A Static Hand Gesture Recognition Algorithm Using K-Mean Based Radial Basis Function Neural Network

Dipak Kumar Ghosh
Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela, India
Email: dipakkumar05.ghosh@gmail.com
Telephone:0661-2464464

Samit Ari
Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela, India
Email: samit.ari@gmail.com
Telephone:0661-2462464

***Abstract*—The accurate classification of static hand gestures is a vital role to develop a hand gesture recognition system which is used for human-computer interaction (HCI) and for human alternative and augmentative communication (HAAC) application. A vision-based static hand gesture recognition algorithm consists of three stages: preprocessing, feature extraction and classification. The preprocessing stage involves following three sub-stages: segmentation which segments hand region from its background images using a histogram based thresholding algorithm and transforms into binary silhouette; rotation that rotates segmented gesture to make the algorithm, rotation invariant; filtering that effectively removes background noise and object noise from binary image by morphological filtering technique. To obtain a rotation invariant gesture image, a novel technique is proposed in this paper by coinciding the 1^{st} principal component of the segmented hand gestures with vertical axes. A localized contour sequence (LCS) based feature is used here to classify the hand gestures. A k-mean based radial basis function neural network (RBFNN) is also proposed here for classification of hand gestures from LCS based feature set. The experiment is conducted on 500 train images and 500 test images of 25 class grayscale static hand gesture image dataset of Danish/international sign language hand alphabet. The proposed method performs with 99.6% classification accuracy which is better than earlier reported technique.**

***Index Terms*—Localized Contour Sequence (LCS), Morphological filter, Multiple Layer Perceptron Back Propagation Neural Network (MLPBPNN), Radial Basis Function Neural Network (RBFNN), Sign Language.**

I. Introduction

In the present day scenario of intelligent computing, an efficient human-computer interaction (HCI), human alternative and augmentative communication (HAAC) are assuming supreme importance in our daily lives. Proper design of gesture recognition algorithm have focused in developing advanced hand gesture interfaces resulting in successful applications like robotics, assistive systems, sign language communication and virtual reality [1]. Generally, gestures can be classified into static gestures [2-4], and dynamic gestures [5-7]. Static gestures are usually described in terms of hand shapes or poses, and dynamic gestures are generally described according to hand movements. However, static gestures can convey certain meanings and some time act as specific transitions state in dynamic gestures. Therefore, the static gestures recognition is one of the important topics for gesture recognition researches. For HCI and other application, the sensing techniques which have used in static hand gesture recognition algorithm have been divided mostly into vision-based techniques [1-11] and glove-based techniques [12-13]. The glove-based techniques utilize sensors to measure the joint angles, the positions of the fingers and the position of the hand in real-time [13]. However, gloves tend to be quite expensive and the weights of the glove and the cables of the associated measuring equipment hamper free movement of the hand. Vision-based techniques have been used one or more cameras to capture the images where gestures are performed by user. The general approach to vision based gesture recognition system can be based on 2D models like the image contour and the shape. A large number of literatures [2-4] are reported by the researchers for recognition of static hand gestures. Few of them [2], [4] used scale and rotation invariant feature set. However, development of significant feature set and accurate classification techniques are still being challenging task for static hand gesture recognition system. In a recent work [14], authors reported a static hand gesture recognition technique based on localized contour sequence(LCS) feature set and classification techniques via linear and non-linear alignment. Since, in real time experiment, the images are captured by camera from different positions and angle, the extracted feature set which are input to the classifier is to be rotation invariant. Therefore, if the numbers of test images or classes are very large in real time situation, the methodology adopted in [14] may not be suitable for static hand gesture recognition system.

Classifier plays an important role in the hand gesture recognition system. A variety of methods [2], [10], [14] have been reported for the classification of hand gesture. However neural network (NN)[15] is widely used as a classifier because (i) it can be used to generate likelihood-like scores that are discriminative in the state level; (ii) it can be easily implemented in hardware platform for its simple structure; (iii) it has the ability to approximate functions and automatic similarity based generalization property; (iv) complex class distributed features can be easily mapped by NN.

In this paper, a k-mean based radial basis function neural network (RBFNN) classifier is proposed for classification of static hand gesture images. Radial basis function neural network (RBFNN)[21] is most popular neural network(NN) because (i) its architecture is very simple, only one hidden layer consist between input and output layer; (ii) in hidden layer localised radial basis function are used to nonlinear transform of feature vector from input space to hidden space; (iii) sensitivity of the hidden neuron is tuned by adjusting spread factor of basis function; (iv) this network is faster and free from local minima problem etc. In this work, the accurate centres of the RBFNN are choosed using k-mean clustering algorithm and the direction of 1^{st} principal component of the segmented hand gesture is used to make it rotation invariant. The localized contour sequence (LCS) which is rotation, scale and position invariant, is used as a feature set of the hand gesture for accurate classification. The proposed algorithm recognizes 25 static hand gesture images of the bare hands with 99.60 percent accuracy which is better than earlier reported multilayer perceptron backpropagation neural network (MLPBPNN) based technique [4].

The rest of the paper is organized as follows: Section II describes the details of the proposed gesture recognition algorithm. Experimental results are discussed in Section III and Section IV concludes the paper.

II. Methodology

The proposed static hand gesture recognition algorithm consists of three following stage: preprocessing, feature extraction and classification. The flowchart of the proposed algorithm is given in Fig. 1

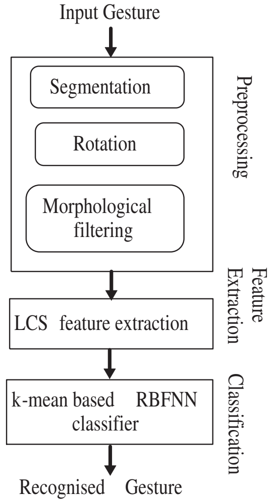

Fig. 1: Operational flowchart of proposed static hand gesture recognition algorithm

A. Preprocessing

Preprocessing of input gesture has been done through three operations: segmentation, rotation and morphological filter.

1) Segmentation: The objective of gesture segmentation is to extract the gesture region from the background of the image. The Otsu segmentation algorithm [16] is applied in this work. The segmentation results are shown in Fig. 2. The algorithm treats the segmentation of a gray scale image as a binary classification problem. Using a threshold T , the L gray levels image is segmented in two classes $\Omega_0 = \{1, 2, \dots, T\}$; and $\Omega_1 = \{T+1, T+2, \dots, L\}$. The optimum threshold T^* is determined as that value of T for which maximizes the ratio between-class variance σ_B^2 to the total variance σ_T^2. If the number of pixels at i^{th} gray level is n_i and the total number of pixels is N, then, for a given T , the between class variance and the total variance are defined and computed as follows:

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2$$
$$\sigma_T^2 = \sum_{i=1}^{L} (i - \mu_T)^2 P_i$$

where

$$\omega_0 = \sum_{i=1}^{T} P_i, \omega_1 = \sum_{i=T+1}^{L} P_i$$
$$\mu_0 = \sum_{i=1}^{T} (iP_i)/\omega_0, \mu_1 = \sum_{i=T+1}^{L} (iP_i)/\omega_1$$
$$\mu_T = \sum_{i=1}^{L} (iP)$$
$$P_i = n_i/N, \left(P_i \geq 0 \text{and} \sum_{\mathrm{i}=1}^{\mathrm{L}} P_i = 1\right)$$

Typically, the entire histogram is scanned to find the optimum threshold T.

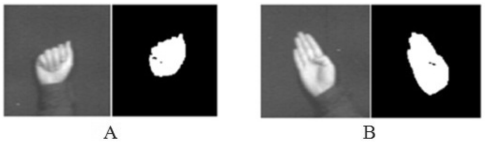

Fig. 2: The original and segmented hand gesture of‘A’and ‘B’

2) Rotation: In this block,segmented hand gesture is being rorated to make it rotation invariant.That is done by coinciding 1^{st} principal axes of the segmented hand gesture with vertical axes. The rotation is peformed by following steps:
Step 1: Find the direction of 1^{st} principal axes of segmented hand gesture.
Step 2: Find the rotation angle between the 1^{st} principal axes of the segmented hand gesture and vertical axes.
Step 3: Rotate the segmented hand gesture,so that 1^{st} principal axes of the segmented hand gesture coinside with vertical axes. Fig. 3 shows the result of segmented and rotated hand gestures.

3) Morphological filtering: A morphological filtering [17] approach was developed to obtain a smooth, closed, and complete contour of a gesture by using a sequence of dilation and erosion operations. In general, the dilation and erosion operations on a binary image P and with a structuring element

Fig. 3: Segmented and rotated hand gesture of‘A’and ‘B’

Q are defined as follows.

Dilation: If P and Q are sets in the 2-D integer space Z^2; $x = (x_1, x_2)$ and ϕ is the empty set, then, the dilation of P by Q is

$$P \oplus Q = \{x \mid (\hat{Q})x \cap P \neq \phi\}$$

where, $\hat{Q}$ is the reflection of Q. Dilation consists of obtaining the reflection of Q about its origin and then shifting this reflection by x. The dilation of P by Q is the set of all x displacements such that $\hat{Q}$ and P overlap by at least one nonzero element. Set Q is commonly referred to as the structuring element.

Erosion: The erosion of P by Q is

$$P \otimes Q = \{x \mid (Q)x \subseteq P\}$$

it indicates that the erosion of P by Q is the set of all points x such that Q, translated by x, is contained in P. Note that dilation expands an image and erosion shrinks it.

Opening: The opening of set P by structuring element Q is

$$P \circ Q = (P \otimes Q) \oplus Q$$

thus, the opening of P by Q is simply the erosion of P by Q followed by a dilation of the result by Q. Opening generally smoothes the contour of an image, breaks narrow isthmuses, and eliminates thin protrusions.

Closing: The closing of set P by structuring element Q is

$$P \bullet Q = (P \oplus Q) \otimes Q$$

this says that the closing of P by Q is simply the dilation of P by Q followed by the erosion of the result by Q. Closing also tends to smooth sections of contour but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour. Fig. 4 shows the result of morphological filtering.

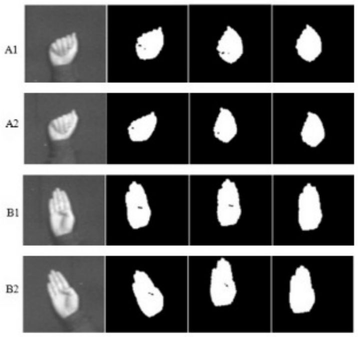

Fig. 4: Original, Segmented and Rotated and Filtered static hand gesture of‘A’and ‘B’

B. Feature Extraction

The shape of the contour is an important property that can be used to distinguish of the static hand gestures from one class to another. The localized contour sequence (LCS), which has been confirmed to be a very efficient representation of contours [18], is selected as a feature set of the hand gesture. A well established canny edge detector [19] used to detect the edge of preprocessed hand gesture. Detected edge of the preprocessed gestures are shown in Fig. 6. A contour tracking algorithm is proposed to track the contour of a gesture in the clockwise direction and the contour pixels are numbered sequentially starting from the topmost left contour pixel. The steps of the proposed contour tracking algorithm are given bellow:

Step 1: Scan the image from top to bottom and left to right to find first contour pixel marked as $P_1(i,j)$. Then record position of P_1 into x and y arrays, respectively, i.e., $x[1] = i$ and $y[1] = j$ and set $i_1 = 0, i_2 = 0, j_1 = 0, j_2 = 0$ and search next contour pixel.

Step 2: To scan clockwise direction searching P_J sequences are $P_J(i, j-1), P_J(i-1, j-1), P_J(i-1, j), P_J(i-1, j+1), P_J(i, j+1), P_J(i+1, j+1), P_J(i+1, j)$ and $P_J(i+1, j-1)$ respectively. where $J = 2, 3, 4, \ldots, N$. and N is total number of detected contour pixel.

Step 3: Scan clockwise until the next pixel value $P_J = 1$.

Step 4: If $P_J = 1$, position of $P_J \neq (i_1, j_1)$ and position of $P_J \neq (i_2, j_2)$, Store position of P_J into x and y arrays and set (i, j) = position of P_J . When $J > 3$ set (i_1, j_1) = position of P_{J-1} and (i_2, j_2) = position of P_{J-2}.

Step 5: If step 4 become false then set $x[J] = x[J-1], y[J] = y[J-1], (i, j) =$ position of $P_{J-1}, (i_1, j_1) =$ position of P_J and $(i_2, j_2) =$ position of P_{J-2}.

Step 6: Repeat steps 2 to 5 until position of $P_J =$ position of P_1.

By computing proposed contour tracking algorithm, the position of all contour pixel is stored into x and y arrays. If $h_i = (x_i, y_i), i = 1, 2 \ldots, N$ is the i_{th} contour pixel in the sequence of N ordered contour pixels of a gesture. The i_{th} sample $h(i)$ of the LCS of the gesture is obtained by computing the perpendicular Euclidean distance between h_i and the chord connecting the end-points $h_{[i-(w-1)/2]}$ and $h_{[i-(w-1)/2]}$ of a window of size w boundary pixels (w odd) centered on h_i. That is

$$h\left(i\right) = \left|u_i / v_i\right|,$$

where

$$\begin{aligned} u_i = x_i&[y_{i-(w-1)/2} - y_{i+(w-1)/2}] \\ &+ y_i[x_{i+(w-1)/2} - x_{i-(w-1)/2}] \\ &+ [y_{i+(w-1)/2}][x_{i-(w-1)/2}] \\ &- [y_{i-(w-1)/2}][x_{i+(w-1)/2}], \end{aligned}$$

and

$$\begin{aligned} v_i = [&(y_{i-(w-1)/2} - y_{i+(w-1)/2})^2 \\ &+ (x_{i-(w-1)/2} - x_{i+(w-1)/2})^2]^{1/2} \end{aligned}$$

The computation of $h(i)$ is illustrated in Fig. 5.

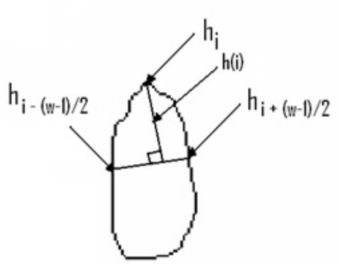

Fig. 5: Computation of the samples of the LCS.

The LCS [14] has the following important properties:
a) The LCS is not limited by shape complexity and it is, therefore, appropriate for gestures which typically have convex and concave contours.
b) The LCS has been used to robustly represent partial contours. Therefore, the representation of the visible part of the gesture will not be affected for a part of the gesture is obscured.
c) No derivative operation was involved during computations; therefore the representation is quite robust with respect to contour noise (random variations in the contour).
d) The amplitudes of the samples of the localized contour sequence proportionally increasing with w. An increase in the amplitudes has enhanced the signal-to-noise ratio for a fixed contour noise level.
LCSs were normalized by setting duration equal to 150 and standard deviation equal to unity. Fig. 6 shows the LCSs of A and B using $w = 45$.

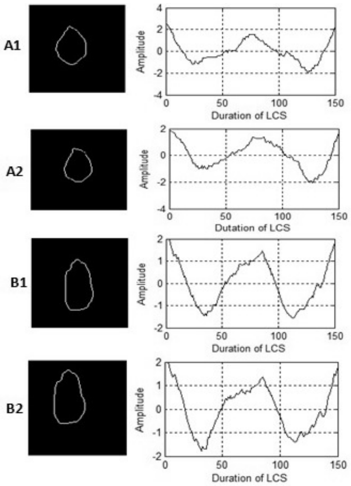

Fig. 6: Detected edge and normalized LCSs of gesture ‘A’ and ‘B’.

C. Classification

Having computed all the above stages, successfully extracted a normalized LCS feature vector of the static hand gesture. The classification job is done via k-mean based radial basis function neural network (RBFNN). Radial basis function neural network (RBFNN) is widely used in pattern recognition tasks for its fast learning algorithms. The centers and spread factor of the radial basis function are important parameter of RBFNN. Several methods have been used to find the centers of the RBFNN. In this paper, we proposed a k-means clustering based approach to determine the centers of the RBFNN. Our proposed approach is as follow:
Step 1: Set value of $m =$ total number of training classes and $i = 1$.
Step 2: Take D_i ,the data of i_{th} class.
Step 3: Set value of $k =$ number of clusters in D_i.
Step 4: Assign initial centroid of each cluster by selecting randomly k number of samples from data D_i or first k number of samples from data D_i.
Step 5: Take each sample of data D_i and compute its euclidean distant from centroid of each of the cluster and assign to the cluster with the nearest centroid.
Step 6: Compute the centroid of each cluster.
Step 7: Repeat step 5 to step 6 until centroid of the each cluster don’t change.
Step 8: Store k number of centroid as centers of RBFNN in i_{th} iteration and increase i by 1.
Step 9: Repeat step 2 to step 8 until $i <= m$.

III. EXPERIMENT RESULTS AND DISCUSSION

The static hand gesture dataset [20] of Danish/international sign language hand alphabet shown in Fig. 7. It consists 1000 grayscale images of 25 gesture, 40 sample each class with spatial resolution (256×248) pixel. The dataset is equally split into training and testing set. Training dataset is used to train the network. However testing set is used for test the performance.

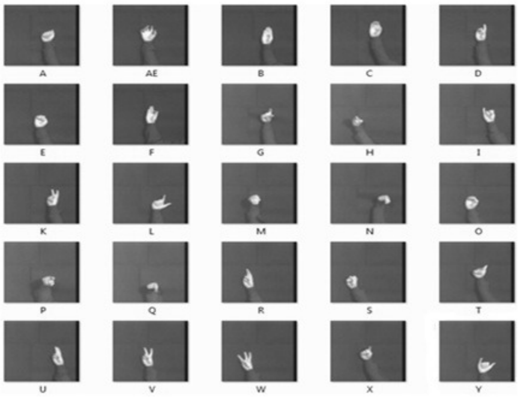

Fig. 7: Danish/international sign language hand alphabets

The performance of the gesture recognition algorithm is evaluated on the basis of its ability to correctly classify samples to their corresponding classes. The recognition rate can be defined as the ratio of the number of correctly classified samples to the total number of samples.

$$\text{Recognition rate} = \frac{\text{Number of corectly classified signs}}{\text{Total number of signs}} \times 100\%$$

For single hidden layer MLPBPNN [21] classifier, the parameters, (goal of mean square error (MSE) =0.0015, learning rate=0.5 and moment 0.9), were used to evaluate performance of gesture recognition algorithm using LCS feature. Variation of hand gesture recognition accuracy with numbers of hidden nodes is shown in Fig. 8. The figure shows the recognition efficiency reach into 99.4% when number of hidden node is 350.

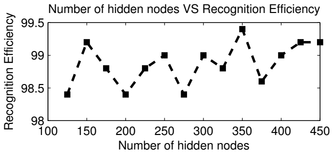

Fig. 8: The variation of recognition efficiency with respect to number of hidden nodes.

Performance evolution of proposed recognition algorithm is done via k-mean based RBFNN classifier. In this classifier 150 centre nodes were chosen by k-mean clustering algorithm. Fig. 9 shows the performance of gesture recognition for different spread factor. With k-mean based RBFNN classifier, the proposed technique achieves 99.6% recognition accuracy when spread factor is 5.5. The confusion matrix for test dataset using k-mean based RBFNN classifier with spread factor 5.5 is shown in Table 1.

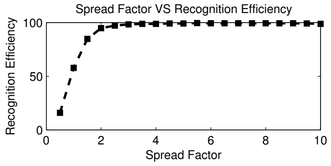

Fig. 9: The variation of recognition efficiency for different spread factors.

The analysis of confutation matrix conformation reveals the results where gestures are sources of error. For example from the Table 1 we have seen that one class D and class E gesture have been misclassified and shifted to R and S class respectively.

IV. Conclusion

A novel static hand gesture recognition algorithm which overcomes the challenges (such as rotation, size and position variation of the images) for detection of hand gesture images, is developed in this work. To obtain a rotation invariant gesture image, this work proposes a technique that coincides the 1^{st} principal component of the segmented hand gestures with vertical axes. The localized contour sequence feature which is position invarient, is normalised to overcome size variation of the hand gesture images. In this work, K-mean based RBF neural network is also proposed as a classifier for recognition of hand alphabets from static hand gesture images. The proposed k-mean based RBF neural network yields 99.6% accuracy for classification of 500 gesture image database and shows better performance compared to MLP-BP Neural Network as reported in earlier research work.

References

[1] Ong and Ranganath, “Automatic Sign Language Analysis: A Survey and the Future Beyond Lexical Meaning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 6, pp. 873-891, June 2005.

[2] Priyal S.P.,and Bora P.K., “A study on static hand gesture recognition using moments”,International Conference on Signal Processing and Communications(SPCOM), pp. 1-5, July 2010.

[3] J. Triesch and C. von der Malsburg, “A System for Person Independent Hand Posture Recognition against Complex Backgrounds,”IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 12, pp. 1449-1453, Dec. 2001.

[4] P. Premaratne and Q. Nguyen, “Consumer electronics control system based on hand gesture moment invariants,” IET Comput. Vis., Vol. 1, No. 1, March 2007.

[5] K. Wan and H. Sawada, “Dynamic gesture recognition based on the probabilistic distribution of arm trajectory”,In Proceedings of 2008 IEEE International Conference on Mechatronics and Automation, pp. 426-431,2008.

[6] R. Yang, S. Sarkar,and B. Loeding, “Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language RecognitionUsing Nested Dynamic Programming,” IEEE Trans. Pattern Anal. Machine Intell., vol. 32, no. 3, Mar. 2010.

[7] J. Lee and T. L. Kunii, “Model-based analysis of hand posture,” IEEE Compute. Graph. Appl., pp. 77-86, Sept. 1995.

[8] B. Moghaddam and A. Pentland, “Probabilistic visual learning for object recognition,”IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp. 696-710, July 1997.

[9] T. Staner, J. Weaver, and A. Pentland, “Real-time American sign language recognition using desk and wearable computer based video,”IEEE Trans. Pattern Anal. Machine Intell., vol. 20, pp. 1371-1375, Dec. 1998.

[10] Y. Cui and J. Weng, “A learning-based prediction-and-verification segmentation scheme for hand sign image sequence,” IEEE Trans. Pattern Anal. Machine Intell., vol. 21, pp. 798-804, Aug. 1999.

[11] V. I. Pavlovic, R. Sharma, and T. S. Huang, “Visual interpretation of hand gestures for human computer interaction: A review,”IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp. 677-694, July 1997.

[12] D. J. Sturman and D. Zeltzer, “A survey of glove-based input,” IEEE Comput. Graph. Appl., vol. 14, pp. 30-39, Jan. 1994.

[13] Wang, C., and Cannon, D.J., “A virtual end-effector pointing system in point-and-direct robotics for inspection of surface flaws using a neural network-based skeleton transform.”, Proc. IEEE Int. Conf. Robot. Automation, vol. 3, pp. 784-789. 1993.

[14] Lalit Gupta and Suwei Ma “Gesture-Based Interaction and Communication:Automated Classification of Hand Gesture Contours,” IEEE Trans. Syst., Man, Cybern. Part.C:App. Rev., vol. 31, no. 1, Feb. 2001.

[15] S. Ari and G. Saha, “In search of an Optimization Technique for Artificial Neural Network to Classify Abnormal Heart Sounds”, Elsevier Applied Soft Computing Journal, vol. 9, 2008, pages 330-340.

[16] N. Otsu, “A threshold selection method from gray-level histogram,” IEEE Trans. Syst., Man, Cybern., vol. SMC-9, pp. 62-66, Jan. 1979.

[17] E. R. Dougherty, An Introduction to Morphological Image Processing. Bellingham, WA: SPIE, 1992.

[18] L. Gupta, T. Sortrakul, A. Charles, and P. Kisatsky, “Robust automatic target recognition using a localized boundary representation,” Pattern Recognit., vol. 28-10, pp. 1587-1598, 1995.

[19] Canny, J., “A Computational Approach to Edge Detection”, IEEE Trans. Pattern Analysis and Machine Intelligence., 8(6):679-698, 1986.

[20] www-prima.inrialpes.fr/FGnet/data/12-MoeslundGesture/database.html

[21] Haykin, S.: ‘Neural networks’ (Prentice-Hall, 1999, 2nd edn.).

TABLE I: Confusion matrix for RBFNN classifier

	A	AE	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
A	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AE	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
E	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
F	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20