Offline Signature Verification through Probabilistic Neural Network

Ooi Shih Yin[1], Andrew Teoh Beng Jin[2], Hiew Bee Yan[1], Pang Ying Han[1]

[1]Faculty of Information Science and Technology
Multimedia University,
Jalan Ayer Keroh Lama,
75450 Melaka, Malaysia.
{syooi, byhiew, yhpang}@mmu.edu.my

[2]School of Electrical and Electronic Engineering,
Yonsei University, Seoul,
South Korea.
bjteoh@yonsei.ac.kr

ABSTRACT

In this paper, we show the positive potential of verifying the offline handwritten signatures through discrete Radon transform (DRT), principle component analysis (PCA) and probabilistic neural network (PNN). Satisfactory results are obtained with 1.51%, 3.23%, and 13.07% equal error rate (EER) for random, casual, and skilled forgeries respectively on our independent database.

Keywords

Offline signature verification, discrete Radon transform, principle component analysis, probabilistic neural network.

1. INTRODUCTION

Offline signature verification has been the subject of considerable research for over 34 years. It is an old pattern classification problem of genuine and forgery 2-D scanned signature images. There are three popular groups of forgery: casual forgery, random forgery and skilled forgery. Skilled forgery is produced by the professional forger that has unrestricted practice to the writer's actual signatures. A casual forgery is produced by the forger who is familiar with the writer's name, but never expose to a sample of the actual signature. Therefore, stylistic differences are prevalent in this case. A random forgery is any random scribble, a genuine signature or a high quality forgery for other writer. Skilled forgery detection emerged as the most challenging task even for expert document examiners.

This paper's main objective is to distinguish a genuine signature from the forged signature. The major challenge is to distinguish between the variations among genuine signatures and the true differences between a signature and a forgery. However, the differences between a genuine signature and a skillfully forged one always can be subtle.

2. LITERATURE REVIEW

Numerous methods and approaches done over two decades are summarized in a number of survey articles. The state of the art before 1989 was discussed by Plamondon and Lorrette [Pla89] and the period from 1989 to 1993 was covered by Leclerc and Plamondon [Lec94]. At 2000, Plamondon and Srihari [Pla00] published a survey which covered the state of the art from the period of 1993 to 2000. Guo et al. [Guo01] included an extensive overview of previous works as well. From the survey, we can see that earlier work on offline signature verification deals primarily with casual and random forgeries, where deceit is generally obvious. As signature databases become larger, researchers are moving toward to more difficult skilled forgery detection task, which is still an open research question. There are plenty of pattern recognition techniques being used in this field. However, we will primarily focus on the neural networks in this work.

A neural network is a computing paradigm that is loosely modeled after cortical structures of the brain.

It consists of interconnected processing elements (neurons) that work together to produce an output function. The output is relies on the cooperation of the individual neurons within the network to operate. Neural networks often process the information parallel rather than in series (or sequentially). Since it relies on its member neurons collectively to perform its function, a unique property of a neural network is that it can still perform its overall function even if some of the neurons are not functioning. Thus, they are very robust to error or failure. It has been extensively used in offline signature verification over the last two decades. Few relevant researches are summarized below; however, due to the lack of standard database available, all results reported are based on the researcher groups' own independent database.

Mighell, Wilkinson, and Goodman [Mig89] proposed a backpropagation learning algorithms to detect random forgeries. By training 10 genuine signatures and 10 forgeries respectively, which latter tested on 70 genuine signatures and 56 forgeries, they reported a false rejection rate (FRR) of 1% with a false acceptance rate (FAR) of 4%.

Abbas [Abb94] investigated the suitability of using multilayered feedforward neural networks for the task of offline verification. The input to the network is a binary bitmap of size 160 X 35 pixels. The performance is evaluated against their private database of 480 signatures. They concluded that the method is the best for the casual forgeries where able to achieve 0% FAR but its ability to deal with skilled forgeries was still limited with FAR ranging from 0% to 60%.

Qi and Hunt [Qi95] proposed a multi-resolution approach to allocate the offline signature verification problem. The top-level representation of signatures is the global geometric features. A multi-resolution representation of signature is obtained using the wavelet transformation. By using a database of 450 signatures from 25 signatories, the classification is done through a vector quantization (VQ) classifier and an artificial neural network classifier respectively. VQ classifier allows the use of a consistent procedure in processing feature vectors of different length or resolution, and it is easy to implement because its training and classification procedures are relatively simple. However, it can only partition the feature space using hyperspheres, and is incapable of drawing complicated, nonlinear class boundaries. While, artificial neural network is capable of delineating arbitrarily complicated class boundaries, anyway, the performance is heavily depends on the network architecture and training method. The best VQ classification function is the accumulative, multi-resolution system which reported on FRR of 6.7%, FAR of 13.3% for skilled forgery and FAR of 0% for simple forgery. On the other hand, the multi-resolution network yields the lowest verification error rate when independent features are used, FRR of 4.0%. FAR of 9.3% for skilled forgery and FAR of 1.3% for simple forgery are reported.

Kaewkongka, Chamnongthai, and Thipakorn [Kae99] proposed to use the Hough transform (general Radon transform) as the feature extractor. It extracts the parameterized Hough space from a signature skeleton as a unique characteristic feature of a signature. Evaluation is done through a backpropagation neural network. By using the dataset of 70 signatures, recognition rate of 95.24% is reported.

Quek and Zhou [Que02] proposed a system which is constructed on the basis of a novel fuzzy neural network called the POPFNN-TVR, which has a five-layer structure. Due to its characteristics, such as the learning ability, generalization ability, and high computational ability, it is very powerful to detect the skilled forgeries. After preprocessing, feature extraction is employed to reduce the image observation vector by measuring certain "properties" or "features" of the signature image. In this work, four kinds of features are extracted from the static image of the signature, which including reference pattern based features, global baseline, pressure features and slant features. All of them will be using as elements of the training vector. Two types of experiments are then conducted; first experiment is using the genuine signatures and forgeries as training data, while the second experiment is using only the genuine signatures as training data. Based on the signatures of 15 different signatories from 3 ethnic groups, the average of the individual EER, 22.4% is obtained for the first experiment. While for the second experiment, they claimed that comparable results are obtained.

Piyush Shanker et al. [Piy07] proposed an offline signature verification by using Dynamic Time Warping (DTW). They extract the vertical projection feature from the signature images, and comparing the reference and probe feature templates using elastic matching. The method is tested against the original DTW and modified DTW. The modified DTW achieved EER 2% which outperformed the original DTW at 29%.

Recently, Abdala Ali and Zhirkov [Abd09] proposed an offline signature verification comparing against Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) classifiers. Their system achieves approximately 80% when using SVM, while approximately 70% for KNN.

Bansal et al. [Ban09] proposed an offline signature verification using critical region matching. This work is mainly focus on the extraction of critical regions which are more prone to mistakes and matching through a modular graph matching approach. They reported 10.81% EER for skilled forgery.

3. OVERVIEW OF WORK

Generally, an offline handwritten signature verification system includes preprocessing, feature extraction and encoding as well as matching as depicted in Fig. 1. These processes will be further discussed in the following sections.

4. PREPROCESSING

Any ordinary scanner with enough resolution can be used as an image acquisition device. However, the scanning hardware may introduce certain noises to a signature image. Another source of noise may be speckled paper background on which the signature is signed on. These noises on signature image may thwart the feature extraction process. We do not figure the real noise distribution, but we use the median filter, which better preserves edges, lines, and corners.

After the smoothing, the images are converted into black-and-white images by using Adobe Photoshop. The threshold level is set to 100.

5. FEATURE EXTRACTION

Discrete Radon Transform (DRT)

DRT [Coe04] is chosen to transform the signature images into a feature space. It is able to transform two dimensional images with lines into a domain of possible line parameters, where each line in the image will give a peak positioned at the corresponding line parameters. DRT has several advantages. Each signature is a static image and contains no dynamic information, thus by calculating projections at different angles, simulated time evolution is created from one feature vector to the next, where the angle represent the dynamic variable [Coe04]. DRT represents a projection (shadow) of the signature at different angle. A set of transform values is produced after the transformation. The DRT of an image can be calculated as follows. Assume that each signature image consists of N pixels in total, and that the intensity of the ith pixel is denoted by I_i, $i = 1,\ldots,N$. The DRT is calculated using β non-overlapping beams per angle and Θ angles in total. The cumulative intensity of the pixels that lie within the jth beam is denoted by R_j, $j = 1,\ldots, \beta\Theta$. This is called the jth beam sum. In its discrete form, the Radon transform can therefore be expressed as $R_j = \sum_{i=1}^{N} w_{ij} I_i,\ j = 1,2,\ldots,\beta\Theta,$ where w_{ij} indicates the contribution of the ith pixel to the jth beam sum [Coe04]. The value of w_{ij} is determined by two-dimensional interpolation. Each projection therefore contains the beam sums that are calculated at a given angle.

Instead of Hough transform, we preferred DRT because it has a nice effect of attenuating the speckle noise in the images through summation, while the use of Hough transform is very delicate especially on noisy images.

Principle Component Analysis (PCA)

PCA has been widely used for dimensionality reduction in computer vision ([Lu03], [Tur91], and [Wan03]). It finds a set of orthogonal basis vectors which describe the major variations among the training images and with minimum reconstruction means square error. The successful implementation of PCA in various recognition tasks popularized the idea of matching images in the compressed subspaces.

Since the number of transformed values after DRT is too huge, PCA is utilized here for feature data compression. In the PCA method, the average of K DRT features with M dimension is defined as R_{avg}.

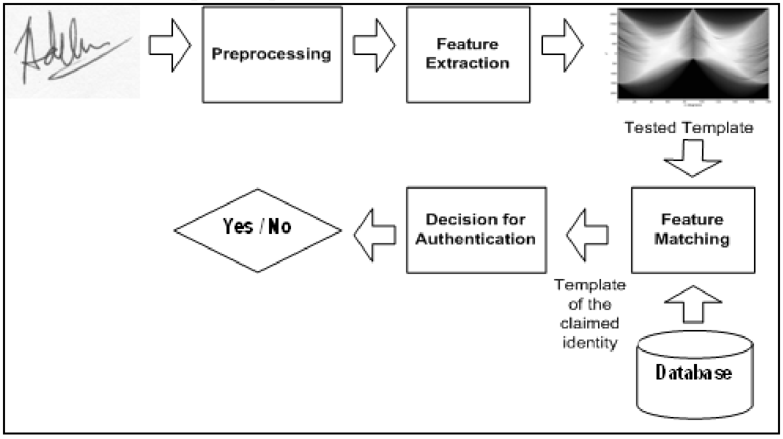

Figure 1. Block diagram of an offline handwritten signature verification.

Then, eigenvectors, v_{k} and eigenvalues, λ_k with symmetric matrix C are calculated. v_{k} determines the linear combination of K difference images with φ to form the EigenSignature, $U_l = \sum_{k=1}^{K} v_{lk}\varphi_k \;\square\; l = 1,\ldots,K$.

Then, $P(<K)$ EigenSignatures are chosen to correspond to the P highest eigenvalues, which imply that the P features are selected. An input DRT feature, R_k is transformed and projected into the EigenSignature space by the operation, $\rho_k = U_k(R_k - R_{avg})$, where $k = 1,\ldots,P$.

Probabilistic Neural Network

Rather than ordinary matching approaches that are based on similarity matching concept, there is another popular method used for classification which the idea is to construct the decision boundaries directly by optimizing an error criterion. PNN which was first introduced by Specht ([Spe88], [Spe90]) is one such technique. It offers several advantages over backpropagation network. The rationale behind this is that, as a kernel-based approach to probability density function approximation, PNN posses the advantages to handle the complex, non-linear and imprecise problems such as signature verification.

In general, a PNN consists of three layers – a pattern, summation and output layers (apart from the input layer) as illustrated in Fig. 2. The pattern layer contains one neuron for each input vector in the training set, while the summation layer contains one neuron for each user class to be recognized. The output layer merely holds the maximum value of the summation neurons to yield the final outcome (probability score).

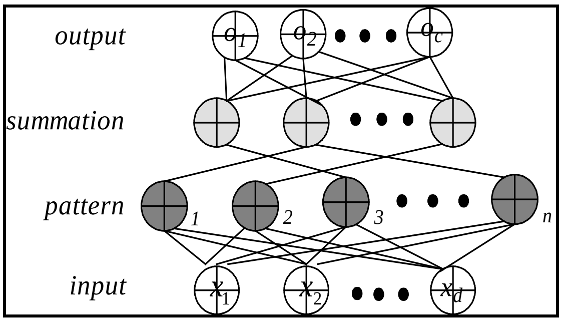

Figure 2. Basic configuration of a probabilistic neural network.

The network can simply be established by setting the weights of the network using the training set. The modifiable weights of the first layer are set by $\omega_{ij} = \rho_{ij}$ where ω_{ij} denoting the weight between ith neuron of the input layer and jth neuron in the pattern layer, and ρ_{ij} is the j element feature of ρ_i in the training set. The second layer weights are set by $\omega_{jk} = \mathrm{T}_{jk}$, where ω_{jk} is the weight between neuron j in pattern layer and neuron k of the output layer, and 1 is assigned to T_{jk} if pattern j of the training set belongs to user k and 0 otherwise. After the network is trained, it can be used for classification task. The outcome of the pattern layer is defined as $out_j\omega = \exp\left(-\sum_{i=1}^{m}(\rho_i \quad _{ij})/\sigma\right)$.Note that out_j is the output of neuron j in pattern layer and σ is the smoothing parameter of the Gaussian kernel which is the only independent parameter that can be decided by the user. The input of the summation layer is calculated as $in_k = \sum_{j=1}^{n} out_j \times \omega_{jk}$ where in_k is the input of neuron k in output layer. The outputs of the summation layer are binary neurons that produce the classification decision, i.e 1 is assigned to out_k if in_k is larger than the input of others neurons and 0 otherwise.

The smoothing parameters ($\sigma_1, \sigma_2, \ldots,$ and σ_j) need to be carefully determined in order to obtain an optimal network. This factor needs to be selected to cause a reasonable amount of overlap; too small deviations will cause a very spiky approximation which cannot generalize, while too large deviations smooth out detail. An appropriate figure is easily chosen by experiment, by selecting a number which produces a low selection error, and fortunately PNNs are not too sensitive to the precise choice of smoothing factor. For convenience sake, we use a straightforward procedure to select the best value for σ . Firstly, an arbitrary value of σ is chosen to train the network, and then test it on a test set. This procedure is repeated for other σ 's values and the σ giving the least errors will be selected.

The motivation of using a PNN is driven by the generalization property and simple training scheme (only one epoch of training is required) of PNN. However, the speed of training is achieved at the cost of increase in complexity and computational/ memory requirements. The time complexity for training is O(nP), where n denotes the number of training samples and P is the length of PCA feature data. In our context, the time complexity of PNN that depends on the P and n can be decreased notably due to the compressed feature data length. As such, the association of DRT and PNN is feasible in practical usage due to its high speed and accuracy performance.

6. EXPERIMENTS & DISCUSSIONS

Database and Setup

Our independent database comprised of 1000 genuine signatures, 500 casual forgeries, and 500 skilled forgeries which were collected from 100 writers and 10 forgers. Due to the non-repetitive nature of variation of the signatures, the signatures produced will have certain variations among same writers. Thus, the data preparation was mainly divided into two stages. In the first stage, five sample signatures are registered per writer at a single contact session producing 500 samples. In the second stage, another set of five genuine signatures were supplied by the same writer during the contact sessions two weeks after the initial session, yielding another 500 samples. Thus, by recording the specific date, we can observe the variations among the same signature for a single session and different sessions. For the forgery part, the casual forgeries are obtained first; the forgers only allow viewing the writer's name but did not have the access to the signatory's signatures. The skilled forgeries are then obtained from the same group of forgers. We provided them with several samples of each signatory's genuine signature and they are allowed ample opportunity to practice on it.

The pen or pencil used by each writer is not prescribed but signatures are written within a pre-drawn 5 x 2 grid on A4 paper. These signatures were scanned into the computer using a 24-bit millions of colors, 600 dot-per-inch resolutions. The individual images are extracted and labeled with both the writer names and the signature class number.

We will evaluated the system based on false acceptance rate (FAR), false rejection rate (FRR), and equal error rate (EER).

Performance Evaluations

This method is evaluated by using random, casual and skilled forgeries from the mentioned independent database.

Four samples of each person are sequentially selected for Eigen basis construction and the remaining six samples are used for testing. To investigate the performance of PCA against the DRT-extracted signature images as the dimensionality reduction agent, we use different number of principle components (or feature length), varying from 10 – 200, as shown in Table 1.

It is interesting to discover that longer feature length leads to better result. The performance peaks when 100 principle components are used. However, this principle only holds to a certain point as the experimental results show that the result remains unchanged when the feature length is extended further. Thus, the PCA length is set to 100 for the following experiments.

Next, we investigate the performance of DRT by using three different distance metrics, which are cosine angle distance, L_1 (Manhattan) and L_2 (Euclidean) distance measure for random random (Fig. 3), casual (Fig. 4) and skilled (Fig. 5) forgeries respectively. DRT β is taken to be equal to the highest dimension of the image (300 X 200 pixels after smoothing and converted into black-and-white image), which is 300, and works on $\Theta = 128$.

From the experiment, the cosine angle distance is outperforming towards L_1 (Manhattan) and L_2 (Euclidean) distances. This is because cosine angle distance usually gives a higher rank to vectors with larger variance (whereas applied to signature images) among its components.

Number of PCA Feature Length	**Random Forgery** **(EER, %)**	**Casual Forgery** **(EER, %)**	**Skilled Forgery** **(EER, %)**
10	8.75	12.00	23.00
30	8.33	11.65	22.45
50	7.45	11.00	21.00
80	7.11	10.20	20.22
100	**6.95**	**9.87**	**19.56**
120	6.95	9.87	19.56
150	6.95	9.87	19.56
180	6.95	9.87	19.56
200	6.95	9.87	19.56

Table 1. Equal error rates (EER, %) of using different number of principle components

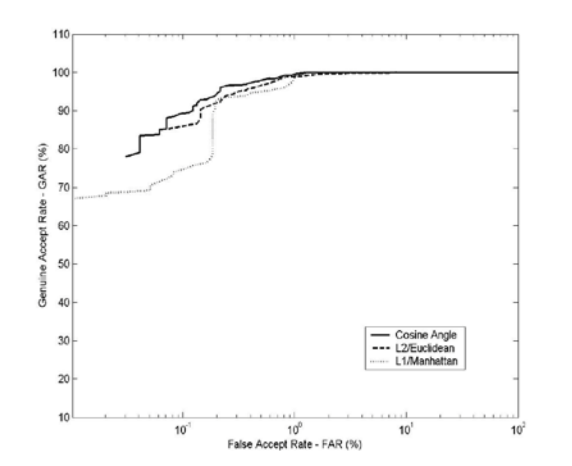

Figure 3. Receiving Operating Characteristic (ROC) curve of random forgery for three different distance metrics: cosine angle, L_1 (Manhattan) and L_2 (Euclidean) respectively.

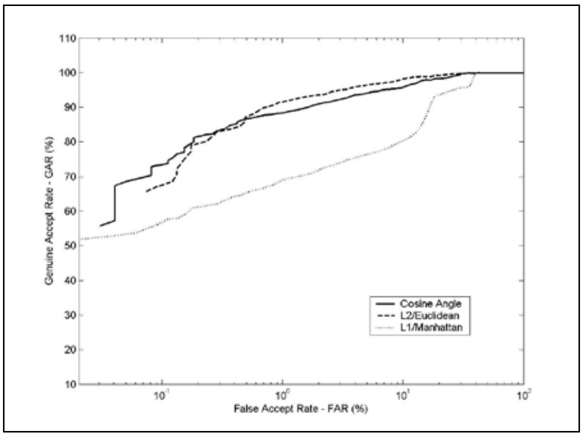

Figure 4. Receiving Operating Characteristic (ROC) curve of casual forgery for three different distance metrics: cosine angle, L_1 (Manhattan) and L_2 (Euclidean) respectively.

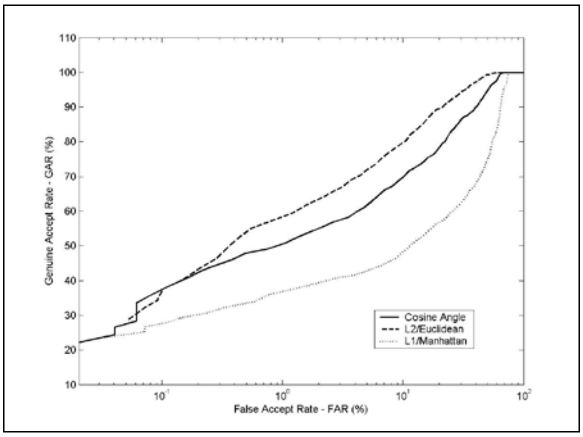

Figure 5. Receiving Operating Characteristic (ROC) curve of skilled forgery for three different distance metrics: cosine angle, L_1 (Manhattan) and L_2 (Euclidean) respectively.

However, it can be anticipated that the classification accuracy of the methods will improve when a more sophisticated classifier, PNN is used. In our system, $^{10}C_4$ = 210 runs are performed with different partitions between the training and testing sets by using a PNN smoothing parameter of σ = 10.

From the ROC curve showing in Fig. 6, the performance is greatly improved especially for casual and skilled forgeries. Table 2 summarizes the performance of PNN towards random, casual and skilled forgeries.

Besides, the experiment also shows that the computation time can be reduced significantly with just slight performance drop when only one template per user is used (as compared to the case of 4 training samples shown in Table 3 for skilled forgery). In this case, the time complexity of PNN that depends on the number of training samples, *n* and the length of PCA feature data, *P* can be decreased notably due to the compressed feature data length through PCA and single training sample per user settings. As such, the association of DRT, PCA and PNN is feasible in practical usage due to its high speed and accuracy performance.

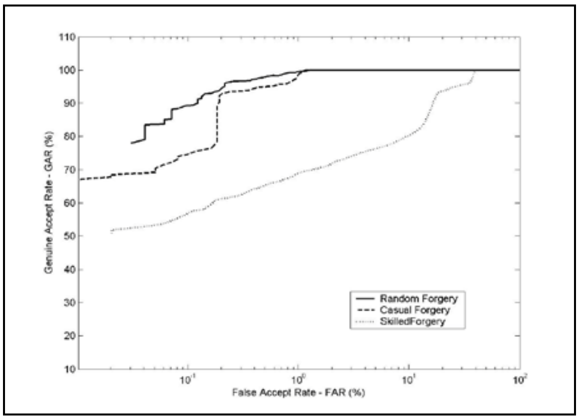

Figure 6. Receiving Operating Characteristic (ROC) curve for random, casual and skilled forgeries respectively when using: Eigen basis construction set = 4, principle component length = 100 when classified through PNN.

	FAR(%)	FRR(%)	EER(%)
Random Forgery	1.50	1.52	1.51
Casual Forgery	3.22	3.24	3.23
Skilled Forgery	12.98	13.16	13.07

Table 2. FAR, FRR and EER achievement (%) for random, casual and skilled forgeries respectively

Training Samples	Total time (minutes)	EER (%)
4	38.5	13.07
1	14	14.20

Table 3. Total time spent to run one course of experiment and the accuracy of PNN in skilled forgery context

Comparison with Other Research Groups' Techniques

It is very difficult to compare the performance of different signature verification systems due to the fact that different systems are using different signature data sets. The lack of a standard international signature database is a big problem for performance comparison.

However, few works that published in year 2009 including Piyush Shanker et al. [Piy07] , Abdala Ali and Zhirkov [Abd09] (we implement only on SVM) and Bansal et al. [Ban09] algorithms have been implemented and tested in our own independent database due to the close-similarity of our implementation details.

	Piyush et al.	Ali and Zhirkov	Bansal et al.	Our method
Random Forgery	1.45	1.13	1.23	1.51
Casual Forgery	3.21	2.43	3.15	3.23
Skilled Forgery	13.05	11.55	12.58	13.07

Table 4. Equal error rates (EER, %) of implementing different approaches towards our independent database

	Piyush et al.	Ali and Zhirkov	Bansal et al.	Our method
Random Forgery	125.0	120.0	80.0	38.5
Casual Forgery	125.0	120.0	80.0	38.5
Skilled Forgery	125.0	120.0	80.0	38.5

Table 5. Computation times (minutes) of different approaches towards our independent database

Referring to Table 4, it can be concluded that their algorithms are slightly outperform our method. However, by referring to Table 5, we can say that our system is more favorable in real world application context due to its shortest computation time. Piyush Shanker et al.'s modified DTW is stable, but somehow it is still not particularly fast. Abdala Ali and Zhirkov's SVM is powerful, but very time consuming to select the appropriate kernel functions and determining the belonging parameters during the development phase. Bansal et al.'s algorithm performs slightly better than ours, but required longer processing time.

7. CONCLUSIONS

This paper proposed an offline signature verification through DRT, PCA and PNN. The high accuracy is feasible to filter the forgery from the genuine signature, especially for skilled forgery; while the speed of the PNN is very favorable in real-world application. The results are encouraging and thus should motivating the research on skilled forgery detection especially for offline handwritten signature.

8. ACKNOWLEDGMENTS

Our thanks and appreciations to those referred work listed in literature.

9. REFERENCES

[Abb94] Abbas, R. A prototype system for offline signature verification using multilayered feedforward neural networks. Minor Thesis, 1994.

[Abd09] Abdala Ali, A.A., and Zhirkov, V.F. Offline signature verification using Radon transform and SVM/KNN classifiers. Transactions of Tambov State Technical University, no.1, pp.62-69, 2009.

[Ban09] Bansal, A., Gupta, B., Khandelwal, G., and Chakraverty, S. Offline signature verification using critical region matching. International Journal of Signal Processing, Image Procesing and Pattern, vol.2, no.1, 2009.

[Coe04] Coetzer, J., Herbst, B.M., and du Preez, J.A. Offline signature verification using the discrete Radon transform and a hidden Markov model. EURASIP Journal on Applied Signal Processing, vol.4, pp.559-571, 2004.

[Guo01] Guo, J.K., Doermann, D., and Rosefeld, A. Forgery detection by local correspondence. International Journal of Pattern Recognition and Artificial Intelligence, vol.15, no.4, pp.579-641, 2001.

[Kae99] Kaewkongka, T., Chamnongthai, K., and Thipakorn, B. Offline signature recognition using parameterized Hough transform. Proceedings of the Fifth International Symposium on Signal Processing and Its Applications, 1999.

[Lec94] Leclerc, F., and Plamondon, R. Automatic signature verification: the state of the art, 1989-1993. International Journal of Pattern Recognition and Artificial Intelligence, vol.8, no.3, pp.643-660, 1994.

[Lu03] Lu, G., Zhang, D., and Wang, K. Palmprint recognition using Eigenpalms features. Pattern Recognition Letters, vol.24, no.9-10, pp.1473-1477, 2003.

[Mig89] Mighell, D.A., Wilkinson, T.S., and Goodman, J.W. Backpropagation and its application to handwritten signature verification. Advances in Neural Information Processing Systems, vol.1, pp.340-347, 1989.

[Piy07] Piyush Shanker, A., and Rajagopalan, A.N. Offline signature verification using dynamic time warping. Pattern Recognition Letters, vol.28, no.12, pp.1407-1414, 2007.

[Pla89] Plamondon, R., and Lorette, G. Automatic signature verification and writer identification – the state of the art. Pattern Recognition, vol.22, no.2, pp.107-131, 1989.

[Pla00] Plamondon, R., and Srihari, S.N. Online and offline handwritten recognition: a comprehensive survey. IEEE Transaction on Pattern Analysis and

Machine Intelligence, vol.22, no.1, pp.63-84, 2000.

[Qi95] Qi, Y.Y., and Hunt, B.R. A multiresolution approach to computer verification of handwritten signatures. IEEE Transactions on Image Processing, vol.4, no.6, 1995.

[Que02] Quek, C., and Zhou, R.W. Antiforgery: a novel pseudo-outer product based fuzzy neural network driven signature verification system. Pattern Recognition Letters, vol.23, no.14, pp.1795-1816, 2002.

[Spe88] Specht, D.F. Probabilistic neural networks for classification, mapping, or associative memory. Proceeding of the IEEE International Conference Neural Networks, vol.1, no.2, pp.525-532, 1988.

[Spe90] Specht, D.F. Probabilistic neural networks (original contribution). Neural Networks, vol.3, no.1, pp.109-118, 1990.

[Tur91] Turk, M.A., and Pentland, A.P. Eigenfaces for recognition. Journal of Cognitive NeuroScience, vol.3, no.1, pp.71-86, 1991.

[Wan03] Wang, X., and Kuldip, K.P. Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition. Pattern Recognition, vol.36, no.10, pp.2429-2439, 2003.