Multi-Layer Framed Offline Signature Recognition Algorithm

W. U. Wickramaarachchi and S. Vasanthapriyan

Department of Computing & Information Systems, Sabaragamuwa University of Sri Lanaka, Belihuloya, Sri Lanka

Email: {wirajudara, svpriyan}@gmail.com

Abstract—**This research experiment ventures to find a solution for automating the framed signature recognition. Here signatures are made on a given frame using a ballpoint pen with a tip size of 0.5. Instead of direct neural networks based algorithm implementation, the extracted non-scale variant and scale variant features are used in a support vector machine in signature recognition algorithm. The outcome of the research appears as a GUI. The final outcome was 100% random signature isolation with over 88% trained forgery rejection. If not for 4 vulnerable signatures, this rate goes over 96% however the research carried out with worst environmental conditions and with least number of features. Thus, the results can be definitely improvable with modifications.**

Index Terms—**image preprocessing, feature extraction, signature recognition, offline**

I. INTRODUCTION

Most documents like bank cheques, travel passports are needed to be authorized by handwritten signatures for authentication, authorization and personal verification. Automating the signature recognition is not a new concept to the electronics field. And it has been patented before development of personal computers. The first patent of this field appeared in 1978 [1]. This dynamic signature feature recognition equipment was based on pressure imposed on the pen by the user. An improved version of it appeared after 7 years which extracts both pressure and acceleration [2]. However, the usages or analytical results obtained using these inventions are cannot be found in the internet.

By now, there are signature extraction devices based on touch pads which do not extract features of signatures but just save a copy of the signature image [3]. There are products like Smart pen, Internet pen, Bio pen which can extract dynamic signature information and verify signatures. But the intra class variations of dynamic features are usually higher. Its' true performances are uncertain because of the privacy and accuracy.

II. DISCUSSED PROBLEMS

Offline signature recognition [4] appeared after the evolution of computers and image processing techniques. It does not involve high-tech hardware designs and implementations; it is cost effective because of versatile application area.

For example, a good online recognition device which costs $500 is mounted with small panel of sensors and a processor to process parallel data inputs and to transmit them serially. Touch pad or a tablet is an alternative. Such device has many other applications. In online signature recognition, the person who signs should present at the input terminal. A pre-determined signature can only be recognized using offline methods. So the person is not practical in most situations. This area is still open for researching. The aim of this research is to develop a framed signature recognition algorithm based on findings of previous researchers' on offline signature recognition.

III. METHODOLOGY

The following activities describe the methodology. Discussing pre-processing of signature images, extracting eighteen selected scale variant features, extraction of grid features, discussing Euclidian distance based classifier and support vector machines with different kernels, extraction and effects of SIFT features on proposed algorithm and development of a simple GUI interface concludes the research.

A. Extraction of Signature Information

Extracting the information of the signature images was the first step. This type of system use scanned images of signatures in the algorithm. This research considers only framed signatures which are been asked to sign inside a given frame. The user should sign in a frame of 7 by 5 cm and should not exceed the given frame. The signatures collected from 45 difference users (5 from each) were scanned separately. The tip size of the pens used in this research is 0.5 and specifications regarding pens were not constrained. Thus minimum resolution of scanning was used in this research. The scanning resolution was set to 100 dpi. Fig. 1 shows an example.

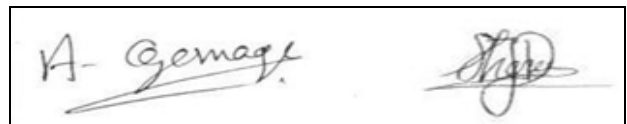After extracting the information, they were Pre-processed order to avoid poor results.

Figure 1. Two samples of cropped images, from Mr. Amalinda (left) and Mr. Dinushka (right).

B. Preprocessing

Preprocessing is a standard process in offline signature recognition. It consists of 4 stages: reduction, data cropping, width normalization and skeletonization. In this research data cropping was followed by noise reduction and skeletonization. The width normalization is not needed at first stages.

1) Conversion from grayscale to binary, im2bin function

The scanned images of the signatures are usually in grayscale. For this activity, the gray threshold level of a point of the signature does not have a big meaning. It may seemed that the intensity of a given point relates to the pressure imposed at that point, but researches shows the resemblance is so little to put into an algorithm. First, the images of scanned signatures are converted into cropped images. This research used the inbuilt function im2bw with a threshold value obtained using Otsu method under the inbuilt function graythresh. Fig. 2 shows an example.

Figure 2. The scanned image (left) and the black and white converted image (right).

2) Auto cropping

The above images have a white background. Background removing is called auto cropping [5]. In this research background, extended underlines and many other features which disturb consistency were removed, in order to prevent them affecting the algorithm adversely. Fig. 3 shows an example.

The logic followed is, the developed function called autocrop starts to evaluate the pixel densities of each row from the top and bottom edges and inspect the point at which the pixel density of the image exceeds a given threshold. In this research the pixel density threshold defined was 3 for columns as well. The obtained minimum values for rows and columns were stored in variables xmin, ymin and xmax, ymax. Thus a sub matrix is obtained.

Figure 3. The effects of auto cropping, image scanned (left), cropped image from the algorithm (right).

3) Morphological operations

Morphological operations in autocrop function are useful in cleaning and refining images. In this research the cropped images are "filled" and "cleaned". The cleaning is removing isolated black pixels in the image and filling is removing isolated white pixels for reducing salt and pepper type noise contamination.

4) Skeletonization

Skelton of an image describes the content of it. Skeletonization on the overall image carried out on a copy of the image at feature extraction. Inbuilt morphological function which would be visible inside feature extraction is used. Fig. 4 shows an example.

After preprocessing, the signatures cannot be compared due to their intra class variations. Thus, a set of suitable features should be selected and extracted from to develop a classifier is called feature extraction.

Figure 4. Effects of skeletonization; Original image (left), Skelton (right).

C. Feature Extraction

The comparison of two signatures is not a simple. It should be followed a suitable method like extracting features of signatures and comparing them. There are two main types of features: scale invariant features and scale variant feature. Scale invariant features can be further subdivided into grid and pixel distribution features as mentioned in [6] and [7].

1) Scale variant features

In this research 18 scale variant features has been identified and extracted.

1. Height of the signature after cropping
2. Width of the signature after cropping
3. The width to height ratio
4. Area subscribed by the image skeleton in pixels
5. Number of horizontal local maxima of the histogram of the Skelton image
6. Number of vertical local maxima of the histogram of the skeleton image
7. Number of horizontal local minima of the histogram of the Skelton image
8. Number of vertical local minima of the histogram of the skeleton image
9. Number of end points of the skeleton
10. Number of edges of the skeleton
11. Number of cross points of the skeleton
12. Horizontal center of gravity of the skeleton
13. Vertical center of gravity of the skeleton
14. Baseline shift
15. Number of loops in the skeleton
16. Maximum vertical researchion of the signature
17. Maximum horizontal researchion of the signature
18. Slant angle

2) Height width and width to height ratio

The height and width of the cropped signature was obtained using the inbuilt function "size" of matrix of the image. The evaluation of width to height ratio is thus straight forward.

3) Image area

The area of the image is measured taking the skeleton of the image and counting pixel in rows and adding up to get the image area prior to the evaluation.

4) Minima and maxima count of the histogram

The histogram of the skeleton can be obtained by evaluating the pixels in the columns and rows and comparing the subsequent pixel densities. The minima are evaluated using *vhminhysto* function and maxima using *vhhysto* function.

5) Number of end points, edge points and cross points

The end points resulted after thinning the skeleton is carried out using inbuilt morphological functions. The edge points have one neighbour. This is evaluated by developed function called *edge_cnt*. Here, 9 pixel frame is selected and the number of neighbours are inspected while moving the frame over the signature skeleton.. Here, a cross point should has at least three neighbours. The function used is *xpnt_cnt* which is similar to the above function in most aspects.

6) Horizontal and vertical center of gravity and baseline shift

The center of gravity evaluation is carried out via user developed function called COG. Having similar weights pixels is an advantage. To evaluate COG, positions of non-zero pixels are found and evaluate mean value. The baseline shift is the difference of vertical COG's of the image when its area is divided into two vertically as the left part and the right part. This gives a measure of pixel distribution in the image. The developed function is called baseline.

7) Number of loops

Calculating the number of loops in an image is challenging which has simple logic. At each cross point the number of leads should be calculated and the number of edges should be reduced for loops. Dividing the result by two and adding one to compensate the main loop gives the loop count. Fig. 5 shows an example.
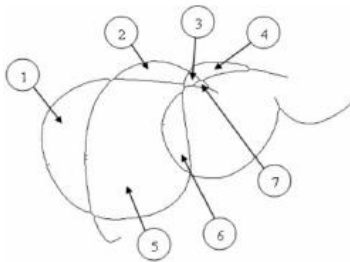

Figure 5. Closed loops

8) Maximum vertical and horizontal researchions and slant angle

The maximum vertical and horizontal researchions are values of the histograms. To evaluate the slant angle unrotate function is used. Thus signature is been rotated between 0 and -45 degrees with a resolution of three degrees. At each point the horizontal center of gravity is measured and at its maximum, the slant angle is defines as the angle by which it had been rotates.

9) Scale invariant features

Out of grid features, pixel distribution and SIFT features, grid features were found to be useful. Pixel distribution and SIFT features were found to be hopeless in this research.

10) Grid features

The grid feature evaluation, *grid_feat*, is carried out by casting the image into 8×12 matrix. The pixel distribution of each casted area is extracted and attached to the feature matrix after reshaping the resulting matrix. Fig. 6 shows an example.
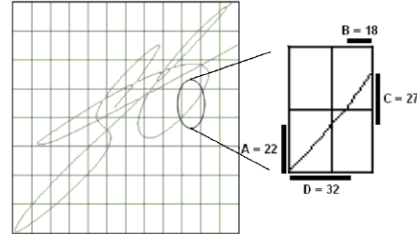

Figure 6. Evaluation of grid features

11) Pixel distribution

The pixel distribution defines the neighborhood inside the above grid. Values for the neighborhood is assigned based on the distribution and the result is presented as a feature matrix. However, implementing pixel distribution with grid features shifts the weight of scale invariant features up and lets the algorithm to neglect the effect of scale variant features, since there are only 18 scale variant features.

12) SIFT features

SIFT features are patented open source toolbox called vlfeat. The toolbox has function called vl_sift. When a grayscale image is fed, the sift features are extracted and gives out as two matrices. The first one contains SIFT descriptor information and second one has SIFT points. SIFT features are compared by the function vl_ubc match. The obtained results give five features: matched points, the minimum and maximum distances between the matched points of the images, the standard deviation and the mean distances of the matched SIFT points. After extraction of features, the classifier has to be developed.

D. Classifier Designs

Usually in pattern recognition, a single classifier is used for the experiment. Several classifiers were tested and the support vector machines were chosen for the final design. The classifiers tested are as below.

1) K nearest neighbour based classifier

The inbuilt knn classifiers were used here. The function contains knnclas classifier. When the database is with the user id, test signature and provided id, the algorithm tells whether the test signature falls into the said class or not. The knn classifier finds the k nearest neighbors to the test image and classifies the test image into the class to which most of the neighbors are belonged to. Fig. 7 shows am example.
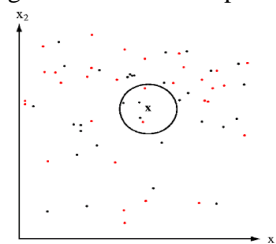

Figure 7. K nearest neighbor based classification example

2) Euclidian distance based classifier

The purpose of this classifier was to identify the mapping positions of the features. The feature matrices of a given signature are mapped and a suitable center is assumed to test signature deduction. According the observed distance values the classification is done.

3) Support vector machines

Support vector machine was the ultimate result. The inbuilt SVM in MATLAB was implemented after tuning. The support vector machines maps given features and finds the corresponding support vectors to decide a hyper plane or a boundary for the classes assigned. In MATLAB, only two classes can be assigned to the SVM. Fig. 8 shows the logic of SVM.

E. Graphical User Interface

Although algorithms do not create a need for developing a GUI, the common users find it hard to work in a command line. When the ID is typed, the authentic signature loads automatically into the window. User can test image using the test image button. If the signature is needed to be matched with a different ID, change ID, press the refresh button. Fig. 9 shows the GUI.
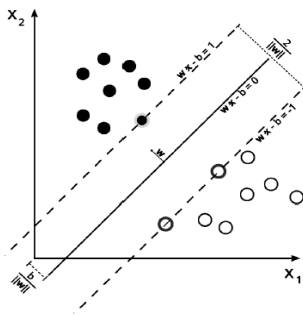

Figure 8. The logic of the SVM classifier: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.
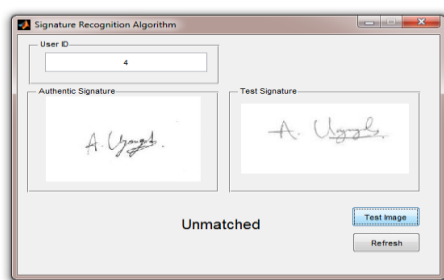

Figure 9. GUI of the algorithm

IV. RESULTS

The specificity for random signatures is 100%. That is, the algorithm can classify two different signatures as unmatched. The random matches carried out are 3916. The sensitivity obtained for the genuine signatures is 76.63%. Only 89 genuine signatures are used for this classification. The specificity obtained for the skilled forgeries is 88.89%. The number of forgeries implemented is 225. Four of the signatures collected were observed to get forged easily. If not for these signatures, the specificity observed will be 96%.

Equation (1) calculates the overall error rate observed for this research based on the number of samples tested.

$$Error = \frac{Total\,errorneous\,classification}{Total\,number\,of\,classifications\,carried\,out} \quad (1)$$

The equation gives a result of 0.011% error. That is, the rate of correct classification 99.989%. However, the figure is biased by higher number of samples of random signatures. If the mean value of the corresponding figures for the signature classes are used. The percentage accuracy obtained would become 88.5%. None of these overall figures describe the performance of the classifier correctly, but the figures given above for different signature classes

As in [4], the overall accuracy rate observed for neural approach is 96.7%; For HMM approach, the best results found was 1.44% for random forgeries, and 22.67% for skilled forgeries. Both figures are lesser than the observed figures. According to [7] the overall error rate observed is around 2.5% with a feature vector of 552 features. The feature vector used in obviously bigger, since, this research involves a feature vector which has only 114 features.

V. DISCUSSION AND CONCLUSIONS

The scanned image resolution is very low like 100 dpi. All the features that can be implemented are not been used in the classifier. Features like pixel distribution and texture features are not extracted from the signatures. The natural signatures comes out from the users are collected. Authentic signature makers were not been asked to carefully make a good copy of their signatures. Thus, the results obtained can be improved with modifications.

The resulting 100% random forgery isolation is very challenging. Such a result was not been mentioned to have obtained from offline signature verification. However, the results represented may not give a correct picture of the algorithm due to the fact of using only 450 signatures. Out of which 225 were skilled forgeries. Also, the results obtained were classified as random forgeries, genuine signatures and skilled forgeries. Thus, an overall figure for the algorithm performances cannot be given. On the other hand such a representation is not giving the observer a clear picture of the performance of the algorithm. In most of the referred documents, the performance is given as an overall figure. Thus, the results cannot be compared with most of the previous researches.

There is scope of studies and experiments to cover to give a final conclusion of this research. The cost is also negligible. The final result of the research is positive. The rates obtained as sensitivity and specificity is acceptable. This algorithm can be implemented in low end signature or pattern recognition applications, as digital locks or in character recognition applications. There are open areas for further improvements.

- Researching the algorithm behavior for higher resolution images like 200 dpi, for different grid sizes and for higher quality signature samples.

- Behavior of the algorithm for different scale variant and scale invariant features like pixel distribution, texture features etc.
- The grid size and the frame dimensions used for the signature verification may modify and checked its performances.
- The SIFT features found to fail to detect signature feature when used alone. But they can be researched with Markov chains and other techniques.

Not just using SVM, the algorithm may be able to implement with combining many classifiers. Such opportunities may also been tested.

REFERENCES

[1] S. Pepe, H. Dearbournes, and J. Brown, "Signature verification pen," USA Patent 4131880, Dec. 26, 1978.
[2] T. Chainer, R. Scranton, and T. Warkinton, "Data input pen for signature verification," USA Patent 4513437, Apr. 23, 1985.
[3] Bio pen. [Online]. Available: http://www.bio-pen.com/Shop/Product_List.asp
[4] V. S. Inamdar and M. S. Arya, "A preliminary study on various off-line hand written," *International Journal of Computer Applications*, vol. 1, no. 9, pp. 50-56, 2010.
[5] M. S. Altarawneh, I. H. Altarawneh, and B. Al-Mahadeen, "Signature region of interest using auto cropping," *International Journal of Computer Science*, vol. 7, no. 2, pp. 1-5, 2010.
[6] H. Baltzakis and N. Papamarkosb, "A new signature verification technique based on a two-stage neural network classifier," *Engineering Applications of Artificial Intelligence*, vol. 14, no. 1, pp. 95-103, 2000.
[7] G. Azzopardi and K. Camilleri, "Offline handwritten signature verification using radial basis function neural networks," St. Martin's Institute of IT, 2006.

Wiraj U. Wickramaarachchi was born in 15th of March, 1990. He is still following B.Sc. Special degree in Computing and information Systems at Sabaragamuwa University of Sri Lanka. Now he works at the same university as a Research Trainee. His current research interests include digital image processing, Bio Informatics, human computer interaction and etc. He has more than 4 years of experience in teaching. In addition, He is currently working as a visiting lecturer in several Private Institutes of Computing.

Shanmuganathan Vasanthapriyan was born in 9th of January, 1979. He has received his B.Sc. in Computer Science degree from University of Peradeniya, Sri Lanka in 2004. He has obtained his Master's degree in Computer Science and Engineering at Eindhoven University of Technology in Netherlands in 2010. He is currently pursuing his doctoral degree at Wuhan University of Technology in China. He has the experience of ten years as an academic lecturer at Sabaragamuwa University of Sri Lanka in Sri Lanka. His research interests are Computer Graphics, Visualization, image processing and Knowledge Engineering. He has published several research papers related to this profession so far.