**
2	0.1842	323	0.8707	Not rejected
3	0.0713	-	-	Falsely rejected
4	0.1394	359	0.8893	Not rejected
5	0.0546	-	-	Falsely rejected
6	0.1605	298	0.8583	Not rejected
7	0.0813	-	-	Falsely rejected
8	0.1565	345	0.882	Not rejected
9	0.1345	287	0.8529	Not rejected
10	0.1446	287	0.8529	Not rejected
11	0.2648	328	0.8733	Not rejected
12	0.1628	384	0.9026	Not rejected
13	0.1321	304	0.8612	Not rejected

Genuine signatures 3, 5 and 7 in folder FRR1 are falsely rejected**: resultFRR1 = 3**

Table 3.6 Result of FRR2

Thresholds used for comparison if K >= 0.1 and Cost >= 0.8 result is Falsely accepted otherwise Not accepted				
Signature 1 Compared with	**K (correlation factor)**	**T (total cost)**	**Cost (cost factor)**	**resultFRR2**
2	0.162	184	0.9286	Not rejected
3	0.1278	195	0.9343	Not rejected
4	0.0731	-	-	Falsely rejected
5	0.1241	176	0.9247	Not rejected
6	0.1144	180	0.9267	Not rejected
7	0.1953	176	0.9247	Not rejected
8	0.1823	176	0.9247	Not rejected
9	0.2175	198	0.9357	Not rejected
10	0.1241	176	0.9247	Not rejected
11	0.0795	-	-	Falsely rejected
12	0.2263	161	0.9173	Not rejected
13	0.1400	180	0.9267	Not rejected

Genuine signatures 4 and 11 in folder FRR2 are falsely rejected**: resultFRR2 = 2**

Table 3.7 Result of FRR3

Thresholds used for comparison if K >= 0.1 and Cost >= 0.8 result is Falsely accepted otherwise Not accepted				
Signature 1 Compared with	**K (correlation factor)**	**T (total cost)**	**Cost (cost factor)**	**resultFRR3**
2	0.1677	232	0.9009	Not rejected
3	0.1337	211	8906	Not rejected
4	0.1647	248	0.9089	Not rejected
5	0.1732	262	0.916	Not rejected
6	0.1103	284	0.9274	Not rejected
7	0.1014	247	0.9084	Not rejected
8	0.0885	-	-	Falsely rejected
9	0.2305	246	0.9079	Not rejected
10	0.1955	211	0.8906	Not rejected
11	0.0744	-	-	Falsely rejected
12	0.1418	220	0.895	Not rejected
13	0.1734	211	0.8906	Not rejected

Genuine signatures 8 and 11 in folder FRR3 are falsely rejected**: resultFRR3 = 2**

Final result of FRR for three folders: resultFRR =
3 2 2

Also %FRR for each folder is displayed as below.
FRR1 = 25 FRR2 = 16.6667 FRR3 = 16.6667
And as shown in fig. 5.4 %AvgFRR value is display GUI screen using text box. Pushbutton Find FRR is used to display the result
of FRR.

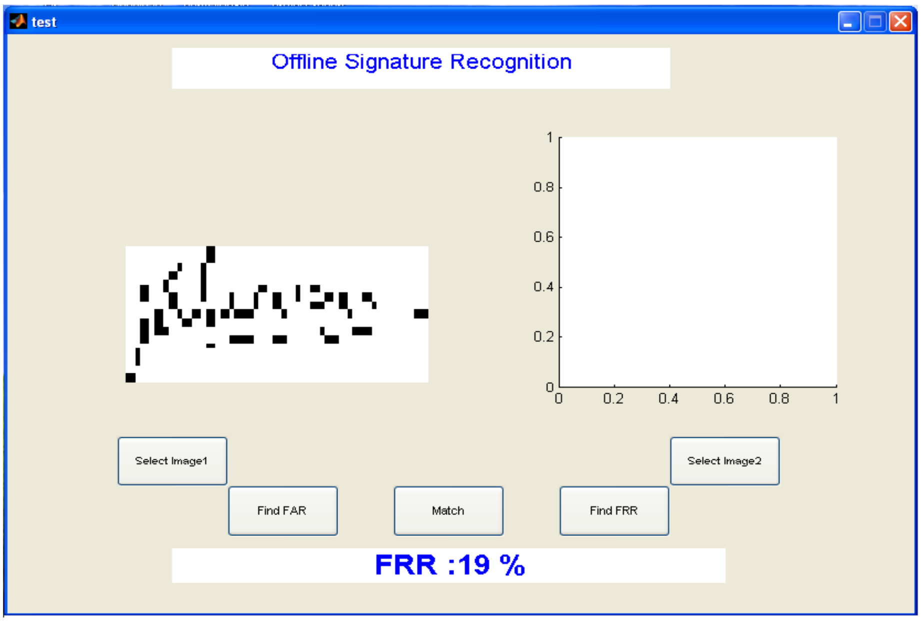


Fig. 3.4 Displays value of %AvgFRR.

IV. CONCLUSION

In this paper we presented a new off-line signature verification technique which is based on graph matching problem solved using Hungarian algorithm. This paper deals with off-line signature verification and presents simple and effective verification method depends on raw binary pixel intensities and avoids complex set of features. This method performs much better results for detection of random and simple forgery, than any other off-line signature verification methods. In general there are different thresholds for different types of forgery detection. But here threshold is same for random, simple and skilled forgeries. Because this method mainly eliminating random and simple forgeries. The result shows reduction in vital parameters such as FAR and FRR considerably in signature verification processes. Carefully chosen discriminating features of signature combined with use of cost matrix. Cost factor (Cost) and co-relation factor (K) used as a threshold for comparison made our system more powerful, compared to other existing systems. Both in terms of success ratio and ease of implementation with optimize run time. GUI (graphical user interface) is used to display results with high accuracy. The positive property of our algorithm is that it provides high accuracy with optimum speed. Drawback of this method is, it cannot classify skilled forgeries from genuine signature correctly. Future direction is to classifying skilled forgeries correctly.

REFERENCES

[1] Ibrahim S. ABUHAIBA, "Offline Signature Verification Using Graph Matching"Department of Electrical and Computer Engg, Islamic University of Gaza, P. O. Box 1276, Gaza-PALESTINE

[2] A. C. Ramchandra, J. S. Rao, K. B. Raja, K. R. Venugopal, L. M. Patnaik. "Robust Offline Signature Verification Based On Global Features". In Proceedings of the IEEE International Advance Computing Conference. Patiala, India, 2009

[3] Cemil OZ, Zafer Demir, Fakret Ercal, "Signature recognition & verification with ANN". Sakarya University Computer Engg. Dept, Sakarya Turky

[4] Almudena Gilperez, Fernando Alonso-Fernandez, Susana Pecharroman, Julian Fierrez, Biometric Recognition Group-ATVS, "Off-line signature verification system based on contour features". Escuela Politecnica Superior-Universidal Autonoma de Madrid Adva. Francisco Tomas y Valiente, 11-28049 Madrid, Spain.

[5] Debasish Jena, Banshidhar Majhi, Sanjaykumar Jena,"Improved Off-line signature verification scheme using feature point extraction method". Centre for IT Education Bhubaneswar, 751010, Orissa,India, NIT, Rourikela,769008, Orissa, India.

[6] Emre Ozgunduz, Tulin Senturk & M. Elif Karsligil, "Off-line signature verification & recognition by support vector machine using the global, directional & grid features of system". Computer Engg, Dept, Yildiz, Istunbul, Turky.

[7] Samaneh Ghandali, Mohsen Ebrahimi Moghaddam, "Off-line signature identification and verification based on image registration & fusion & discrete wavelet transform". Electrical & Computer Engg. Dept, Shahid Beheshti University G. C. Tehran, Iran.

[8] K. Huang, H. Yan, "Off-line signature verification based on geometric feature extraction and neural network classification," Pattern Recognition, 30(1), pp. 9–17, 1997.

[9] Abhay Bansal, Bharat Gupta, Gaura Khandelwal, and Shampa Chakraverty, "Offline Signature Verification Using Critical Region Matching", Dept of Computer Science Netaji Subhash Institute of Tech, Delhi University.

[10] Digital Image Processing Using MATLAB – Rafael C. Gonzalez, Richard E. Woods.

[11] Digital Signal & Image Processing Using MATLAB – Gerald Blanchet , Maurice Charbit- ISTE Edition