ISSN: 1226-7244 (Print)
ISSN: 2288-243X (Online)
논문번호 15-04-11

j.inst.Korean.electr.electron.eng.Vol.19,No.4,526~534,December 2015
http://dx.doi.org/10.7471/ikeee.2015.19.4.526
67

Offline Handwritten Numeral Recognition Using Multiple Features and SVM classifier

Gab-Soon Kim* , Joong-Jo Park*★

Abstract

In this paper, we studied the use of the foreground and background features and SVM classifier to improve the accuracy of offline handwritten numeral recognition. The foreground features are two directional features: directional gradient feature by Kirsch operators and directional stroke feature by local shrinking and expanding operations, and the background feature is concavity feature which is extracted from the convex hull of the numeral, where the concavity feature functions as complement to the directional features. During classification of the numeral, these three features are combined to obtain good discrimination power. The efficiency of our scheme is tested by recognition experiments on the handwritten numeral database CENPARMI, where SVM classifier with RBF kernel is used. The experimental results show the usefulness of our scheme and recognition rate of 99.10% is achieved.

Key words: Numeral recognition, Directional stroke feature, Gradient feature, Concavity feature, SVM

I. Introduction

Handwritten numeral recognition is a typical example among complex pattern recognition problems, and is an active topic in the practical OCR applications such as zip code recognition, document analysis, bank check processing, job application form sorting, automatic sorting of tests containing multiple choice questions and factory sautomation system, etc. To achieve high recognition results, several methods have been proposed and implemented in a number of different ways [1]-[13]. The performance of an overall numeral recognition system generally depends on the features used and the designed classifier. Among them, extracting good features is very important in order to achieve a high recognition performance. Curved shapes could be used for extracting numeral features because numerals consist of curved strokes. As one of the most effective numeral features, directional features, which represent the local stroke directions of numerals, have been widely used and have yielded high performances [9],[10],[12]. To further improve the recognition rate, local structure features [10],[23], curvature features [11], concavity features [10],[12],[13] or projection features [14],[15] have been combined by many researchers. Conventionally, directional features have been extracted from chaincode or gradient maps, where the gradient can be computed with various operators, such as Roberts, Sobel, and Kirsh [9],[10],[13]. On the other hand, the performance of numeral recognition also largely depends on the classification scheme. Statistical techniques such as nearest neighbor, Bayes discriminant function, linear and quadratic discriminant functions [10]-[13], neural networks

* Dept. of Control and Instrumentation Engineering, Gyeongsang National University
★ Corresponding author
e-mail : jjpark@gnu.ac.kr, 055-772-1743
Manuscript received July 24, 2015; revised Nov. 10, 2015; accepted Dec 10. 2015

[3]-[9],[12]-[14],[24],[25], and support vector machine [12]-[15] have been widely used for classification due to their implementation efficiency.

In our earlier work [3], we developed a new directional feature extraction method, which extracts the local directional strokes of a numeral by using directional shrinking and expanding operations. While directional features extracted from gradient maps by Kirsch operators represent the directions of the edges of local strokes, directional features by our method represent the directions of local strokes themselves. We showed that these two kinds of directional features cooperated to improve the recognition performance. In this paper, we use the foreground and background features and SVM classifier for high recognition rates of handwritten numerals. The foreground features are directional gradient feature by Kirsch operators and directional stroke feature by our method, and the background feature is concavity feature which is extracted from the convex hull of the numeral and contains the background information of a numeral image. The concavity feature complements the directional features to increase the discriminative power [12],[16],[19]. We use the support vector machine (SVM) as a classifier and the CENPARMI database as the handwritten numeral test data. SVM [18], which is based on the statistical learning theory and quadratic programming optimization, has been known as an efficient classifier, and its superior classification performance has been justified in numerous experiments [12],[13],[15]. CENPARMI database has been used in classifier design and numeral recognition research for performance evaluation [12],[13].

This paper is organized as follows. Section II describes the feature extraction techniques and Section III describes the multi-class calssifier by using several SVM classifiers. Section IV presents the experimental results and Section V contains concluding remarks.

II. Feature Extraction

In this section, we describe three feature extraction methods for handwritten numeral recognition: directional gradient features by Kirsch operators, directional stroke features by local shrinking and expanding operations and concavity features.

2.1 Directional gradient features

Generally, the directional features are obtained in three steps: image normalization, direction extraction, and feature measuring. We use the Kirsch operator to compute the gradient (strength and direction) from the numeral image. Unlike other operators, such as Roberts and Sobel, the Kirsch operator directly gives the strengths of four directions [9],[12]. And compared with chain code, the Kirsch operation is more robust for noisy images [14]. Directional gradient features are calculated, as described below:

(i) A binary input image is denoised by removing the convex pixels and filling in the concave pixels. Thereafter, size normalization by using a bilinear interpolation algorithm is applied to the image so that the image has a standard width and height. In our experiments, the size of a normalized image was set to 32×32.

(ii) Kirsch operators are applied to each pixel of the normalized image to obtain four gradient images. Thereafter, four directional images are generated by binarizing each of the four gradient images. These four binary directional images can be viewed as directional sub-images derived from the original image.

(iii) For feature measuring, each binary directional image is partitioned into 16 (4 horizontal x 4 vertical) uniform zones and the number of '1' pixels in each zone is counted. Then, the measurement of each zone is calculated by normalizing these counts to a range of values [0,1]. A simple equation for the calculation of these features is given as follows:

$$F_l^k = \begin{cases} 1 & , \text{if } s_i^k \geq T \\ s_i^k / T & , otherwise \end{cases} \quad (1)$$

$$where,\ s_i^k = \sum_{y_i} \sum_{x_i} g^k(x,y)$$

Where, F_l^k is the feature value of the i^{th} zone in the k^{th} directional image, and s_i^k is the number of '1' pixels in the i^{th} zone of the k^{th} directional image, and $g^k(x,y)$ is the pixel value of the k^{th} directional image at the position x and y. The threshold value T should be carefully chosen after several preliminary experiments.

This way, a directional gradient feature vector of size 64 (4 horizontal, 4 vertical, and 4 directional resolutions) is produced.

2.2 Directional stroke features

The basic idea of this feature extraction method is described as follows: First, the thickness of numeral lines in an input image is changed to a 3-pixel width by using thinning and expanding operations. Then, from this processed image with numeral lines of 3-pixel width, four directional images (horizontal, vertical, and the two diagonal directions) are generated by eroding the image in each direction by using four directional shrinking and expanding operations. Finally, directional stroke features are obtained from these four directional images by using the zoning method.

This method uses the directional shrinking and expanding operations in addition to a 4-neighbor expanding operation, which are defined as follows: The 8-neighborhood of x_0 in a binary image is shown in Fig. 1. The pixel value of x_k (0 or 1) is denoted by f_k. Then, in the directional shrinking operation, the pixel value at x_0 for direction i is calculated by:

$$f_s^{[i]}(x_0) = \min\{f_i,\ f_0,\ f_{i+4}\},\ i = 1,2,3,4 \quad (2)$$

Similarly, in the directional expanding operation, the pixel value at x_0 for direction i is calculated by:

$$f_e^{[i]}(x_0) = \max\{f_i,\ f_0,\ f_{i+4}\},\ i = 1,2,3,4 \quad (3)$$

And, in the 4-neighbor expanding operation, the pixel value at x_0 is calculated by:

$$f_E(x_0) = \max\{f_k,\ k = 0,1,3,5,7\} \quad (4)$$

x_4	x_3	x_2
x_5	x_0	x_1
x_6	x_7	x_8

Fig. 1. 8-neighborhood of a pixel x_0

By using above operations, Directional stroke features can be obtained, as described below:

(i) A binary input image is denoised by removing the convex pixels and filling in the concave pixels. Thereafter, the size-normalized image, in which the thickness of numeral lines is 3-pixel width, is obtained as follows: First, a thinning operation is applied to the denoised image and the size of the thinned image is adjusted by cutting out four outside blank lines so that the image is fully filled with its numeral pattern. After that, size normalization by using a bilinear interpolation algorithm is applied to the image so that the image has a standard width and height. Thereafter, the resulting image is binarized and a thinning operation is applied again so that the thickness of numeral lines is 1-pixel width. Lastly, a 4-neighbor expanding operation of Eq. (4) is applied to the thinned image, to make the thickness of numeral lines to be 3-pixel width. In our experiments, the size of a normalized image was set to 24×24.

(ii) Next, we generate four directional images by applying directional shrinking operations of Eq. (2) two times to the preprocessed image obtained in step (i). In this process, if Eq. (2) with direction $i=1$ is applied to the preprocessed image, the strokes of the horizontal direction remain and the strokes of all the other directions disappear by shrinking effect, because the thickness of the numeral lines in the preprocessed image is 3-pixel width. Likewise, if Eq. (2) with direction $i=3$ is applied to the preprocessed image, only the strokes of the vertical direction survive. In this process, surviving strokes have been over-shrinked along their directions. So, the directional expanding operation of Eq. (3) is applied one time to each processed directional image for compensation. Fig. 2 shows four directional images obtained by this method, compared with those images obtained by the Kirsch operation method.

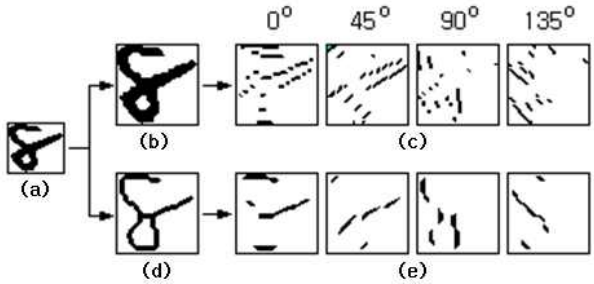

Fig. 2. Comparison of directional images: (a) input image; (b) size normalized image of (a); (c) directional images from (b) by Kirsch operation; (d) preprocessed image of (a); (e) directional images from (d) by our method.

(iii) For feature measuring, each binary directional image is partitioned into 16 (4 horizontal x 4 vertical) uniform zones and the zoning method of subsection 2.1 is applied, in which Eq. (1) is also used for the calculation of a feature vector.

This way, a directional stroke feature vector of size 64 (4 horizontal, 4 vertical, and 4 directional resolutions) is produced.

2.3 Concavity features

The concavity features, which represent the background information of a numeral, are used as complements to the directional features [12],[16],[19].

The concavity features are extracted from the convex hull of a numeral, as shown in Fig. 3. Concavity features are calculated, as described below:

(i) A binary input image is denoised by removing the convex pixels and filling in the concave pixels. Thereafter, size normalization by using a bilinear interpolation algorithm is applied to the image so that the image has a standard width and height. In our experiments, the size of the normalized image was set to 40×40.

(ii) The normalized image is binarized and a convex hull image is obtained from this binary image.

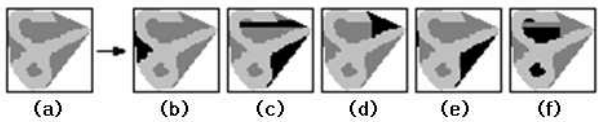

Fig. 3. Five concavity images from a convex hull image (each image is superimposed with numeral strokes): (a) convex hull image; (b) left-opening; (c) right-opening; (d) top-opening; (e) bottom-opening; (f) closing.

(iii) Concavity features are extracted from the convex hull image superimposed with numeral strokes as shown in Fig. 3(a). Let's call this image a revised convex hull image. All image pixels inside the convex hull and outside the numeral strokes are considered as concavity pixels, and by examining concavity pixels, five types of concavity images are extracted: left-opening, right-opening, top-opening, bottom-opening and closing. The left-opening concavity image is obtained as follows: The revised convex hull image is scanned from left to right. All the concavity pixels which the scanning line intersects before hitting the numeral pixel form the object pixels of the left-opening concavity image. Likewise, the right-opening, top-opening, bottom-opening concavity images are obtained by scanning the revised convex hull image from right to left, top to bottom and bottom to top, respectively. Lastly, the concavity pixels which do not belong to the four above concavity images become the object pixels of the closing concavity image. In every concavity image, the values of object pixels are 1's and the values of the others are 0's. Fig. 3 shows five concavity images obtained by this method. In this figure, each concavity image is composed of black pixels.

(iv) For feature measuring, each binary concavity image obtained in step (iii) is partitioned into 25 (5 horizontal x 5 vertical) uniform zones and the zoning method of subsection 2.1 is applied, in which Eq. (1) is also used for the calculation of a feature vector. This way, a concavity feature vector of size 125 (5 horizontal, 5 vertical, and 5 concavity types) is produced.

When extracting two directional features in subsections 2.1 and 2.2, we used different sizes for normalization, that is, 32x32 for directional gradient features by Kirsch operation and 24x24 for directional stroke features by our method. The reason for this is explained as follows: Directional images for directional gradient features contain the directional edges of strokes, so that directional strokes are represented as two lines. On the other hand, the directional images for directional stroke features contain directional strokes themselves which are represented as one line. In image processing, properly representing the edges of strokes requires more space than representing the strokes themselves. Accordingly, the size of normalization for directional gradient features was set to be larger than the size of normalization for directional stroke features.

III. Multi-class classifier by SVMs

To obtain high recognition rates, we use Support vector machine (SVM) as a classifier. SVM is a powerful learning technique based on statistical learning theory and quadratic programming optimization [17],[18]. In the last few years, SVM has played an important role in machine learning research, due to its excellent generalization performance [19],[20],[22].

Basically, SVM is a binary classifier. For our task, we need multi-class SVMs. The simplest strategies for implementing multi-class SVMs are "one against the rest" and "one against one". In the "one against the rest" strategy, k SVM classifiers are constructed, where k is the number of classes. For each class, it builds a two-class classifier with positive and negative class labels, where the samples of the considered class are labeled as positive while all samples of the remaining classes are labeled as negative. Then, it obtains k decision functions by solving k binary problems. The right class is determined by the largest positive value of the decision functions. In the "one-against-one" strategy, k(k-1)/2 two-class SVM classifiers are constructed from each pair of classes for the given k-class problem. The same number of classifiers is trained on each of the two-class data by labeling two counterpart classes as positive and negative, respectively. For the testing stage, the voting scheme known as "Max Wins" algorithm is commonly used. Among these two strategies, we used the "one-against-one" multi-class strategy for numeral classification.

In SVM implementation, the optimization method should be used to solve the quadratic programming problem by using training samples. We used the SVM library [21], which is a fast SVM training algorithm that deals with large scale data [22].

IV. Experiment and results

In our experiments, we used three different features and the SVM classifiers to recognize totally unconstrained handwritten numerals. Our experiments were performed on the well-known CENPARMI database of handwritten numerals. It contains 6000 numeral images collected from the envelope images of USPS. In this database, 4000 images (400 samples per class) are specified for training and the remaining 2000 images (200 samples per class) are specified for testing. All images are binary and vary in size. Figure 4 shows the examples of the CENPARMI numeral. The SVM classifiers were implemented using the SVM library [21], and image processing and feature extractions were performed using C code.

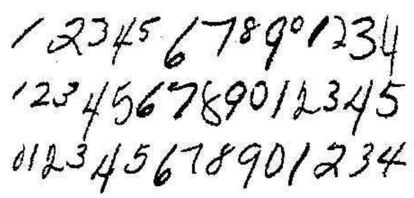

Fig. 4. Some examples of numeral data

For handwritten numeral recognition, we used three types of features: directional gradient features by Kirsh operator, directional stroke features by local shrinking and expanding operations and concavity features. Because images of the CENPARMI database vary in size, size normalization was applied to a binary input image before extracting each feature. The size of the normalized image was set to 32×32, 24×24 and 40×40 for the directional gradient features, the directional stroke features and the concavity features, respectively. We used a bilinear interpolation algorithm for size normalization. From these normalized images, two sets of directional images were obtained for two directional features and concavity images were obtained for the concavity features. Thereafter, by using the zoning method, two directional features were extracted from the pertinent directional images and the concavity features were extracted from the concavity images. In the zoning method, the measurement of each zone was calculated and normalized to a range of values [0,1] by using Eq. (1), where threshold values used for normalization were 16, 10 and 54, for the directional gradient features, the directional stroke features and the concavity features, respectively. These threshold values were carefully chosen after several preliminary experiments by using the SVM with individual features. The extracted feature vectors are listed as follows:

- Directional gradient features (*grad*): [4x4], 4-directions
- Directional stroke features (*strk*): [4x4], 4-directions
- Concavity features (*conc*): [5x5], 5-concavity types

We used SVM classifiers for classification. The feature vectors were scaled before they were used in classifiers in the training and testing stages [19],[22]. In our experiments, scaling was defined as making the Euclidian distance between any two feature vectors lies within the interval of 0 and 1. Scaling of a feature vector was achieved by dividing the feature vector by the scaling factor, where the scaling factor is equal to the farthest distance between two feature vectors in a training feature set.

Since SVM was a binary classifier, the "one-against-one" method was used to construct a ten-class classifier for our numeral recognition. On the given feature sets, k(k-1)/2 two-class SVM classifiers, where k=10, were trained on the training data set, and then the trained classifier was used to classify the test data set. For SVM, we selected the RBF kernel, which is known for giving the highest accuracy [12],[20]. In experiments with each feature set, we varied the values for the SVM parameters C and σ, and searched for the best ones which correspond to the highest recognition rate, where C is a parameter to control the tolerance of classification errors in training and σ is the radius parameter of the Gaussian RBF kernel. In the testing stage, the voting scheme "Max Wins" was used as the decision method.

Table 1 and Table 2 show how the classification accuracy is influenced by the SVM parameters C and σ. We select these two parameters according to the results of many preliminary experiments. These two tables show the results of preliminary experiments. Table 1 shows the recognition rates vs. σ^2 when C is set to 10. This result shows that in the case of testing each single feature set, the directional features(*grad, strk*) are superior to the concavity features(*conc*), and the directional stroke features (*strk*) is slightly better than the directional gradient features(*grad*) in recognition performance. The highest recognition rates for each single feature set occurred at different σ^2

values. The peak recognition rate was achieved at $\sigma^2 = 0.1$ for the composite feature made by two directional feature sets and one concavity feature set. This table shows that the classification accuracy was influenced by the value of the SVM parameter σ. Table 2 shows the recognition rates vs. parameter C when σ^2 is set to 0.1. This result shows that the parameter C value influences the classification accuracy of each single feature set, but has no effect on the classification accuracy of the composite feature made by the three feature sets.

Table 1. Recognition rates(%) vs. σ (C = 10)

σ^2 / used features	0.06	0.08	0.1	0.12	0.14
grad	97.80	97.60	97.65	97.70	97.60
strk	97.75	97.90	97.85	97.80	97.70
conc	97.15	97.10	97.20	97.20	97.10
grad + strk + conc	98.75	98.95	99.10	99.00	98.90

Table 2. Recognition rates(%) vs. C (σ^2 = 0.1)

C / used features	2	6	10	14	18
grad	97.95	97.70	97.65	97.65	97.65
strk	97.85	97.85	97.85	97.85	97.85
conc	96.90	97.20	97.20	97.25	97.20
grad + strk + conc	99.10	99.10	99.10	99.10	99.10

Table 3 shows the recognition rates of individual and composite features when the SVM parameters are set as C=10 and $\sigma^2 = 0.1$. From this table, we can see that each combination of two feature sets gives a better performance than a single feature set. This means that each single feature set works with a different discriminating power and it cooperates with other feature sets to enhance the recognition accuracy. By using the composite feature of the three feature sets, we obtain a recognition rate of 99.10%. Table 3 also shows the recognition rates[3] by MLP(multilayer perception) neural network classifier for comparison, For impartial comparison, the same database should be used in the experiments. In reference [3], the CENPARMI numeral database was used, and this table showed that performance of our work was improved by using the discriminative power of SVM.

Table 3. Recognition rates (%) of individual and composite features (C=10, σ^2=0.1)

Multiplicity	used features	recognition rate	
		SVM	M:P [3]
single feature set	*grad*	97.65	96.45
	strk	97.85	96.20
	conc	97.20	96.09
two feature sets	*grad + strk*	98.95	96.65
	grad + conc	98.35	97.69
	strk + conc	98.70	97.70
three feature sets	*grad + strk + conc*	99.10	98.35

In our experiments, we could see that there was several misclassified numerals, although they looked recognizable by their distinct shapes. This means that further research to generate the features with more discriminating power is necessary.

The computation time for the three feature extraction methods and SVM classifier have been obtained on a personal desktop computer with Intel Core i5-2400 CPU 3.1GHz 4.00GB RAM., Each method was implemented using the C code. Table 4 lists the computation times obtained by each method, which shows that the total computation time of our method for one numeral is 0.016107 seconds. This speed seems to be good for several applications.

Table 4. Computation time for each step of our scheme [millisecond]

feature extraction			classification	total
grad	*strk*	*conc*	*SVM*	
0.882	1.685	3.034	10.506	16.107

V. Conclusion

In this paper, we use the foreground and background features and SVM classifier to improve the accuracy of handwritten numeral recognition. In our scheme, the foreground features are two kinds of features: the directional gradient feature by conventional Kirsch operators and the directional stroke feature by our method, and the background feature is the concavity feature which functions as complement to the directional features.

The efficiency of our scheme was tested by recognition experiments on the handwritten numeral database CENPARMI, where we used SVM with RBF kernel as a classifier. The experimental results showed that each combination of two or three feature sets gave a better performance than a single feature set. This means that each single feature set works with a different discriminating power and it cooperates with other feature sets to enhance the recognition accuracy. By using the composite feature of the three feature sets, we achieved a recognition rate of 99.10%.

References

[1] O.D. Trier. A.K. Jain and T. Taxt, "Feature Extraction Methods for Character Recognition -A Survey", Pattern Recognition, Vo.29, No.4, pp.641-662, 1996.

[2] N. Arica and F.T. Uarman-Vural, "An Overview of Character Recognition Focused on Off-Line Handwriting", IEEE Transactions on Systems, Man, And Cybernetics -Part C: Application and Reviews, Vol.31, No.2, pp. 216-233, May 2001.

[3] S.W. Jung, J.J. Park, “Directional Feature Extraction of Handwritten Numerals using Local min/max Operations", Journal of The Institute of Signal Processing and Systems, Vol.10, No.1, pp.7-12, 2009.

[4] A.D. Parkins and A.K. Nandi, "Method for calculating first-order derivative based feature saliency information in a trained neural network and its application to handwritten digit recognition", Vision, Image and Signal Processing, IEE Proceedings Vol.152, Issue 2, pp.137-147, 2005

[5] U. Bhattacharya, S.K. Parui, B. Shaw, K. Bhattacharya, "Neural Combination of ANN and HMM for Handwritten Devanagari Numeral Recognition", 10th-International Workshop on Frontiers in Handwriting Recognition(IWFHR), pp. 613-618, 2006

[6] D.J. Romero, L.M. Seijas, A.M. Ruedin, "Directional Continuous Wavelet Transform Applied to Handwritten Numerals Recognition Using Neural Networks", Journal of Computer Science and Technology, Vol.7, No.1, pp.66-71, 2007.

[7] Ujjwal Bhattacharya and B.B. Chaudhuri, "Handwritten Numeral Databases of Indian Scripts and Multistage Recognition of Mixed Numerals", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vo.31, No.3, pp.444-457, 2009

[8] Rajeshwar Dass and Seema Asht, "Handwritten Numeral Pattern Recognition Using Neural Network", International Journal of Computer Science and Telecommunications, Vol.3, Issue 9, pp.1-5, 2012.

[9] S.W. Lee, "Off-line Recognition of Totally Unconstrained Handwritten Numerals Using Multilayer Cluster Neural Network", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.18, No. 6, pp.648-652, 1996.

[10] J.T. Favata and G. Srikantan, "A Multiple Feature/Resolution Approach to Handprinted Digit and Character Recognition", International Journal of Imaging Systems and Technology, Vol.7, pp.304-311, 1996.

[11] M. Shi, Y. Fujisawa, T. Wakabayashi and F. Kimura, "Handwritten Numeral Recognition Using Gradient and Curvature of Gray Scale Image", Pattern Recognition, Vol.35, pp.2051-2059, 2002.

[12] C.L. Liu, K. Nakashima, H. Sako and H. Fujisawa, "Handwritten Digit Recognition: Benchmarking of State-of-the-art Techniques", Pattern Recognition, Vol.36, pp.2271-2285, 2003.

[13] C.L. Liu, K. Nakashima, H. Sako and H. Fujisawa, "Handwritten Digit Recognition: Investigation of Normalization and Feature Extraction Techniques", Pattern Recognition,

Vol.37, pp.265-279, 2004.

[14] D. Gorgevik and D. Cakmakov, "An Efficient Three-Stage Classifier for Handwritten Digit Recognition", Proceedings of 17th Int. Conference on Pattern Recognition, ICPR2004, Vol. 4, pp. 507-510, IEEE Computer Society, Cambridge, UK, 23-26, August 2004.

[15] D. Gorgevik, D. Cakmakov, "Handwritten Digit Recognition by Combining SVM Classifiers", Proceedings of EUROCON 2005, pp.1393-1396, Serbia & Montenegro, Belgrade, November 22-24, 2005.

[16] Chun Lei He, *Error Analysis of a Hybrid Multiple Classifier system for Recognizing Unconstrained Handwritten Numerals*, A Master Thesis in the department of Computer Science & Software Engineering, Concordia University, May. 2005.

[17] B. Boser, I. Guyon and V. Vapnik, "A Training Algorithm for Optimal Margin Classifiers", Proceedings of the fifth annual workshop on Computational learning theory, pp. 144-152, 1992.

[18] C. Cortes and V. Vapnik, "Support Vector Networks", Machine Learning, Vol.20, pp. 273-297, 1995.

[19] Sapargali Kambar, *Generating Synthetic Data by Morphing Transformation for Handwritten Numeral Recognition (With v-SVM)*, A Master Thesis in the department of Computer Science & Software Engineering, Concordia University, Aug. 2005.

[20] Hui Guo and Huayan Zhang, "Chinese Classifier Assignment Using SVMs", Fourth SIGHAN Workshop on Chinese Language Processing, pp.25-31, 2005.

[21] J. X. Dong, *HeroSVM Support Vector Machine User Guide*, 2002, Software available at http://www.cenparmi.concordia.ca/~jdong/HeroSvm.html

[22] J. X. Dong, C. Y. Suen and A. Krzyzak, "A Fast SVM Training Algorithm", International Journal of Pattern recognition and Artificial Intelligence, Vol.17, No.3, pp.367-384, 2003.

[23] B. S. Saritha, S. Hemanth, An Efficient Hidden Markov Model for Offline Handwritten Numeral Recognition, InterJRI Computer Science and Networking, Vol. 1, Issue 1, pp.7-12, 2009.

[24] Pritpal Singh, Sumit Budhiraja, Offline Handwritten Gurmukhi Numeral Recognition using Wavelet Transforms, I.J.Modern Education and Computer Science, pp.34-39, 2012.

[25] Lisha Singla, Shamandeep Singh, Offline Handwritten Devanagari Numerals Recognition using GLCM Features & Neural Networks, International Journal of Engineering Research & Technology, Vol. 3 Issue 6, pp.1476-1480, 2014.

BIOGRAPHY

Gab-Soon Kim (Member)

1986 : BS degree in Precision Mechanical Eng., Jeonbook National University.

1990, 1999 : MS and Ph.D degrees in Precision Mechanical Eng., Hanyang University.

1990~2000 : senior researcher, Korea Research Institute of Standards and Science.

2003~2004 : visiting researcher, Dept. of Intelligent Robot Lab., Univ. of Tsukuba.

2000~Present : professor, Dept. of Control and Instrumentation Eng., Gyeongsang National University.

Joong-Jo Park (Member)

1981, 1988, 1995 : BS, MS and Ph.D degrees in Electrical Eng., Korea University.

1996~Present : professor, Dept. of Control and Instrumentation Eng., Gyeongsang National University.