Sādhanā Vol. 27, Part 5, October 2002, pp. 585–594. © Printed in India

Two-tier architecture for unconstrained handwritten character recognition

K V PREMA* and N V SUBBA REDDY

Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal 576 119, India
e-mail:{prema_kv, dr_nvsreddy}@rediffmail.com

Abstract. In this paper, we propose an approach that combines the unsupervised and supervised learning techniques for unconstrained handwritten numeral recognition. This approach uses the Kohonen self-organizing neural network for data classification in the first stage and the learning vector quantization (LVQ) model in the second stage to improve classification accuracy. The combined architecture performs better than the Kohonen self-organizing map alone. In the proposed approach, the collection of centroids at different phases of training plays a vital role in the performance of the recognition system. Four experiments have been conducted and experimental results show that the collection of centroids in the middle of the training gives high performance in terms of speed and accuracy. The systems developed also resolve the confusion between handwritten numerals.

Keywords. Feature extraction; self-organizing map; learning vector quantization; handwritten numeral recognition; substitution error; classification.

1. Introduction

Since the late 1960's, research on recognition of unconstrained handwritten characters has made impressive progress and many systems have been developed, particularly in machine-printed and on-line character recognition (Rocha & Pavlidis 1994; Elms 1996; Hu *et al* 1996; Lee 1996; Cho 1997). However, there is still a significant performance gap between humans and machines in the recognition of off-line totally unconstrained handwritten character recognition.

Character extraction and recognition techniques have potential application in any domain where a large mass of document image-bearing texts must be interpreted or analysed. Conventionally, such images are processed by human operators who act according to what has been written or simply key in what they read onto a computer system that carries out further processing, say of postal address. However, automation of the entire process requires a high recognition rate, as well as maximum reliability. The dead letter problem in postal services arises due to the conflict in the identification of the PIN or ZIP code. This problem can be

*For correspondence

avoided by resolving the confusion between unconstrained characters of the code. When a pattern is incorrectly recognized or for any pattern the system assigns the same confidence values to two different classes, a substitution error is said to occur. For example, handwritten 7 is recognized as 1. In other words, the system assigns the same confidence values to both the classes 1 and 7. Conversely, as the substitution error increases, reliability of the system goes down. As substitution rate decreases, reliability increases (reliability is 100% when substitution rate is 0).

Interest in neural networks is rapidly growing and several neural network models have been proposed for various difficult problems, especially classification problems. Traditional classifiers test the competing hypothesis sequentially, whereas neural network classifiers test the competing hypothesis in parallel, thus providing high computational rates. There are two approaches to neural networks based on (i) supervised learning principles, and (ii) unsupervised learning principles. In order to take advantage of both these methods and achieve almost 100 percent classification, we have proposed a method which combines both these principles. Our approach employs the modified Kohonen self-organizing map (MSOM) and the learning vector quantization (LVQ) methods.

Broadly, off-line handwritten character recognition system includes three stages: image preprocessing, feature extractor, and classifier. Preprocessing is primarily used to reduce variations of handwritten characters. A feature extractor is essential for efficient data representation and extracting meaningful features for later processing. A classifier assigns the characters to one of the several classes.

As a major factor influencing recognition performance, features play a very important role in handwriting recognition. This has led to the development of a variety of features for handwritten recognition and their recognition performances have been reported on standard databases (Trier *et al* 1996; Suen 1998). Some recent papers include those proposing directional distance features (Oh & Suen 1998), gradient-based features (Lee *et al* 1996), pixel distance features (Strathy & Suen 1995), and concavity features (Favata *et al* 1994).

Features do not necessarily convey any intuitive meaning to a human and the dimensionality of the feature vector is very high, in hundreds, so it is difficult to understand their discriminative characteristics. A systematic evaluation of features in a specific feature vector is very important for designing a new feature vector by combining different features.

2. Feature extractor

In recent years, different methods and techniques for the recognition of handwritten English, Chinese, Japanese, and Arabic letters have been developed (Tappert *et al* 1990). The conventional method is to consider a region which includes the letter. If a line passes through a pixel, we give the corresponding pixel a value of 1, otherwise it is taken as 0. Thus we have to store the pixel value combinations for every letter. Now when an input pattern is given, a suitable match with these stored patterns is checked. Only if the new input pattern exactly matches with any of the stored patterns, it will be identified. In case of the English alphabet, there are different fonts for every character. The same letter written by different people and even by the same person at different times will be different. Hence, a generalized condition for recognizing a particular character cannot be specified, which is a must for the algorithmic approach. This is the major drawback of the conventional method – the inability to generalize. Another drawback is the large memory requirement for storing the pixel values.

Artificial neural methods are used for achieving recognition because rather than programming them, we train neural networks by examples. Programmers need not give neural networks the qualitative description of objects being recognized and sets of logical criteria to distinguish such objects from similar objects. Instead, we give examples of objects with their identification. The network memorizes this information by modifying the values in its weight matrix and will produce correct response when the object is seen again (Nielson 1990). This learning ability of neural networks has made it very appropriate for the present problem.

If we use a neural network to recognize the characters from their pixel value combinations, we get the needed generalization. But there are a large number of input units and hidden units and, accordingly, the weighted connections also increase. This results in a very complex network. One way to reduce the complexity of the network as well as the storage requirements is the bar mask encoding method (Burr 1997). The bar mask used in the experiment is similar to the seven segment alpha numeric display used in the familiar digital or electronic watches. If we have to develop a system for recognising hand-printed numbers, a bar mask array resembling this display can be considered as the encoder. But this seven-segment encoder performs poorly on alpha-numeric characters, since vertical symbols such as 'I' and 'T' and cross symbols such as 'N' and 'X' remain undetected. This is remedied by adding additional bar sensors (David *et al* 1991).

Figure 1 shows a fifteen segment encoder. It consists of three horizontal bars (HF1, HF2, HF3), three vertical bars (VF1, VF2, VF3), two central bars (CF1, CF2), and seven diagonal bars (DF1, DF2, DF3, Df4, DF5, DF6, DF7). The diagonal bars DF1, DF2, . . . , DF7 take care of cross strokes as in the letters 'N', 'X', 'W' etc . The capture region associated with DF5 helps in distinguishing letter 'E' from 'F', '6' from 'C' etc. DF4 and DF6 help in distinguishing '4' from '9'. The vertical bars VF1 to VF3 take care of vertical strokes as

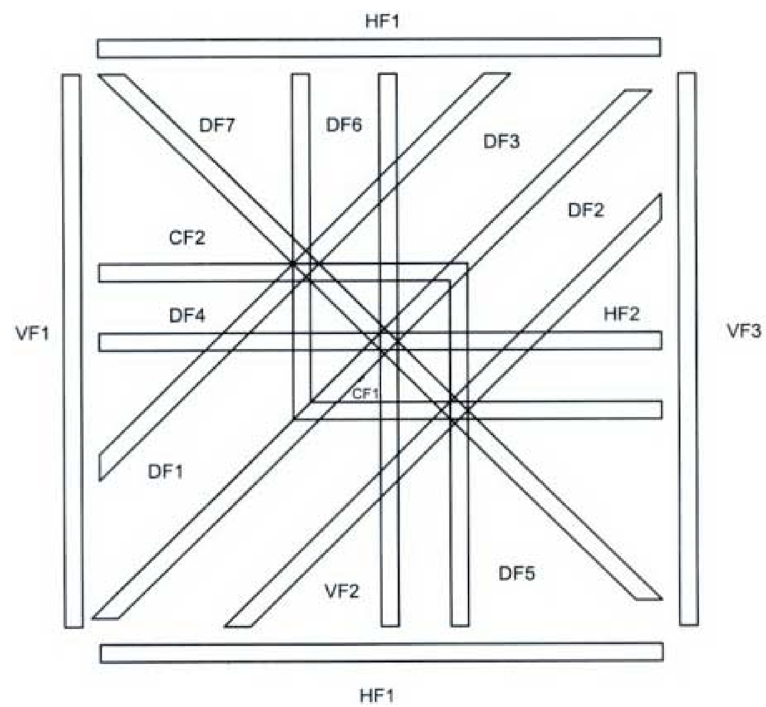


Figure 1. Fifteen segment encoder.

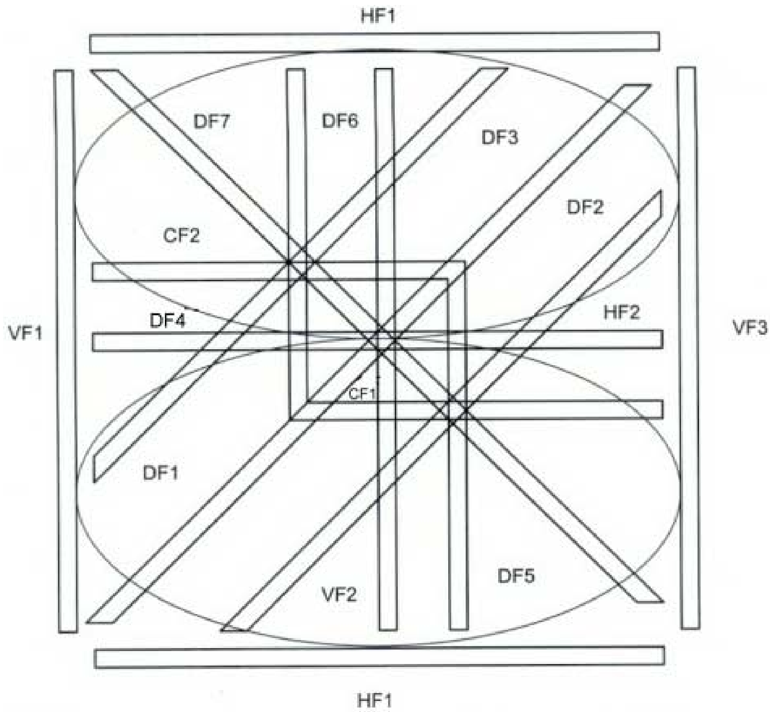
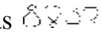


Figure 2. Fifteen segment encoder on which sample ‘8’ is mapped.

in the letters ‘I’ and ‘T’. The central bars CF1 and CF2 help in distinguishing character ‘3’ from ‘8’, ‘8’ from ‘0’ etc. They also take care of displaced centre features such as  etc. Thus using this encoder it is possible to extract 15 features. This encoder converts the input character into a highly compressed format suitable for recognition. The character to be encoded is first standardized by scaling it so that it extends to the full height and width of the 15-segment encoder region. Figure 2 shows the fifteen segment feature encoder on which sample ‘8’ is mapped. Each cell in the character is assigned a value of 1 or 0 depending on whether it contains a portion of the character or not. For example: The ith row and jth column cell Cij is assigned a value $Vcij$, where $Vcij = 1/(i \times 10 + j)$.

Associated with each segment is a capture region. The word ‘capture region’ is used to denote that the segment captures the values of all cells within its specified region. However the feature value F, associated with each region is given as

$F =$ sum of values of ON cells/sum of values of all cells.

Thus the feature values $F1, F2, \ldots, F15$ are evaluated and stored in a feature array.

Algorithm:

1. Read the pattern.
2. For each cell $Cij = [i, j]$ of the pattern assign a unique value $Vcij$, where $Vcij = 1/(i \times 10 + j)$.
3. For each region
   Feature value $F =$ sum of values of ‘ON’ cells/sum of values of all the cells.
4. Repeat this for all the patterns.

For example: Feature value HF1 is calculated as follows:

for ($i = 1$ to 5)

{ for ($j = 1$ to 15)

{ $F[i][j] = 1/(i * 10 + j)$

}

} Numerator = Denominator = 0

Read the pixel matrix $P[i][j]$ containing the pattern.

for ($i = 1$ to 5)

{ for ($j = 1$ to 15)

{ if ($P[i][j] == 1$)

Numerator = Numerator + $F[i][j]$

Denominator = Denominator + $F[i][j]$

}

}

HF1 = Numerator/Denominator.

The extracted features are used for training the Kohonen self-organizing and the learning vector quantization models.

3. Two-tier architecture

Kohonen's self-organizing map uses unsupervised learning to modify the internal state of the network and to model the features found in the training data set. The map is automatically organized by a cyclic process of comparing the input patterns to the vectors at each node. The node vector with which the input matches is selectively optimized to represent an average of the training data. Thus starting with a randomly organized set of nodes, the proposed method proceeds to the creation of a feature map representing the prototypes of the patterns.

In the conventional SOM the misclassification is more (Kohonen 1990) because the minimum distance formula is applied to both the standard (machine printed numerals) and the distorted sets of samples. This is overcome in MSOM by applying the minimum distance formula for only the standard set of samples. The sample is compared only with the nodes represented by the standard set of samples (which form the centroids for each class). Using the minimum distance formula and the winner node (that is the node which most matches the sample), the weights in the neighbourhood of this winner node are updated. The MSOM is organized by a cyclic process of comparing the input pattern to vectors at each node twice. The MSOM so obtained is further fine-tuned using the LVQ technique (Hush & Horne 1993).

Also in conventional SOM the misclassification is greater because of the collection of samples after training. This is overcome in M1SOM and M2SOM where we collect the centroids at the middle of training and at 3/4th training respectively. By means of the SOM algorithm described above, the numeral data are self-organized into a feature map. The nearest neighbour classifier can then be applied to perform the classification.

3.1 *Learning vector quantization*

Kohonen has suggested that when the nodes of the self-organizing map are used for pattern recognition, their classification accuracy can be multiplied if nodes are fine-tuned using the supervised learning principle (Kohonen 1990). The learning vector quantization uses supervised learning to modify the internal state of the network provided by MSOM and to remodel the features found in the training data. A fine tuned map is autonomously organized by a cyclic process of comparing the input pattern to the vectors at each node. Fine-tuning is achieved by selecting training vectors X with known classification, and presenting them to the network to examine the cases of misclassification. The best match comparison is performed at each node and the winner node is noted. The patterns are self-organized into a fine tuned feature map because of the LVQ algorithm.

4. Experiments and results

Experiments have been carried out to investigate the classification performance of the four different techniques. The sample database has been generated by taking the samples selected from various research papers of Concorde University, which were originally selected from the sample database of US postal services collected from various parts of the USA as seed values. The database consists of 10,000 samples each of size 64×64. Out of these 10,000 samples, 5,000 are used for training, and the remaining 5,000 are used for testing.

We have chosen the net size as 22×22 in which the initial neighbourhood size is 13 and the number of nodes in the input layer are 15. The learning rate at various stages during the training process is between 0.923971 and 0.048231.

The recognition rate, rejection rate and substitution error of the technique are estimated using the following decision rule:

1. For all samples collect the confidence levels from neural network.
2. For all samples do

If $R(x)$ is a recognized character for sample x and j be the character with highest confidence level, then if this maximum confidence level is greater than a threshold β and $j = x$, then increment rec. (which is the number of samples recognized), else find the possible substitution by checking the second, third and fourth confidence levels. If any of these is greater than β, increment rej. (number of samples rejected).

Recognition rate (REC) = (number of samples recognized/total number of samples) $\times$ 100.
Substitution rate (SUB) = (number of samples substituted /total number of samples) $\times$ 100.
Rejection rate (REJ) = (number of samples rejected/total number of samples) $\times$ 100.
Reliability (REL) = (Recognition/(Recognition + Substitution)) $\times$ 100.

The results of the four experiments are shown in table 1. The classification accuracy corresponds to the trained samples. The recognition rate is the percentage of input samples recognized correctly. The rejection rate is the percentage of input samples that could not be assigned to any particular class.

The following can be inferred from table 1.

Table 1. Results of four techniques for 5000 unconstrained handwritten numerals.

Techniques	REC (%)	SUB (%)	REJ (%)	REL (%)	Training time(s)	Classification (%)
M1SOM	92.00	00.00	08.00	100.00	55	92.00
M2SOM	87.00	00.00	13.00	100.00	64	87.00
M1SOM+LVQ	99.60	00.00	00.40	100.00	470	100.00
M2SOM+LVQ	98.40	00.00	01.60	100.00	4232	100.00

(1) The M1SOM method in which centroids were collected at the middle of training resulted in a recognition rate of 92% with 08.00% rejection and a classification accuracy of 92%. The training time was 55 s. The performance graph is shown in Figure 3.
(2) The M2SOM method in which the centroids were collected after 3/4th training resulted in a recognition rate of 87% with a rejection rate of 13% and a classification accuracy of 87%. The training time increased to 64 s. The performance graph is shown in Figure 4.

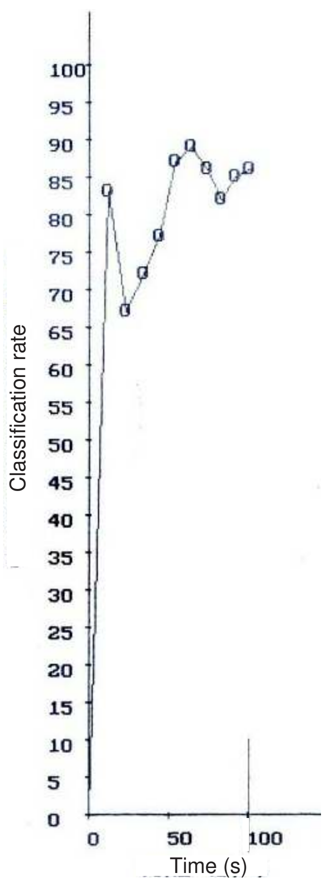


Figure 3. Performance curve for M1SOM technique.

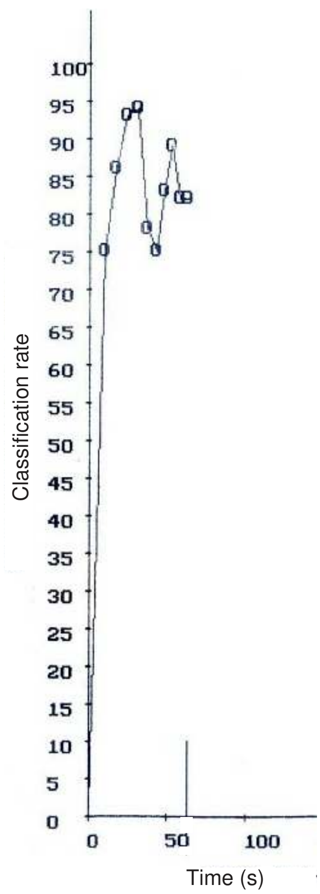


Figure 4. Performance curve for M2SOM technique.

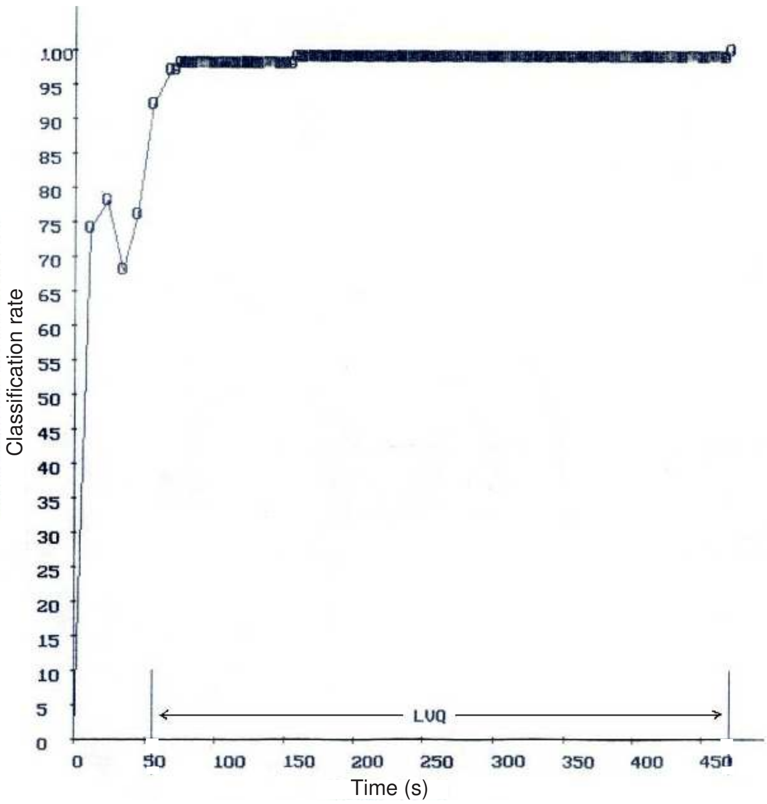


Figure 5. Performance curve for the M1SOM + LVQ technique.

(3) When the LVQ method was combined with M1SOM, the recognition rate increased to 99.6% with a classification accuracy of 100% and training time 470 s. The performance graph is shown in Figure 5.
(4) Combining M2SOM and LVQ resulted in a recognition rate of 98.4% and increase in the training time to 4232 s, for the same classification accuracy of 100%. The performance graph is shown in Figure 6.

It is also observed in the above experiments that the confusion between the numerals has been eliminated and hence the performance of the overall system is better than the combined method proposed by Cai & Liu (1999).

5. Conclusions

A new approach for unconstrained handwritten numeral recognition has been proposed. This approach is able to combine unsupervised and supervised learning methods, and achieve best overall performance. The experimental results confirm that the proposed method results in high performance in terms of recognition rate and classification accuracy, at the same time completely eliminating the substitution error. Hence the developed two-tier architecture is robust in the recognition of unconstrained handwritten characters.

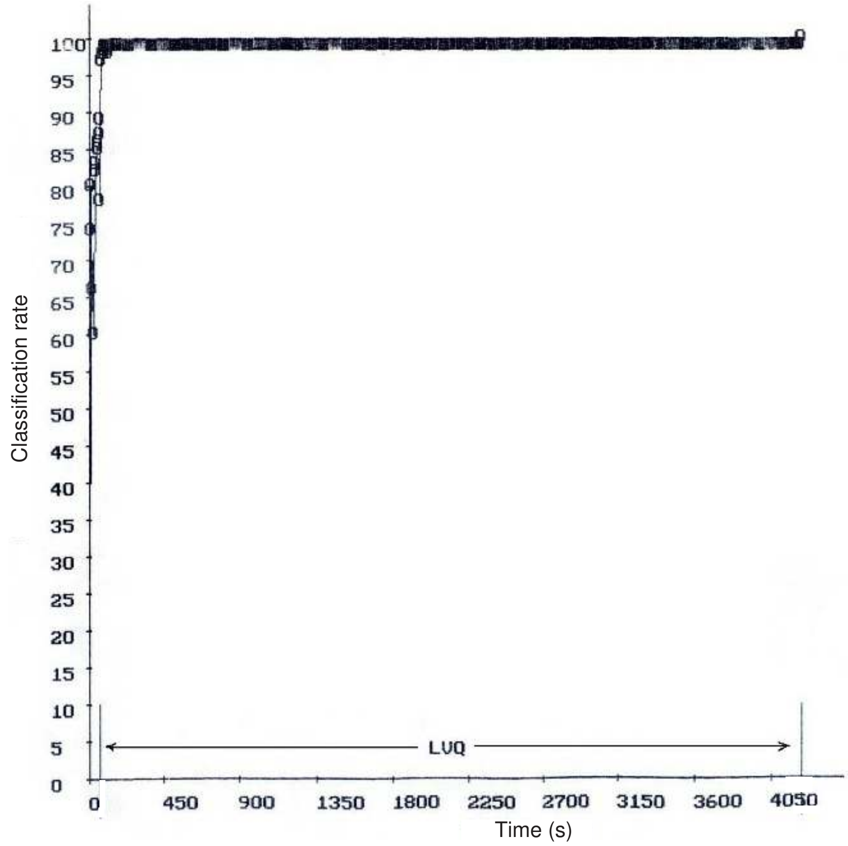


Figure 6. Performance curve for M2SOM + LVQ technique.

References

Burr D J 1987 Experiment with a connectionist text reader. *IEEE First Conf. on Neural Networks* 4: 717–724

Cai J, Liu Z-Q 1999 Integration of structural and statistical information for unconstrained handwritten numeral recognition *IEEE Trans. Pattern Anal. Mach. Intell.* 21: 263–267

Cho S B 1997 Neural network classifiers for recognizing totally unconstrained handwritten numerals. *IEEE Trans. Neural Networks* 8: 43–53

David P S, Santhya P, Shyne T P 1991 Handwritten character recognition - An experience. *Comput. Sci. Inf.* 21:

Elms A J 1996 *The representation and recognition of text using hidden Markov models.* Ph D thesis, Dept. of Electronics and Electrical Engineering, University of Surrey, Guildford, Surrey, UK

Favata J T, Srikantan G, Srihari S N 1994 Hand printed character/digit recognition using a multiple feature/resolution philosophy. *Proc. Int. Workshop on Frontiers of Handwriting Recognition, 1994 (IWFHR'94)*, pp. 57–64

Hu J, Brown M K, Turin W 1996 HMM based on-line handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 18: 1039–1045

Hush D R, Horne B 1993 Progress in supervised neural networks – What's new since Lippman. *IEEE Signal Process. Mag.* : 8–39

Kohonen T 1990 The self-organizing map. *Proc. IEEE* 78: 111–123

Lee S W 1996 Off-line recognition of rotally unconstrained handwritten numerals using multi layer cluster neural network. *IEEE Trans. Pattern Anal. Mach. Intell.* 18: 648–652

Lee S W, Kim C H, Tange Y Y 1996 Multi resolution recognition of unconstrained handwritten numerals with wavelet transform and multi layer cluster neural network. *Pattern Recogn.* 29: 1953–1961

Oh I S, Suen C Y 1998 Distance feature for neural network-based recognition of handwritten characters. *Int. J. Document Anal. Recogn.* 1: 73–88

Nielson R H 1990 *Neuro-computing* (Boston, MA: Addison-Wesley)

Rocho J, Pavlidis T 1994 A shape analysis mode: With application to a character recognition system. *IEEE Trans. Pattern Anal. Mach. Intell.* 16: 393–404

Srikantan G, Lam S W, Srihari S N 1996 Gradient-based contour encoding for character recognition. *Pattern Recogn.* 29: 1147–1160

Strathy N W, Suen C Y 1995 A new system for reading handwritten ZIP codes. *Proc. ICDAR*, pp 74–77

Tappert C C, Suen C Y, Wakahare T 1990 The state of the art in on-line handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 12: 787–803

Trier O D, Jain A K, Taxt T 1996 Feature extraction methods for character recognition – A survey. *Pattern Recogn.* 29: 641–662