PERGAMON

Pattern Recognition 35 (2002) 2061–2071

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Neural networks with enhanced outlier rejection ability for off-line handwritten word recognition

Jinhui Liu[a], Paul Gader[b,*]

[a] *Verity, Inc., 894 Ross Drive, Sunnyvale, CA 94089, USA*
[b] *CISE Department, University of Florida, E301 CSE Building, Gainesville, FL 32611, USA*

Received 5 April 2001; accepted 17 September 2001

Abstract

For a segmentation and dynamic programming-based handwritten word recognition system, outlier rejection at the character level can improve word recognition performance because it reduces the chances that erroneous combinations of segments result in high word confidence values. We studied the multilayer perceptron (MLP) and a variant of radial basis function network (RBF) with the goal to use them as character level classifiers that have enhanced outlier rejection ability. The variant of the RBF uses principal component analysis (PCA) on the clusters defined by the nodes in the hidden layer. It was also trained with and without a regularization term that was aimed at minimizing the variances of the nodes in the hidden layer. Our experiments on handwritten word recognition showed: (1) In the case of MLPs, using more hidden nodes than that required for classification and including outliers in the training data can improve outlier rejection performance; (2) in the case of PCA–RBFs, training with the regularization term and no outlier can achieve performance very close to training with outliers. These results are both interesting. Result (1) is of interest because it is well known that minimizing the number of parameters, and therefore keeping the number of hidden units low, should increase the generalization capability. On the other hand, using more hidden units increases the chances of creating closed decision regions, as predicted by the theory in Gori and Scarselli (IEEE Trans. PAMI 20 (11) (1998) 1121). Result (2) is a strong statement in support of the use of regularization terms for the training of RBF-type neural networks in problems such as handwriting recognition for which outlier rejection is important. Additional tests on combining MLPs and PCA-RBF networks showed the potential to improve word recognition performance by exploiting the complementarity of these two kinds of neural networks.

Keywords: Handwriting recognition; Outlier rejection; Multilayer perceptron; RBF network; Principal component analysis; Regularization

1. Introduction

1.1. Handwritten word recognition

Handwritten word recognition has many practical applications, such as reading handwriting in pen-input devices, automated mail sorting, check reading, and form processing, etc. This very challenging pattern recognition problem has two cases, which are distinguished as off-line and on-line handwriting recognition. In the on-line case, the input is in the form of successive points of strokes collected in time order, possibly with additional information about pen-down, pen-up, or pen pressure, etc. during writing. In the off-line case, the kind we are concerned with here, the input is in the form of a digital image of handwritten word. Plamondon and Srihari [1] provided a recent survey of both on-line and off-line handwritten recognition.

Usually an off-line handwritten word recognition system takes two inputs: a word image and a list of strings called a lexicon, representing candidate identities for the

* Corresponding author. Tel.: +1-352-392-1526; fax: +1-352-392-1220.
E-mail address: pgader@cise.ufl.edu (P. Gader).

PII: S0031-3203(01)00191-1

word image. The recognition process assigns a match score to each candidate string and the highest score determines the recognition result. The major difficulty of handwritten word recognition is the wide variety of writing styles. Over the years there have been chiefly four different trends in this research area: (1) Segmentation and dynamic programming (DP)-based approaches, which split a word image into characters or partial characters and use character classifier and DP to obtain the optimal segmentation and recognition result [2–7]; (2) segmentation and hidden Markov model (HMM)-based approaches, which split a word image into characters or partial characters and generate the observation sequence accordingly for the HMMs, which produce the word recognition result [8–10]; (3) segmentation-free and HMM-based approaches, which usually generate the observation sequence based on moving a window over a word image for the HMMs, which produce the word recognition result [11–14]; (4) Holistic approaches, which do not attempt to segment a word image but treat it as a whole pattern and recognize it [15–18]. As we usually see in the pattern recognition area, there have also been blends of those different trends [12,17,19].

Some of the most promising results have come from segmentation and DP-based approaches. Our baseline system is one of them, which is illustrated in Fig. 1. A word image is segmented into sub-images called *primitives*, each of which should be the image of a single character or a partial character. A *union* is defined as either a primitive or the combined image of some neighboring primitives. The DP module uses character confidence values of unions (assigned by a neural network character classifier) and compatibility scores of pairs of neighboring unions (assigned by a neural network to account for the spatial relationships and relative sizes between neighboring unions) to group the primitives of the word image into a sequence of unions that best matches a given string. This system has been fully described in [5,12,20].

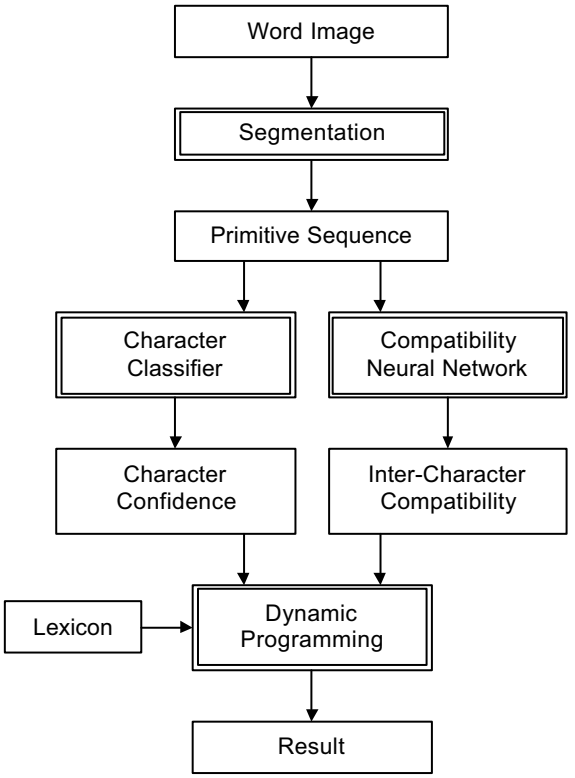


Fig. 1. Diagram of our baseline handwritten word recognition system.

1.2. Outlier rejection for handwritten word recognition

There are some efforts to enhance the performance of the segmentation and DP-based systems. Kim and Govindaraju [4] used the distributions of the numbers of segments (called "duration") into which different characters can be split by the segmentor to reward the character confidences. Kimura et al. [2] used a splitting cost to punish the bad cuts. Scagliola et al. [21] used a number of complementary sources of information, including the split and joint cost, duration cost, extra ink (stroke) processing, etc. Gader et al. [5] used the inter-character spatial compatibility, as illustrated in Fig. 1.

In our previous work [5–7], it was often observed that some errors were caused by non-character images that were assigned high character confidence values. Our focus here is to investigate assigning low character confidence values to non-character images to improve word recognition performance. To assign low character confidence values to non-character images means to reject outliers.

There has been some practical work on outlier rejection for character or word recognition using neural network classifiers. Chiang and Gader [6] uses self-organizing feature maps (SOFM) to form a uniform representation of non-characters. Chiang and Gader reported the use of SOFM to augment a multilayer perceptron (MLP) on its digit recognition reliability [22]. Kim et al. [23] used an additional node for outliers in the output layer of their MLP. Fogelman et al. [24] devised a hybrid network for digit recognition consisting of a shared weight network for feature extraction followed by a radial basis function (RBF) network and investigated its ability to reject white noise and alphabetic characters.

HMMs have been used to model classes of garbage, such as silence in speech and spaces in handwriting. However, modeling of garbage is difficult since the class of non-characters has no inherent structure, it consists of all patterns that are not character patterns and there is no inherent shape relationships between those shapes. In addition, modeling garbage is different than attempting to reject it by building better models of characters, which is the focus of the research presented here.

There has also been some theoretical work of interest. We know that what a classifier essentially does is to separate the pattern space into different regions, with each region mapping to a specific class. Here, we refer the bounding surface of each area as its corresponding separation surface. When outlier rejection is desired, it is important for a classifier to have closed separation surfaces to enclose the clusters in which valid patterns are distributed as close as possible to exclude outliers. Gori and Scarselli investigated the condition for MLPs to have closed separation surfaces [25]. They proved that MLPs with sigmoid activation functions and a number of hidden units (in the first layer) less than or equal to the number of inputs draw open separation surfaces in the pattern space. When using more hidden units than inputs, the separation surfaces can be closed, but there is no guarantee of it. The results of our previous study [26] confirmed their conclusions regarding MLPs. In that study, we investigated outlier rejection performance of MLP, RBF network and a variant of RBF network, which precedes an RBF with Principal Component decomposition, and is therefore called PCA-RBF network (to be explained in Section 2.2.1). Our experiments on two-dimensional artificial data showed:

1. Including the outlier samples in the training data and using more hidden nodes than that required for classification can improve the outlier rejection performance for the MLP and RBF networks.
2. The PCA-RBF network can achieve as good an outlier rejection performance as that of the MLP and RBF networks with the simplest structure (in terms of the number of hidden units).
3. Adding a regularization term in the training of a PCA-RBF network can achieve outlier rejection performance equivalent to that of other networks without using outliers in the training data.

In this paper, we describe an investigation into whether or not the same conclusions hold for MLP and PCA-RBF networks on real world handwriting data, where the feature spaces are high dimensional ($\geqslant 100$). The outlier rejection performance will be evaluated using our baseline system. The remainder of this paper is organized as follows: Section 2 describes the structures of the neural networks and the training algorithm for the PCA-RBF; Section 3 reports the experimental results and conclusions are drawn in Section 4.

2. Neural networks

The neural networks used here have class-coded output nodes, i.e., they have the same number of outputs as the number of character classes. With this kind of network structure, outlier rejection can be realized by having a valid pattern only activate the output node corresponding to the class which the pattern belongs to (output nodes of similar classes also have some low activation when the fuzzy desired output [27] is used), and having outliers not activate any output node. Neural networks with closed separation surfaces in the feature space are expected to have this kind of behavior.

2.1. MLP

MLPs demonstrate excellent performance in pattern classification tasks for which the input is known to be from a finite set of pattern classes. Unfortunately, they can also produce high outputs when a sample that is not from one of the classes is presented as input.

The MLPs used in this study have one input layer, one hidden layer, and one output layer. These networks are fully connected, use the logistic activation function, and are trained with standard back-propagation.

2.2. PCA-RBF network

2.2.1. The structure of the PCA-RBF network

RBF networks with localized basis functions have closed response domains in feature space, making them good candidates for rejecting outliers. The structure of an RBF network is illustrated in Fig. 2. It can be formulated as

$$o_k = \sigma(y_k), \tag{1}$$

$$y_k = \sum_{j=1}^{m} w_{kj}\Phi_j(\mathbf{X}) + b_k, \quad k = 1,\ldots,c, \tag{2}$$

where c is the number of outputs, m is the number of radial basis functions, $\sigma(\cdot)$ is the activation function at the output layer, $\mathbf{X}$ is an n-dimensional input feature vector, and usually $\Phi_j(\cdot)$ is a Gaussian form radial basis function with $\mathbf{U}_j$ as its center and Σ_j as its covariance matrix:

$$\Phi_j(\mathbf{X}) = \exp(-\tfrac{1}{2}(\mathbf{X}-\mathbf{U}_j)^{\mathrm{T}}\Sigma_j^{-1}(\mathbf{X}-\mathbf{U}_j)), \quad j = 1,\ldots,m, \tag{3}$$

where

$$\mathbf{U}_j = [u_{j1},\ldots,u_{ji},\ldots,u_{jn}]^{\mathrm{T}},$$

$$\Sigma_j = \mathrm{diag}\left(\frac{1}{\sigma_{j1}^2},\ldots,\frac{1}{\sigma_{ji}^2},\ldots,\frac{1}{\sigma_{jn}^2}\right).$$

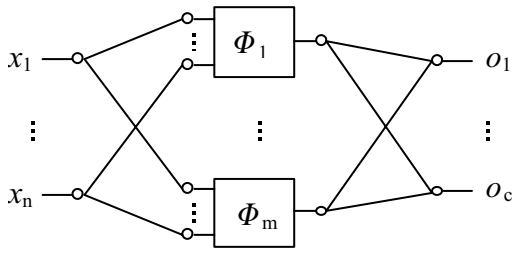
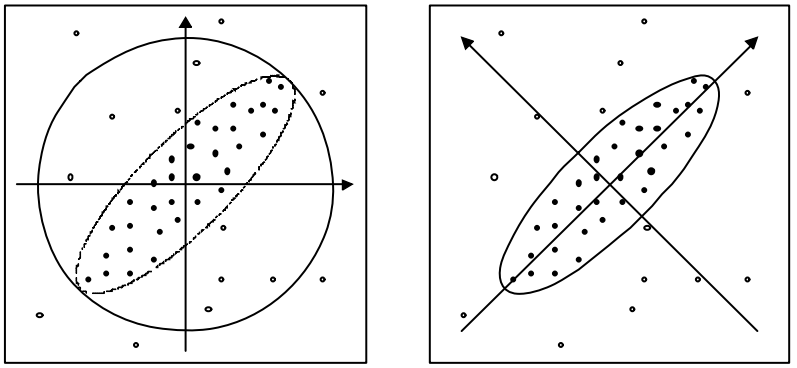


Fig. 2. The RBF network.



(a) (b)

Fig. 3. The relation between the axis direction and outlier rejection performance of a radial basis function (dots and circles represent the valid patterns and outliers, respectively): (a) the axes of the radial basis function are parallel to the axes of the feature space; (b) the axes of the radial basis function are parallel to the principal components of the cluster of the valid patterns.

Note Σ_j is a diagonal matrix, which is usually the case for RBF networks, then only n parameters need to be estimated instead of $n(n+1)/2$ parameters in the case of a full covariance matrix.

However, using diagonal covariance matrices in RBF networks constrains the axes of the hyper-ellipsoid of the basis functions to be parallel to the feature space axes. The weakness of these basis functions for outlier rejection is illustrated in Fig. 3(a). In this example, the valid patterns distribute in a cluster whose axes are not parallel to the axes of the pattern space. The smallest separation surface produced by a radial basis function using diagonal covariance matrix to enclose the cluster still includes some outliers.

To reject outliers better, it is desirable to make a radial basis function fit a cluster of patterns more closely. We conjecture that rotating the axes of the hyper-ellipsoid of the basis function helps fit a cluster of patterns more closely (Fig. 3(b)). This can be achieved by using a full covariance matrix in the basis function. However, covariance matrices must be positive-definite, which is difficult to enforce during training. Furthermore, the training problem is difficult because there are $O(n^2)$ rather than $O(n)$ parameters to learn for each covariance matrix.

We propose a variant of RBF network to solve the above problems. This new network is constructed using PCA on the clusters found in the training data, and is therefore called PCA-RBF network. After projecting an input from the original feature space into the new space spanned by the principal components of a cluster, a diagonal covariance matrix can be used in the radial basis function for that cluster. Therefore, the radial basis functions for a PCA-RBF network can be computed as

$$\begin{aligned} &\mathrm{P}_j(\mathbf{X}) = \exp(-\tfrac{1}{2}\mathbf{Z}_j^{\mathrm{T}}\Sigma_j^{-1}\mathbf{Z}_j),\\ &\mathbf{Z}_j = \Psi_j^{\mathrm{T}}(\mathbf{X}-\mathbf{U}_j), \quad j=1,\ldots,m, \end{aligned} \tag{4}$$

where $\mathbf{U}_j$ is the center of the jth cluster, Ψ_j is the $n \times n_j$ matrix with columns equal to the n_j normalized eigenvectors of the jth cluster corresponding to the n_j largest eigenvalues, $\mathbf{Z}_j$ is the projection of the input vector $\mathbf{X}$ (shifted with regard to the center $\mathbf{U}_j$) in the new space, and Σ_j is the diagonal covariance matrix for the jth cluster in the new space. The dimensionalities of Ψ_j, $\mathbf{Z}_j$ and Σ_j depend on the number of the principal components used. Supposing that n_j is the number of principal components used for the jth radial basis function, Ψ_j, $\mathbf{Z}_j$ and Σ_j can be expressed as

$$\Psi_j = [t^j_{i',j'}]_{n\times n_j},$$

$$\mathbf{Z}_j = [z_{j1},\ldots,z_{ji},\ldots,z_{jn_j}]^{\mathrm{T}},$$

$$\Sigma_j = \mathrm{diag}\left(\frac{1}{\sigma_{j1}^2},\ldots,\frac{1}{\sigma_{ji}^2},\ldots,\frac{1}{\sigma_{jn_j}^2}\right).$$

Then the PCA-RBF network can be formulated almost the same way as Eqs. (1) and (2) by just substituting $\mathrm{P}_j(\cdot)$ for $\Phi_j(\cdot)$ in Eq. (2). Because Σ_j is diagonal, now we can avoid adjusting full covariance matrices in the training.

2.2.2. The training algorithm without the regularization term

We derived a backpropagation-style training algorithm for the PCA-RBF network. Suppose that the lth training sample is $\mathbf{X}(l)=[x_1(l),\ldots,x_i(l),\ldots,x_n(l)]^{\mathrm{T}}$ and its desired outputs are $d_1(l),\ldots,d_k(l),\ldots,d_c(l)$. The error function is defined as

$$E(l) = \frac{1}{2}\sum_{k=1}^{c}(o_k(l)-d_k(l))^2. \tag{5}$$

Let

$$\delta_k(l) = \frac{\partial E(l)}{\partial y_k(l)} = \frac{\partial E(l)}{\partial o_k(l)} \frac{\partial o_k(l)}{\partial y_k(l)}$$

$$= (o_k(l) - d_k(l)) \frac{\partial o_k(l)}{\partial y_k(l)}, \quad k = 1, \ldots, c. \tag{6}$$

The following partial derivatives can be obtained:

$$\frac{\partial E(l)}{\partial b_k} = \delta_k(l), \qquad \frac{\partial E(l)}{\partial w_{kj}} = \delta_k(l) \mathrm{P}_j(\mathbf{X}(l)),$$

$$\frac{\partial E(l)}{\partial u_{ji}} = \left(\sum_{k=1}^{c} \delta_k(l) w_{kj} \right) \left(\sum_{k'=1}^{n_j} \mathrm{P}_j(\mathbf{X}(l)) \right.$$

$$\left. \times \left(-\frac{z_{jk'}(l)}{\sigma_{jk'}} \right) \left(\frac{1}{\sigma_{jk'}} \right) (-t^j_{i,k'}) \right),$$

$$\frac{\partial E(l)}{\partial \sigma_{ji}} = \left(\sum_{k=1}^{c} \delta_k(l) w_{kj} \right) \mathrm{P}_j(\mathbf{X}(l)) \left(-\frac{z_{ji}(l)}{\sigma_{ji}} \right) \left(-\frac{z_{ji}(l)}{\sigma^2_{ji}} \right).$$

During the training, b_k, w_{kj}, u_{ji} and σ_{ji} can be updated using the gradient descent technique. For example, b_k is updated using

$$\Delta b_k = -\eta \frac{\partial E(l)}{\partial b_k},$$

where η is the learning rate.

2.2.3. The training algorithm with the regularization term

For PCA-RBF networks the expansion parameter (σ_{ji}) of the radial basis functions seems to have a major role in outlier rejection. So, we tried to use a regularization term instead of the outlier data for training. The regularization term is a term added to the error function that aims to limit σ_{ji} during training. We modified the error function as

$$E(l) = \frac{1}{2} \sum_{k=1}^{c} (o_k(l) - d_k(l))^2 + \frac{1}{2} \frac{\lambda}{\sum_{j=1}^{m} n_j} \sum_{j=1}^{m} \sum_{i=1}^{n_j} \sigma^2_{ji}, \tag{7}$$

where the second term is for regularization and λ is called the regularization coefficient. The training algorithm uses almost the same partial derivatives as those in the preceding section, except

$$\frac{\partial E(l)}{\partial \sigma_{ji}} = \left(\sum_{k=1}^{c} \delta_k(l) w_{kj} \right) \cdot \mathrm{P}_j(\mathbf{X}(l))$$

$$\times \left(-\frac{z_{ji}(l)}{\sigma_{ji}} \right) \left(-\frac{z_{ji}(l)}{\sigma^2_{ji}} \right) + \frac{\lambda}{\sum_{j=1}^{m} n_j} \sigma_{ji}.$$

3. Experimental results and discussions

3.1. Evaluating networks in the base-line system

3.1.1. Features and data sets

Two sets of features were used for character classification. One is called bar feature, which has 120 dimensions. The other is called transition feature, which has 100 dimensions. Detailed description of these features can be found in [5,28].

Because the word images used for tests can be strings in upper case or mixed case (the first character is in upper case), separate neural networks are used for upper and lower case character recognition. So, there may be four neural networks involved in our following experiments for each type of neural networks: two of them using bar feature inputs for lower and upper case classification, respectively; the other two using transition feature inputs for lower and upper case classification, respectively.

The image data used for training came from images of addresses from the USPS mail [5]. There are three data sets consisting of isolated character or non-character images. Data set 1 is referred to as Char250, which has 250 samples per category for lower and upper cases, respectively; data set 2 is referred to as Char1000, which has 1000 samples per category for lower and upper cases, respectively. Char250 is a subset of Char1000; data set 3 is referred to as Garbage8310, which has 8310 non-character samples. Garbage8310 consists of 3587 sub-garbage samples that are images of partial characters and 4723 super-garbage samples that are images of concatenated characters or concatenated partial characters.

The test data is a set of 317 word images (bd-317) from the CEDAR CD-ROM image database [29], which is referred to as BD317. No image in the training sets comes from BD317. The CEDAR CD-ROM set came with three lexicon sets generated to emulate mail-sorting applications. Two of them were used for testing using our base-line system, which performs lexicon-driven word recognition [5]. In both lexicon sets, there is one lexicon for each word image in BD317. The lexicons may be of different lengths. The lexicon sets are referred to as lex100 and lex1000 and have lexicons of average length 100 and 1000, respectively.

3.1.2. Training and test

We used a fuzzy k-nearest neighbor algorithm [27] to generate the desired outputs for valid character patterns. The effectiveness of fuzzy desired outputs for word level recognition was supported by the experiments reported in [28]. The values for desired outputs are in interval $[0.1, 0.9]$. All desired outputs are set to 0.1 for outlier training samples.

The learning rate was set to 0.02 for all neural networks. The training is completed when the error function converges and the classification performance on the validation set reaches its peak. The character recognition rate is the percentage of characters for which the highest neural net-

Table 1
Training (outliers included in the training data) and test results for bar feature MLPs

Network structure	Character level		Word level (BD317)	
	Character case	Char250 (%)	lex100 (%)	lex1000 (%)
$120 \times 65 \times 39 \times 26$	Upper case	94.40	79.81	63.41
	Lower case	89.11		
$120 \times 150 \times 26$	Upper case	93.65	84.23	72.56
	Lower case	88.10		

Table 2
Training (outliers included in the training data) and test results for transition feature MLPs

Network structure	Character level		Word level (BD317)	
	Character case	Char250 (%)	lex100 (%)	lex1000 (%)
$100 \times 65 \times 39 \times 26$	Upper case	88.32	76.66	57.10
	Lower case	78.43		
$100 \times 150 \times 26$	Upper case	91.00	82.97	65.93
	Lower case	80.34		

work output was associated with the true class. No garbage samples were used in the character classification test.

Word recognition was performed using the base-line system introduced in Section 1.1. The word recognition rate is the percentage of word images for which the truth string in the lexicon gets the highest match score. By substituting different neural networks in the system, we can compare their effects on word recognition.

3.1.3. MLPs with different number of hidden units

We believe that including outliers in the training data is important to encourage MLPs to form closed separation space. This was supported by our previous study [26]. So outlier data were included for MLP training, i.e., the training data consist of Char250 and Garbage8310. For bar feature MLPs, two structures are compared: $120 \times 65 \times 39 \times 26$ and $120 \times 150 \times 26$. The former demonstrated very good classification performance if no outlier is given as an input and was used historically. In fact, networks with these feature sets and architectures performed in the top 5 in an NIST sponsored OCR competition test included over 38 classifiers from 26 companies and academic groups [30]. The latter has more nodes in the hidden layer than that in the input layer, which is the requirement to form closed separation space as stated in [25]. Similarly, two structures are also compared for transition feature MLPs: $100 \times 65 \times 39 \times 26$ and $100 \times 150 \times 26$. All network parameters were randomly initialized for training.

Tables 1 and 2 show significant improvement on word recognition rates when more nodes are used in the hidden layer than the input layer. After checking word images in BD317, we found that the MLPs with the structure $120 \times 150 \times 26$ corrected some errors made by the MLPs with the structure $120 \times 65 \times 39 \times 26$ and introduced few new errors. We show two examples here. One example (Fig. 4) illustrates how an error was corrected after using the MLPs with structure $120 \times 150 \times 26$, which not only decreases the confidence values for outliers, but also decreases confidence values for non-perfect valid patterns (Fig. 4(c)). In this case, the word "Louisville" was mistakenly read as "Roundstone" by the MLPs with the structure $120 \times 65 \times 39 \times 26$. In the other example (Fig. 5), the error made by the MLPs with the structure $120 \times 65 \times 39 \times 26$ was apparently caused by assigning a high confidence value as "*k*" for union 10–12 (Fig. 5(c)). The MLP with structure $120 \times 150 \times 26$ significantly decreased this union's confidence value as "*k*", but it also made a (favorable) mistake by assigning a high confidence for union 8–8 as '*r*', which probably means that there is still some open separation surfaces after training. These results support the assertion that including the outlier samples in the training data and using more hidden nodes than required for classification can improve the outlier rejection performance for MLPs.

3.1.4. PCA-RBF trained with and without the regularization term

The initial centers for PCA-RBF networks were established based on *k*-means clustering on data set Char1000 with 11 clusters per class on average. Char1000 was used here in order to obtain a better representation of the feature distribution, which is important for principal component analysis for each cluster. With 98% of the total component energy retained after principal component analysis, each cluster retains 22 and 27 principal components on average for the bar and transition features, respectively. The initial expansion (σ_{ji}) was set as 2.0 times the square root

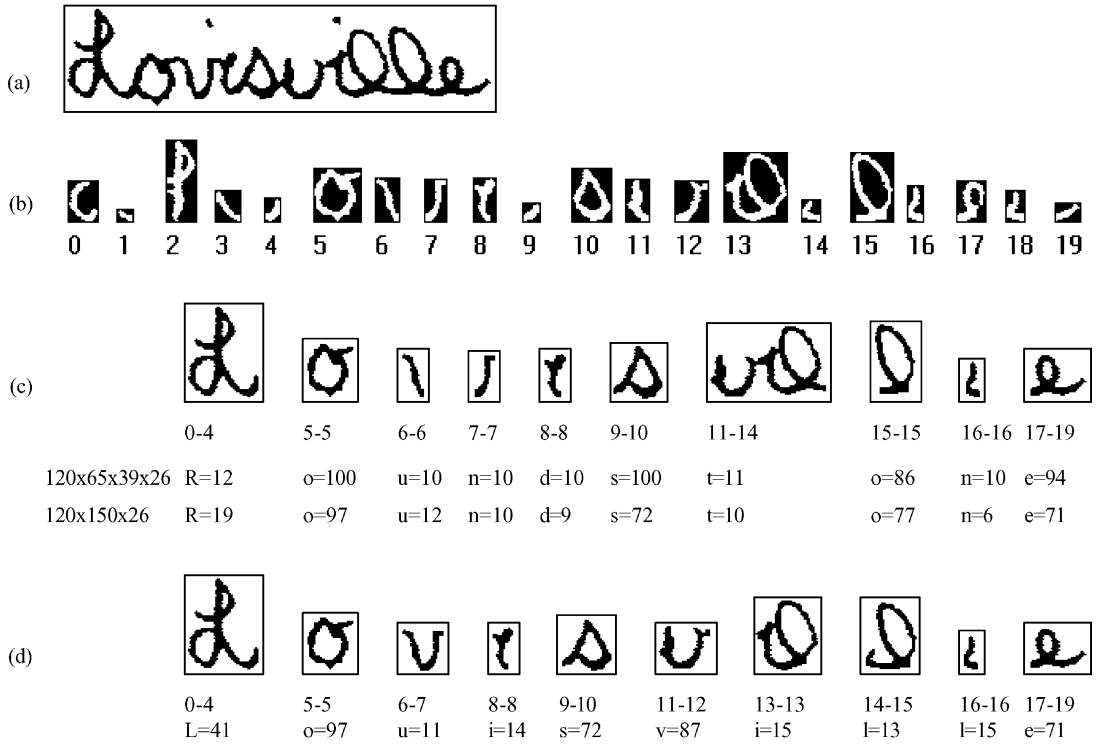


Fig. 4. Word recognition (lex100) using the bar feature MLPs. The truth string for this word image is "Louisville": (a) the word image; (b) the primitive sequence generated by the segmentation module; (c) the optimal segmentation obtained by the base-line system using the MLPs with the structure $120 \times 65 \times 39 \times 26$; (d) the optimal segmentation obtained by the base-line system using the MLPs with the structure $120 \times 150 \times 26$. The "digit–digit"s under union images in (c) and (d) indicate the primitives from which the union images are formed.

of the corresponding eigenvalue. The weights and biases were randomly initialized. During training, the centers of basis functions were fixed.

For PCA-RBF networks trained without the regularization term, Char1000 and Garbage8310 were used in training. The latter was used for the same reason as in MLP training, but the goal here is more intuitive, i.e., to limit the expansion of basis functions, which is controlled by σ_{ji}.

When PCA-RBF networks were trained with the regularization term, only Char1000 was used in training. The regularization term is supposed to limit the expansion of basis functions (Eq. (7)) during training. The regularization coefficient λ was empirically set as 0.02.

Tables 3 and 4 suggest that the regularization term can achieve outlier rejection effect close to that achieved using outlier data for PCA-RBF training. This means that by using the regularization term, the tedious and time-consuming work of collecting outlier data can be avoided and the CPU and memory resources for training can be reduced.

According to Tables 1–4, it seems that MLPs perform better on word recognition than PCA-RBFs do, because even though their performances are close to MLPs, PCA-RBF networks have more parameters and used more samples in training. We think the reason is that the classification ability of PCA-RBF networks is not as good as that of MLPs.

3.2. Combining neural networks for word recognition

We known that MLPs are good at pattern classification tasks for which the input is known to be from a finite set of pattern classes. Usually, their outlier rejection performance is not so good because the separation surfaces are not closed. Even though we can increase the number of units in the hidden layer to improve their outlier rejection performance, there may still be open separation surfaces if the outlier samples that can lead to shrinking openness are not presented in training.

On the contrary, PCA-RBF networks with localized basis functions inherently have closed separation surfaces. Generally, they are not so good at pattern classification tasks as MLPs are.

Using weighted average over character level confidences assigned by different neural networks, we hope the complement between MLPs and PCA-RBF networks can be exploited to improve the word recognition performance. In our test, four weights w_1–w_4 were applied to four neural networks, i.e., bar feature MLP, transition feature MLP,

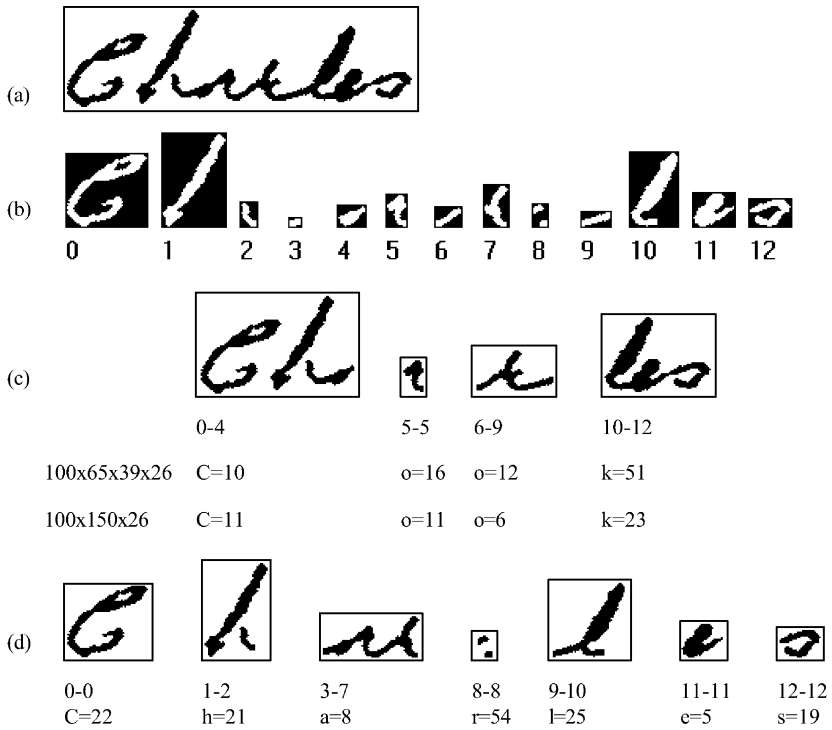


Fig. 5. Word recognition (lex100) using the bar feature MLPs. The truth string for this word image is "Charles": (a) the word image; (b) the primitive sequence generated by the segmentation module; (c) the optimal segmentation corresponding to string "Cook" obtained by the base-line system using the MLPs with the structure $120 \times 65 \times 39 \times 26$; (d) the optimal segmentation corresponding to string "Charles" obtained by the base-line system using the MLPs with the structure $120 \times 150 \times 26$. The "digit–digit"s under union images in (c) and (d) indicate the primitives from which the union images are formed.

Table 3
Training and test results for bar feature PCA-RBF neural networks

Training method	Character level		Word level (BD317)	
	Character case	Char1000 (%)	lex100 (%)	lex1000 (%)
Outliers are used	Upper case	87.20	82.65	70.03
Regularization term is not used	Lower case	73.29		
Outliers are not used	Upper case	88.76	82.96	67.82
Regularization term is used	Lower case	74.77		

Table 4
Training and test results for transition feature PCA-RBF neural networks

Training method	Character level		Word level (BD317)	
	Character case	Char1000 (%)	lex100 (%)	lex1000 (%)
Outliers are used	Upper case	89.43	82.65	64.67
Regularization term is not used	Lower case	74.45		
Outliers are not used	Upper case	90.26	81.07	64.35
Regularization term is used	Lower case	75.64		

Table 5
Word recognition results by neural network combination on data set BD317

	w_1 (bar/MLP)	w_2 (tsn/MLP)	w_3 (bar/PCA-RBF)	w_4 (tsn/PCA-RBF)	lex100 (%)	lex1000 (%)
Average all networks	0.25	0.25	0.25	0.25	88.01	74.76
Average bar feature networks	0.5	0	0.5	0	85.80	75.08
Best weighted average	0.45	0.1	0.45	0	87.38	77.60
SOFM/MLP					87.70	76.03

bar feature PCA-RBF network (trained with the regularization term) and transition feature PCA-RBF network (trained with the regularization term), respectively, with the weights satisfying

$$\sum_{i=1}^{4} w_i = 1, \quad w_i \geqslant 0.$$

The best results obtained by exhaustive search of weights on BD317 with step 0.05 is shown in Table 5. Compared to the results in Tables 1–4, this result shows the potential of combining MLPs and PCA-RBF networks. The SOFM/MLP approach proposed by Chiang and Gader [6] uses SOFM to achieve the clustering effect and form uniform representation for outliers. In addition, the SOFM/MLP approach also combines the bar feature and transition feature. Thus, the SOFM/MLP has similar conceptual goals to those of the networks in this paper and they use the same features and dynamic programming algorithm. The SOFM/MLP used human selection of initial prototypes in the training of the SOFM, whereas the networks in this paper did not require human intervention. Therefore, training of the networks in this paper is more automated. The comparative results are presented in Table 5.

4. Conclusions and future work

This study investigated the method to enhance the outlier rejection ability of MLPs and PCA-RBF networks for handwritten word recognition. The experimental results support the following conclusions:

1. Using more hidden nodes than required for classification for MLPs and including outliers in the training data can improve outlier rejection performance.
2. Training the PCA-RBF network with the regularization term and no outlier can achieve as good a performance as training with outliers.

The open question is what the optimal number of hidden unit is for MLPs and PCA-RBF networks for specific applications that require outlier rejection ability, such as our handwritten word recognition system. Currently, it is very difficult to obtain analytical solutions for neural networks working in high dimensional feature spaces. "Trial and error" approaches are usually limited by computing resources for real world problems where high dimensional features are to be handled.

Our test on combining MLPs and PCA-RBF networks just showed the potential to improve word recognition performance by exploiting the complement of these two kinds of neural networks. Objective experiments should be carried out to obtain the weights on independent training data. Other classifier combination schemes should also be pursued [31,32].

Acknowledgements

This work was supported by National Science Foundation under Grant No. 9732914. The authors would like to thank Dr. W. Chen for helpful discussions.

References

[1] R. Plamondon, S. Srihari, On-line and off-line handwriting recognition: a comprehensive survey, IEEE Trans. PAMI 22 (1) (2000) 63–84.
[2] F. Kimura, M. Shridhar, Z. Chen, Improvements of a lexicon directed algorithm for recognition of unconstrained handwritten words, Proceedings of the Second International Conference on Document Analysis and Recognition, Tokyo Japan, 1993, pp. 18–22.
[3] M. Shridhar, G. Houles, F. Kimura, Handwritten word recognition using lexicon free and lexicon directed word recognition algorithms, Proceeding of the Fourth International Conference on Document Analysis and Recognition, Ulm Germany, August, 1997.
[4] G. Kim, V. Govindaraju, A lexicon driven approach to handwritten word recognition for real-time applications, IEEE Trans. PAMI 19 (4) (1997) 366–379.
[5] P. Gader, M. Mohamed, J. Chiang, Handwritten word recognition with character and inter-character neural networks, IEEE Trans. SMC 27 (1) (1997) 158–165.
[6] J.H. Chiang, P. Gader, Hybrid fuzzy-neural systems in handwritten word recognition, IEEE Trans. Fuzzy Systems 5 (4) (1997) 497–511.
[7] W. Chen, P. Gader, H. Shi, Lexicon driven handwritten word recognition using optimal linear combinations of order statistics, IEEE Trans. PAMI 21 (1) (1999) 77–82.

[8] A. Kundu, Y. He, P. Bahl, Recognition of handwritten word: first and second order hidden Markov model based approach, Pattern Recognition 22 (3) (1989) 283–297.
[9] M. Chen, A. Kundu, J. Zhou, Off-line handwritten word recognition using hidden Markov model type stochastic networks, IEEE Trans. PAMI 16 (5) (1994) 481–496.
[10] A. El-Yacoubi, M. Gilloux, R. Sabourin, C.Y. Suen, An HMM-based approach for off-line unconstrained handwritten word modeling and recognition, IEEE Trans. PAMI 21 (8) (1999) 752–760.
[11] A. Gillies, Cursive word recognition using hidden Markov models, Proceedings of the Fifth United States Postal Service Advanced Technology Conference, Washington, DC, November, 1992, pp. 557–563.
[12] M. Mohamed, P. Gader, Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques, IEEE Trans. PAMI 18 (5) (1996) 548–554.
[13] A. Senior, A. Robinson, An off-line cursive handwriting recognition system, IEEE Trans. PAMI 20 (3) (1998) 309–321.
[14] W. Cho, S.W. Lee, J.H. Kim, Modeling and recognition of cursive words with hidden Markov models, Pattern Recognition 28 (12) (1995) 1941–1953.
[15] J. Simon, Off-line cursive word recognition, Proceedings of the IEEE 80 (7) (1992) 1150–1161.
[16] S. Madhvanath, V. Govindaraju, Holistic lexicon reduction, Proceedings of the Third International Workshop on Frontiers in Handwriting Recognition, Buffalo, NY, 1993, pp. 71–82.
[17] J. Hull, T. Ho, J. Favata, V. Govindaraju, S. Srihari, Combination of segmentation-based and holistic handwritten word recognition algorithms, Proceedings of the Second International Workshop on Frontiers in Handwriting Recognition, Chateau de Bonas, France, 1992, pp. 229–240.
[18] S. Madhvanath, E. Kleinger, V. Govindaraju, Holistic verification of handwritten phrases, IEEE Trans. PAMI 21 (12) (1999) 1344–1356.
[19] J. Mao, P. Sinha, M. Moidin, A system for cursive handwritten address recognition, International Conference on Pattern Recognition (ICPR), Brisbane, Australia, 1998.
[20] P. Gader, M. Whalen, M. Ganzberger, D. Hepp, Handprinted word recognition on a NIST data set, Mach. Vision Appl. 8 (1995) 31–40.
[21] C. Scagliola, G. Nicchiotti, F. Camastra, Enhancing cursive word recognition performance by the integration of all the available information, Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, Amsterdam, Netherlands, September, 2000, pp. 363–372.
[22] J.H. Chiang, P. Gader, Improving digit recognition reliability by a hybrid neural model, in: W. Chiang, J. Lee (Eds.), Proceedings of the International Joint Conference of CFSA/IFIS/SOFT '95 on Fuzzy Theory and Applications, 1995, pp. 182–187.
[23] S. Kim, S. You, Y. Choi, Rejection of garbage patterns for improving reliability in continuous handwritten numeral recognition, Proceedings of the Fourth International Conference on Soft Computing, Vol. 2, 30 September–5 October, 1997, Fukuoka, Japan, pp. 778–781.
[24] S.F. Fogelman, B. Lamy, E. Viennet, Multi-modular neural network architectures for pattern recognition: applications in optical character recognition and human face recognition, Int. J. Pattern Recognition Artif. Intell. 7 (4) (1993) 721–756.
[25] M. Gori, F. Scarselli, Are multilayer perceptrons adequate for pattern recognition and verification? IEEE Trans. PAMI 20 (11) (1998) 1121–1132.
[26] J. Liu, P. Gader, Outlier rejection with MLPs and variants of RBF networks, Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, September, 2000.
[27] J. Keller, M. Gray, J. Givens, A fuzzy k-nearest neighbor algorithm, IEEE Trans. SMC-15 (4) (1985) 580–585.
[28] P. Gader, M. Mohamed, J. Chiang, Comparison of crisp and fuzzy character neural networds in handwritten word recognition, IEEE Trans. Fuzzy Syst. 3 (3) (1995) 357–363.
[29] J. Hull, A database for handwritten text recognition research, IEEE Trans. PAMI 16 (5) (1994) 550–554.
[30] R. Wilkenson, J. Geist, S. Janet, P. Grother, C. Burges, R. Creecy, B. Hammond, J. Hull, N. Larsen, T. Vogl, C. Wilson, The First Census Optical Character Recognition Systems Conference, National Institute of Standards and Technology, Gaithersburg MD, NISTIR 4912, August 1992.
[31] P. Gader, M. Mohamed, J. Keller, Fusion of handwritten word classifiers, Pattern Recognition Lett. 17 (1996) 577–584.
[32] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, A. Gelzinis, Soft combination of neural classifiers: a comparative study, Pattern Recognition Lett. 20 (1999) 429–444.

About the Author—PAUL GADER received his Ph.D. in Applied Mathematics (applications to image processing) in August 1986 from the University of Florida. After receiving his Ph.D. for research in Image Processing, he worked as a Senior Research Scientist at Honeywell Systems and Research Center, as an Assistant Professor of Mathematics at the University of Wisconsin, Oshkosh, as a Research Engineer and Manager at the Environmental Research Institute of Michigan (ERIM). He then served on the faculty of the University of Missouri, Columbia for 10 years before returning to join the faculty of the University of Florida in 2001.

He has worked on a variety of basic and applied research problems in both industrial and academic settings. He has worked on basic mathematical research, various image processing applications, obstacle detection, and other problems. Recently, he has become extremely active in the area of landmine detection. He has been actively involved in handwriting recognition research since 1989. At ERIM, he led a project to develop computer systems to read handwritten addresses funded by the U.S. Postal Service Office of Advanced Technology. He continued this work when he returned to an academic position at the University of Missouri, first as a subcontractor and then independently. He has performed research in handwritten and machine-printed line, word, and character segmentation, on ZIP code and street number location and recognition, on P.O. Box detection and recognition, on handwritten digit and alphabetic character recognition, on handwritten word recognition, and on multiple classifier fusion in digit, character, and word recognition.

Dr. Gader has over 150 technical publications, including 41 refereed journal articles.

About the Author—JINHUI LIU received the B.S., M.S. and Ph.D. degrees in Electronic Engineering from Tsinghua University, Beijing, China, in 1990, 1992 and 1997, respectively. He was a postdoctoral researcher in the Department of Computer Science and Engineering and Center for Microbial Ecology at Michigan State University from 1997 to 1999 and was a visiting scientist at the IBM Almaden Research Center from July to October 1998. He was a postdoctoral researcher in the Department of Computer Engineering and Computer Science at the University of Missouri, Columbia from 1999 to 2000. He is now a technical staff member at Verity, Inc. His research interests include Document Image Understanding, Pattern Recognition and Machine Intelligence. He has performed research on Handwriting Recognition, Landmine Detection, Form Processing, Handprinted Chinese Character Recognition, Postal Address Recognition and Bacterial Image Analysis.