Offline Signature Recognition Using Machine Learning

Mohammad Ikhsan Bin Zakaria, GunayKarli
Engineering and Information Technologies, International Burch University,
Sarajevo, Bosnia and Herzegovina.
E-mails: *mohammad.ihsan.z@gmail.com*, *gkarli@ibu.edu.ba*

Abstract

Biometric behavior can be recognized through the signature behavior of a person. It is mostly used for authorization and authentication in legal documentation papers. Signature recognition has two ways of verification, dynamic or online recognition and static or offline recognition. In this paper we use offline recognition to analyze signature images using Artificial Neural Network. We used mark minutia masking as the feature extraction. We proposed offline signature recognition using machine learning with supervised learning algorithm. The aim of using artificial neural network is to automatically find signatures that match to the owners of the signatures. Based on our evaluation, after we compared feed forward backpropagation and other supervised learning network such cascade-forward network, it revealed cascade-forward shown the highest accuracy100 % with low mean square error 0.

***Keywords*:** biometric, offline signature, machine learning

1.INTRODUCTION

Offline signature recognition is the technique to prevent forgery against security issue on legal documentation papers. In many legal companies they use this system to protect their customers. The process of gathering signature image is done by taking signatures from volunteers to sign on papers for ten times and we take that signatures scan to the computer and format as 200 dpi into gray scale image format. Reducing noisy and mark minutia arethe difficult tasks here, because besides we have to keep the information of signature images as valid as we can. There are few methods that applied offline signature recognition such as signature region of interest using auto cropping [1]. The signature images will be cleaned up from unwanted space or image around signatures. In this method the authors proposed image auto cropping as it is mentioned on image normalization. In [2] they proposed offline signature recognition and verification scheme which is based on extraction of several features including one hybrid set from the input signature and compare them with the already forms. In feature extraction [2] they used Euclidean distances from vertical and horizontal sectioning of signature. In [3] they proposed offline handwritten signature recognition which is trained in low-resolution scanned signature images using learning vector quantization classifier. The accuracy rate [3] was 98% for random test set of 150 handwritten signature images of 10 persons. Offline signature recognition and verification [4] based on four speed stroke was proposed. In [4] they used stroke angle and stroke speed as feature extraction.

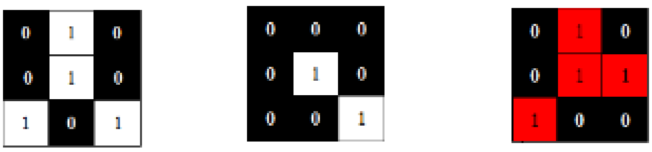
This paper is organized into five sessions. The following is an introduction of the topic in this session 1, session 2 describes the proposed method, in session 3 describe signature image preprocessing and feature extraction, in session 4 describes implementation, results. In final session describes conclusion.

2.SIGNATURE IMAGE PREPROCESSING

In this paper signature image preprocessing can be done in six steps as follows: (1) Histogram Equalization (2) Fourier Transform (3) Binarization (4) Signature Direction (5) Region of Interest (ROI) Area and (6) Thinning. Thinning image process is one most particular step in this stage, because thinning produces single layer line of signature. Minutia marking stage needs thinning before applying bifurcation skim step. Signature image preprocessing is influenced by the original which was taken using colors pen. Thinning process produces skeleton of signature which has single-pixel image.

2.1. Minutia Marking Feature Extraction

During image preprocessing, we include minutia marking as our feature extraction; here the mask digit skimmed all possible digits with 1s and 0s value. We carried out minutia marking to state image bifurcation and decision or termination. In general we have 3x3 matrices, if the central pixel is one and have exactly three one-value neighbors; the central pixel is a ridge branch. If the central pixel is one and has only one-value neighbor, then the central pixel is a ridge ending [5].Using minutia detection on the binary skeleton would be performed by labeling as minutiae pixels which is cross number (CN). Some methods consider the pixels which CN >= 3 correspond to bifurcation as shown in figure 1 (a) or if CN = 2 it correspond to ridge ending[5], [6].



(a) (b) (c)

Figure 1: (a) Bifurcation (b) Termination (c) Triple counting branch

Figure 1 (c) describes the special case which a genuine branch is triple counted. If both uppermost pixel with value 1 and the rightmost in same 3x3 block has pixel 1, so the two pixels are marked as the braches [6]. All three figures 1 (a), 1 (b) and 1 (c) are filtered using bifurcation template. Ridge thinning signature images are filtered using this bifurcation masking. In [5] discussed about mark minutia extraction. The bifurcation template is used to cover all possible high bit 1s and eliminate 0s bit after thinning process. Basically CN for pixel P in bifurcation template is in [5] and shown in figure 2 CN is estimated using equation (1).

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

Figure 2: Basic format CN for P

$$CN = \frac{1}{2}\sum_{i=1}^{8}|P_i - P_{i+1}| \tag{1}$$

Where Pi is the bi-level pixel value in the neighborhood of P with Pi = 0s or 1s and P1 = P9.

3. IMPLEMENTATION AND RESULTS

In implementation we used Artificial Neural Network supervised learning to classify signature images that are given in training and we tested to find the match of signatures and the owners. We evaluated the result in testing session. The experimental platform is the Intel dual core T3400 2.10GHz, 4 GB RAM, Windows 7 and the software is MATLAB 7.0.0.199 (R.14). On the first part of training and testing, we experimented feed-forward backpropagation and then followed by other supervised learning network such as Cascade-forward network, Elman Recurrent network and Learning vector quantization.

3.1. Proposed Method

The offline signature recognition using machine learning or Artificial Neural Network as proposed method in this study is illustrated in figure 3.

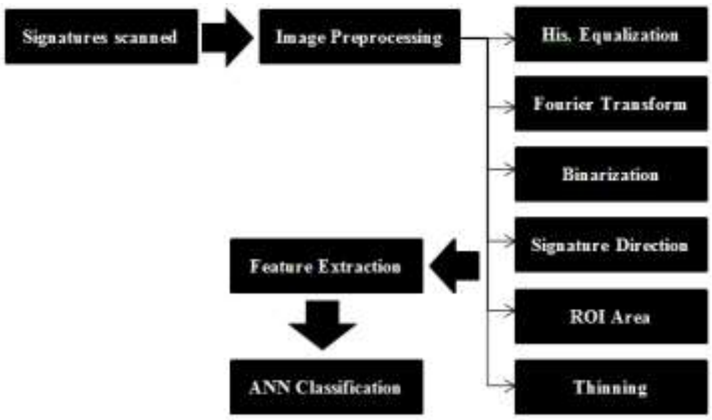


Figure 3: Block diagram of proposed method

The first step in the proposed method deals with collecting of signatures and scanned them, the second step describes signature image preprocessing in session 2. The third step describes feature extraction, in this step we used minutia marking. The final step describes the signatures classification processing using feed-forward backpropagation, cascade-forward network, Elman recurrent network and learning vector network. One of the sample testing results for each classification neurons are plotted in figure 4. Original or genuine signatures were collected from 30 students at International Burch University; each student gave 10 signatures samples. After converting 300 signatures into gray scale format, we divided them into 300 single signature images. The file was analyzed for neuron classification session. The following session describes ANN classification and testing results.

3.2. Feed-forward Backpropagation Network (newff)

In this experiment we used feed-forward backpropagation network to calculate mean square error as the measurement for performance on the neural networks. We also consider the influence of training algorithm and transfer function which can change the approximation of recognized signatures. In figure 4 (a) shows the example of testing results. In that testing session we obtained combination of attributes such as number of inputs, hidden layers, training algorithm and transfer function. It was the highest accuracy 66.6667 % and the lowest mse 0.4286. Table 1 shows the attributes training algorithm and transfer function influenced the final result of testing. The biggernumber of hidden layers with different combination of transfer functions, the bigger time it took the machine to analyze. Moreover, number of hidden layer and combination of transfer functions tansig or logsig did not make big changes or differences for accuracy rate. The lower result of mean square error, the higher the rate of accuracy we got. However the results of neural network testing were not precisely matched but we rounded into the nearest integers. After integers are rounded and there were compared with the predicted integers or classes.

Table 1 Testing on Feed-forward Backpropagation Networks

Input	Architecture of NN	Training Algorithm	Transfer Function	MSE	Accuracy
10	10-1	*traingdm*	*logsig, purelin*	0.7143	61.9048 %
10	10-1	*traingdm*	*tansig, purelin*	0.5714	57.1429 %
10	10-1	*traingdx*	*tansig, purelin*	0.5714	57.1429 %
10	10-10-1	*traingdm*	*tansig, logsig, purelin*	0.4762	66.6667 %
10	10-10-1	*traingdx*	*tansig, logsig, purelin*	0.4762	66.6667 %
20	20-10-10-1	*traingdm*	*tansig, logsig, logsig, purelin*	0.6190	52.3810 %
20	20-10-10-1	*traingdx*	*tansig, logsig, logsig, purelin*	0.7143	66.6667 %

20	20-10-10-1	*traingdm*	*logsig, tansig, tansig, purelin*	0.619 0	52.3810 %
20	20-10-10-1	*traingdx*	*logsig, tansig, tansig, purelin*	0.428 6	66.6667 %

The performance of training is influenced by number of hidden layers, training algorithm, learning methods. Generally,mseis calculated in MATLAB using logic below. In equation (2) it is just additional description of calculating mse using MATLAB. In equation (3), we used the logic to compare between target output and actual output. We calculate the integers in target output that are larger or equal to actual output and converted them into 1s.

$$\text{error} = \text{target output} - \text{actual output;}$$

$$\text{mse} = \text{mse(error);} \tag{2}$$

$$\text{if target}_{\text{output}} \geq \text{actual}_{\text{output}}$$

$$\text{calculate number of target}_{\text{output}} \text{ that matched to actual}_{\text{output}} \text{ end;}$$

$$\text{efficiency} = \left(\frac{\text{number of matched}}{\text{all testing signatures}}\right) * 100 \tag{3}$$

3.3. Cascade-forward Network (newcf)

Table 2 shows training and testing using cascade-forward networks, we calculated the mseto find the significant error during our testing.

Table 2 Testing Cascade-forward Networks

Input	Architecture of NN	Training Algorith m	Transfer Function	MSE	Accuracy
10	10-1	*trainlm*	*logsig, purelin*	0.4286	71.4286 %
10	10-1	*trainlm*	*tansig, purelin*	0.4762	66.6667 %
10	10-1	*trainbfg*	*tansig, purelin*	0.4286	57.1429 %
10	10-10-1	*trainlm*	*tansig, logsig, purelin*	0.3810	76.1905 %
10	10-10-1	*trainbfg*	*tansig, logsig, purelin*	0.5238	61.9048 %
20	20-10-10-1	*trainlm*	*tansig, logsig, logsig, purelin*	0.0952	90.4762 %

20	20-10-10-1	*trainbfg*	*tansig, logsig, logsig, purelin*	0.5238	61.9048 %
20	20-10-10-1	*trainbfg*	*logsig, tansig, tansig, purelin*	0.4762	52.3810 %
20	20-10-10-1	*trainlm*	*logsig, tansig, tansig, purelin*	0	100 %

Our attributes in table 2 are training algorithm trainlm and trainbfg, where during testing session trainbfg spent more time than trainlm to find output. In final testing we obtained 20 inputs with two hidden layers and tansig as transfer function, we got 100 % matched in accuracy rate and 0 in mse error. Thus we concluded that the lowest mse in this network produced the highest accuracy we got. However, mse does not always affect the changes of accuracy rate or neural network output. It is because the output of neurons is not always precise. As a sample of training and testing, figure 4 (b) shows testing result. Figure 4 (b) shows the testing result with mse 0.4286 and accuracy rate was 71.4286 %.

3.4. Elman Recurrent Network (newelm)

The basic structure table in Elman networks is the same as previous networks in feed-forward backpropagation and cascade-forward networks as shows in table 3.

Table 3 Testing on Elman Recurrent Network

Input	Architecture of NN	Training Algorithm	Transfer Function	MSE	Accuracy
10	10-1	*trainlm*	*logsig, purelin*	0.4286	57.1429 %
10	10-1	*trainlm*	*tansig, purelin*	0.1429	85.7143 %
10	10-1	*trainbfg*	*tansig, purelin*	0.6190	66.6667 %
10	10-10-1	*trainlm*	*tansig, logsig, purelin*	0.8095	71.4286 %
10	10-10-1	*trainbfg*	*tansig, logsig, purelin*	0.4286	71.4286 %
20	20-10-10-1	*trainlm*	*tansig, logsig, logsig, purelin*	0.7143	57.1429 %
20	20-10-10-1	*trainbfg*	*tansig, logsig, logsig, purelin*	0.7143	42.8571 %

20	20-10-10-1	*trainlm*	*logsig, tansig, tansig, purelin*	0.0476	95.2381 %
20	20-10-10-1	*trainbfg*	*logsig, tansig, tansig, purelin*	0.4762	95.2381 %

In this experiment the lowest mse is 0.0476 and the highest accuracy is 95.2381 %. From table 3 shows that there are two highest accuracy rates but with difference mse, thus the best output is the one that has lower mse error, even though it has same accuracy and uses same inputs, hidden layer but different training algorithms. Trainlm shows the lowest mse result. As a sample of testing session in this network, figure 4 (c) shows 71.4286 % accuracy and 0.8095 mse.

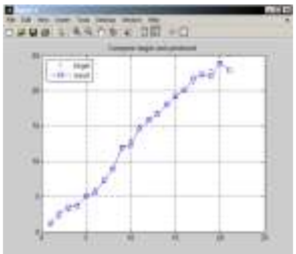
3.5. Learning Vector Quantization (newlvq)

In learning vector quantization, the hidden layer value has to be positive integers so it became limited for us to analyze. Relating to the classes, we provided 21 classes of signatures. We trained 105 signatures and we tested using 21 signatures. In excel file we put addition column as the name of each classes such as class 1 has five 1s, class 2 has five 2s and so on. So here we provided different kind of table which consists only training algorithm, mse and efficiency.

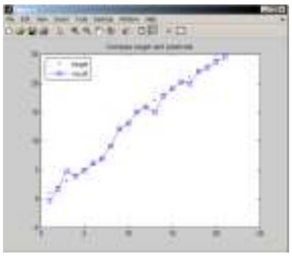
Table 4 Training and testing *newlvq*

No. Hidden Neurons	Class Percentages	Training Algorithm	MSE	Accuracy
10	.6 .4	*learnlv2*	0.4286	71.4286 %
20	.6 .4	*learnlv2*	0.4286	71.4286 %
10	.6 .4	*learnlv1*	0.4286	71.4286 %
20	.6 .4	*learnlv1*	0.4286	71.4286 %
10	.8 .2	*learnlv2*	0.4286	71.4286 %
10	.8 .2	*learnlv1*	0.4286	71.4286 %

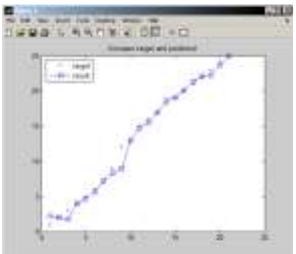
Table 4 (d) illustrates combination of learning algorithm, typical of classes and number of hidden neurons. The results show us, there are no significant changes during testing either using learnlvq1 or learnlvq2 and hidden neurons. Even though, we combined all possible values. Thus learning vector quantization gave the highest accuracy 71.4286 % with 0.4286 mse.



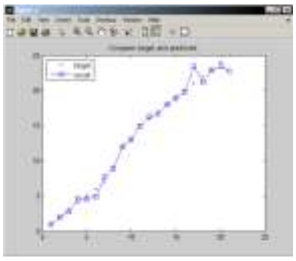
(a)



(b)



(c)



(d)

Figure 4: (a) Feed-forward backpropagation, (b) Cascade-forward, (c) Elman Recurrent (d) Learning Vector Quantization

4. CONCLUSION

Based on experiments in previous chapter, we can conclude few points which related to the results. The highest accuracy in feed-forward backpropagation testing result was 66.6667 % and the lowest mse in that network was 0.4286. In cascade-forward network testing, the highest accuracy rate was 100 % and the lowest mse in that testing was 0. Moreover, when we tested Elman, the highest accuracy in that testing network was 95.2381 % and mse was 0.0476.On the other hand, learning vector quantization network has some differences in attributes. For instance, we used learnlv1 or learnlv2 as learning algorithm and compet as training algorithm, so we don’t compare this network with other three network algorithms in previous evaluation. The highest accuracy in learning vector quantization was 71.4286 % with 0.4286 mse. Thus cascade forward network was the best fit in this method, because the network produced 0 errors and 100 % accuracy with 20 inputs.

REFERENCES

Souvola, J. &Pietikainen, M. (2000), Adoptive document image binarization, The Journal of The Pattern Recognition Society, page 225-236.

Bhuyan, M., Sarma, K. K., & Das, H. (2010). Signature Recognition and Verification using Hybrid Features and Clustered Artificial Neural Network (ANN). International Journal of Electrical and Computer Engineering.

Khuwaja, G. A. &Laghari, M. S. (2011). Offline Handwritten Signature Recognition. World Academy of Science, Engineering and Technology 59.

Basavaraj, L. &Sudhaker Samuel, R.D. (2009). Offline-line Signature Verification and Recognition: An Approach Based on Four Speed Stroke Angle. International Journal of Recent Trends in Engineering, Vol 2.

Zhao, F., & Tang, X. (2006). Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction, Pattern Recognition 40 (2007) 1270 – 1281, The Journal of Pattern Recognition Society.

Zhili, W. (2002). Fingerprint Recognition. Unpublished Bachelor’s Thesis, Hong Kong Baptist University.