Neural Network-Based Systems for Handprint OCR Applications

Michael D. Garris, Charles L. Wilson, *Senior Member, IEEE,* and James L. Blue

Abstract—**Over the last five years or so, neural network (NN)-based approaches have been steadily gaining performance and popularity for a wide range of optical character recognition (OCR) problems, from isolated digit recognition to handprint recognition. In this paper, we present an NN classification scheme based on an enhanced multilayer perceptron (MLP) and describe an end-to-end system for form-based handprint OCR applications designed by the National Institute of Standards and Technology (NIST) Visual Image Processing Group. The enhancements to the MLP are based on i) neuron activations functions that reduce the occurrences of singular Jacobians; ii) successive regularization to constrain the volume of the weight space; and iii) Boltzmann pruning to constrain the dimension of the weight space. Performance characterization studies of NN systems evaluated at the first OCR systems conference and the NIST form-based handprint recognition system are also summarized.**

Index Terms— **Boltzmann weight pruning, handprint, multilayer perceptron, neural networks, optical character recognition, public domain.**

I. Introduction

OPTICAL character recognition (OCR) research using electronic and electro-mechanical methods dates back to the 1950's. This work was initially directed toward machine print and was restricted to fixed-format applications. In the late 1980's, decreasing costs for collection and transmission of electronic images and increasing computer power generated renewed interest in OCR for the unrestricted machine print problem and for handprint on forms. The key modules that make up an end-to-end system for handprint OCR applications are preprocessing (form identification, field and line isolation), character segmentation, segment reconstruction, character recognition, and word and phrase construction without human intervention or human correction. For many of these modules, methods based on two primary approaches have been taken: rule-based and image-based recognition.

The rule-based approaches [1]–[4] have applied existing image analysis techniques in a straightforward manner. The image-based approaches originated from the application of classical pattern recognition techniques but are now dominated by neural network (NN) approaches. The most useful characteristics of NN's are their ability to learn from examples, their ability to operate in parallel, and their ability to perform well using data that are noisy or incomplete.

In 1989, the Image Recognition Group, National Institute of Standards and Technology (NIST) began developing methods for testing large-scale OCR systems for handprint on forms for Census and Internal Revenue Service (IRS) applications. In May, 1992 [5] and February, 1994 [6], NIST conducted two OCR Systems Conferences that were designed to provide baseline information on OCR technology and specific application-related information on Census industry and occupation data from the 1990 Census (these conferences will be referred to collectively as the NIST OCR Systems Conferences, and separately as the First OCR Systems Conference and the Second OCR Systems Conference.) All of these efforts were designed to provide a common framework for the evaluation of research results in machine and handprinted OCR that would allow commercial feasibility to be tested.

The results of the First OCR Systems Conference [5] demonstrated the applicability of a wide variety of pattern recognition methods for solving the isolated character recognition problem. NN methods were the most common in the high accuracy systems. The most accurate of these systems perform comparably to humans on recognition of isolated characters; most systems provide estimates of recognition confidence which, with proper rejection, allow results to be produced that are more accurate than can be produced by humans on the part of the data that remains after rejection.

In the First OCR Systems Conference, strictly rule-based methods were uniformly less accurate than methods based on machine learning or Bayesian statistical models. The general procedure used in the most successful recognition systems was to develop large, rich feature sets and use a large number of examples to train a recognition device. If the training and test sets used are sufficiently similar, these methods produce results comparable to the best previously published recognition rates. The remaining research challenge in this area is dealing with the last few percentage points of error. These cases are also hard for humans, and usually include both ambiguous and rare forms of characters.

In this paper, we present a detailed exposition of our NN-based approach to a practical OCR problem. In particular, our focus is on a specific handprint recognition problem which meets three constraints. First, the data is on forms, which implies that the data of interest is found in predetermined locations. Second, the image quality is sufficient to provide legible images, which implies that the forms have adequate space to enter the required information and that scanning

Manuscript received December 27, 1996; revised September 12, 1997.

The authors are with the Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (e-mail: mgarris@nist.gov).

Publisher Item Identifier S 1057-7149(98)05316-0.

resolution is sufficient to resolve the text. Third, the material to be read from the form has specific, known content, which restricts the lexicon of expected answers. The success or failure of form-based OCR is strongly influenced by the degree to which the application adheres to these constraints. Good form design is essential to economical OCR-based data input and is application-specific. High-quality scanning and good image quality are necessary to lower processing cost and allow the use of electronic images without extensive cleanup processing. Prior knowledge of field content is important both for recognition correction and for the detection of human errors in placing information on forms.

We describe the design of an NN classifier for handprint classification problems. This NN is an improved MLP with the enhancements coming from well-conditioned Jacobians, successive regularization, and Boltzmann pruning techniques. The enhanced MLP improves the error-reject performance on handprint classification problems by factors of two to four, while at the same time reducing the complexity (number of nonzero weights) of the network by about 40% to 60%. The effectiveness of the NN classifier is further illustrated by integrating it into an end-to-end system for form-based handprint recognition applications developed by the Visual Image Processing Group at NIST. The performance of the NIST system is characterized using carefully designed evaluation experiments.

The organization of the paper is as follows: A brief review of existing NN algorithms and systems for feature extraction, classification and recognition in the context of OCR applications is in Section II. The enhanced MLP classifier is described in Section III. Section IV presents the NIST form-based handprint recognition system followed by a discussion of performance characterization results in Section V. Finally, summary and conclusions are in Section VI.

II. Prior NN-Based Approaches

A. Feature Extraction

The feature extraction stage is an important component of any recognition system. It is also very much dependent on the task, input, and recognition algorithm used. In OCR applications, the types of features extracted depend on whether gray-scale, binary, or vector representations are used. If statistical pattern recognizers are used, then the standard features tend to be based on contours of characters, with features such as Fourier descriptors, moment invariants, and other boundary features. If structural recognizers are used, features that represent the structure of the character are preferred.

With NN's, the feature extraction stage has gone through some evolutionary developments. Earlier NN approaches used local feature maps [7], [8]. Methods developed later sidestepped the issue of feature extraction, and used the segmented character or digit image as the input [9]. This works well for the case of isolated digits or characters, but when touching characters or digits are present, the input image (corresponding to a zip code, handprinted, or cursive word [10]) is difficult to accurately segment into isolated characters.

One of the important considerations in the feature extraction stage is invariance with respect to shape transformations such as translation, rotation, and dilation. Some features are designed to be insensitive to these transformations, while in other cases transformations of the extracted features achieve invariance to some of the shape transformations. Another consideration is whether the original patterns (digits or characters) can be reconstructed. Global descriptions such as Fourier descriptors, moments, and coefficients of circular autoregressive models can reconstruct the original patterns, the quality of reconstruction being dependent on the number of coefficients or parameters retained.

A recent survey of most of the feature-extraction techniques used in statistical pattern recognition and rule-based systems for OCR can be found in [11]. The rule-based approach to OCR has traditionally relied on classification or matching of specific features such as those mentioned in [11]. NN approaches have rather used windows of raw pixels or global transforms such as principal component analysis [12] or the projection on to a Gabor basis set. Features derived from principal components have been used in the NIST form-based recognition system [12], [13]. Projections on the Gabor basis set have been used in the NIST [14] as well as the KODAK system [15]. In [15], the utility of global transforms such as Gabor, Fourier, and cosine has been evaluated as a source of features for a standard NN recognizer of isolated handprinted characters. Experimental results indicate that when windows of raw image pixels are replaced by projections on the Gabor basis set, significant reductions in errors have been observed. An NN recognizer using Gabor basis set projections was evaluated in the First OCR System Conference and was one of a tightly bunched group of leaders. This feature set was also used in the KODAK system reported in Section II-D.

Deformable spline models whose shapes are determined by the positions of eight control points have been suggested as features [16] for handprinted digit recognition in zip codes. Shapes are deformed when the control points wander. An energy function that penalizes shape deformations due to displacement of the control points is minimized. The fitted models are then used for recognition by deciding on the model of the digit that could have generated the data. One of the vexing problems of fitting deformable templates is initialization of the parameters characterizing the surface. In [17], an NN is used to learn the associations between images and the instantiation parameters of deformable templates that characterize these images. A three-layer NN was trained using the backpropagation (BP) technique and the CEDAR CD-ROM data base [18]. The trained NN was used to provide better initial conditions for the new images.

B. Neural Network Classification

Since the late 1980's, applications of NN's to recognition of handprinted digits, characters, and cursive handwriting has become a very active area. Earlier attempts concentrated on recognition of handprinted, isolated digits and characters, with more recent efforts looking at integrated segmentation and recognition (ISR) methods.

One of the earlier papers that explored the applicability of NN's, along with classical recognition methods such as Parzen window and k-nearest-neighbor techniques, is reported in [7]. The problem considered is that of recognizing handwritten zip code digits. After preprocessing steps, such as skew correction and skeletonization, about 49 convolutional masks are used for feature extraction, focusing on lines and line ends. Examples of typical features include horizontal strokes, right-hand ends of horizontal strokes, etc. These 49 feature outputs are combined (subsets are combined using logical AND or OR) to produce 18 feature maps. These feature maps are input to a Parzen window, a k-nearest-neighbor classifier, and a two-stage NN. Although initial results did not show improved performance using the NN, a finely-tuned preprocessor and a large training set enabled the NN to exceed the performance of the statistical pattern recognition techniques mentioned above. In a follow-up work [19], a multilayer network with BP training has been used for the handwritten digit recognition problem. Using a combination of alternating hidden and averaging/sampling layers and local convolution feature detectors in the hidden layers, improved classification rates are reported. One of the issues in situations involving multiple layers is the need to keep the number of free parameters to a minimum. Using the concept of weight sharing [20], which in effect imposes shift invariance on the feature maps, the number of free parameters is reduced. The effects of weight pruning using principal component analysis (PCA), optimal brain damage (OBD) [8], and weight decay are reported in [19] for a modest data set consisting of handwritten digits. The effects of smoothing followed by regularization, as well as using higher-order networks, have also been studied. All of these enhancements yield improved performance.

Instead of the local feature maps used in [7] and [21], a holistic approach that uses PCA to generate representations for handwritten digit images is reported in [13]. This can be viewed as an "eigen-digits" approach to digit recognition. A multilayer network with BP training has also been independently reported in [9], for recognizing handprinted digits scanned from bank checks and letters interactively entered through a stylus digitizer. This work argues in favor of a large representative training set. Another departure from [7] and [21] is the use of presegmented, size-normalized gray-scale character or digit images, instead of local feature maps. The weight sharing principle is used to reduce the number of free parameters. The absence of specific feature maps appears to have enabled easy generalization to integrated segmentation and recognition schemes, as discussed later on.

An application of autoassociative MLP to the recognition of handwritten characters taken from a subset of the NIST 19 data set is reported in [22]. In this work, a network is designed for each character, using only instantiations of that character for training. Each NN can thus be viewed as generating a discriminant function. Since it is preferable for an object recognition system to be insensitive to transformations such as translation and rotation, error measures derived from tangent distances [23] are suggested for learning. One can view this approach as letting the NN learn (or unlearn) skew. Since in most applications, one can deskew and normalize using simple image-processing operations, the advantage of requiring the NN to account for translation and rotation at the cost of increasing the complexity of the NN may be minimal. One can obtain better payoffs by integrating segmentation and recognition, as discussed below.

By feeding candidate segments produced by vertical cuts to a recognizer and scoring the output of the recognizer, the best possible segmentation of an input digit string is obtained in [24]. Knowledge of the number of the digits expected is used as an input; since the recognizer used is a convolutional NN, the best segmentation can be obtained by segmenting the feature map.

C. Integrated Segmentation/Recognition

For the recognition of touching characters, a method that performs ISR using a BP algorithm is described in [25]. In a subsequent extension of this work [26], an NN with two hidden layers, the second of which is connected to a block of exponentials before being connected to the output layer, is used for simultaneous segmentation and recognition of handwritten digits from the NIST data base. The interconnections between the hidden layers, the second hidden layer, and the exponential block are local three-dimensional (3-D) interconnections. The output levels are derived from ratios of the sums of the responses of the exponential blocks. The synaptic weight updating rules are derived from minimizing the total sum of squared error measure. A combination of hand-segmented and original (not hand-segmented) field data sets is used for training two NN's. The major drawback may be the need to normalize the digits, albeit crudely. The system is able to handle normalizations up to a factor of two or more; severe changes in scale seem to reduce the accuracy by 3–5%.

The NN's for ISR described above have primarily used multilayer feed-forward networks trained using the BP technique. It has been argued in the NN literature that networks that permit feedback from activation levels at the output could yield improvements in classification as well as generalization. One form of a network that allows such a feedback is the recurrent NN [27]. In [28], the use of a recurrent NN for ISR of digits in the NIST database has been experimented with. Comparisons with other NN-based ISR techniques using the same NIST dataset seem to indicate that the recurrent NN method compares favorably to other NN-based ISR techniques.

D. Some NN-Based Handprint OCR Systems

The NIST OCR System Conferences evaluated a number of OCR systems developed by university researchers and companies in the United States and abroad. A general discussion of the evaluations is presented in Section V. Due to proprietary considerations, only a high level understanding of system configurations and module functionalities has been possible. For the sake of completeness, we present brief descriptions of three of the systems evaluated at the NIST OCR System Conferences. These are the systems designed at ERIM [29], [30], MCC [31], [32] and KODAK [33]. All of these systems have been improved since the 1994 evaluation. Descriptions

of ERIM and MCC systems as they existed in 1994 may be found in [6].

ERIM: The ERIM approach uses an effective combination of techniques such as feature-based matching, NN's for learning feature attributes, and unsupervised clustering schemes for generating character prototypes. The feature matching approach is based on Ullman's minimal mapping theory [34]. After preprocessing steps such as noise removal, slant correction, and scaling to a fixed height, feature extraction driven by morphological operations is performed on multi-letter images. By identifying valleys in the core stroke image, possible segmentation points (PSP's) are identified and features are extracted between pairs of PSP's. The features used are the core area (where the center stroke information falls) and the discrete features extracted from six feature images (stroke above and below the core, holes and concavities, lower and upper limbs in the core). Finding the discrete features amounts to extracting connected components for holes, concavities, and lower and upper strokes; for lower and upper limbs, the extracted connected components are broken into local maxima and minima. To generate the matches between character samples and character models, a random graph structure is used in which vertices represent character features and edges represent the spatial relationships between feature pairs. Features extracted as described above are represented using the attributes of position and shape. The shape attribute is extracted using a NN training scheme.

For each PSP pair, a correspondence matching algorithm formulated as an assignment problem is invoked. For each string in the given lexicon, an optimal match between the word image features and the model of the string is determined using a dynamic programming (DP) technique. Since handprinting can produce many variations, several character templates for each of the 26 lowercase printed, 26 uppercase printed, and 26 lowercase cursive characters are generated from the training set using an unsupervised clustering procedure. A total of 457 character model prototypes derived from 996 word images is used.

One of the important features of the ERIM work is the ability to generate confidence measures along with the matched word. The word confidence measure and the ranking is derived from the probability that a word model for the given string generated the set of features extracted from the word image. The distributions for the position and shape attributes of the extracted features are derived as part of the feature extraction process. Using assumptions about the conditional independence of the probability distribution of features, one can generate the probability of a word from probabilities of its component features.

In [30], ERIM presented the design details of a phrase recognition system, where a phrase can include multiple words, abbreviations, etc. Such applications arise in matching a data base record such as business, street name, etc. The key components of this system are a word segmentor, a word recognizer, a lexicon generator, and a dynamic phrase recognizer. As the intent is to handle both printed and cursive words, several techniques are used in the system. In the case of the word segmentor module, machine-printed words can be separated by using simple projection-based techniques. Since projection-based techniques do not work for cursive words, both over- and undersegmentation of the line image is done to mark possible locations of word breaks. Undersegmentation is done using breaks at large gaps between successive components, while oversegmentation is done at locations such as capital letters, punctuation, etc., to produce multiple hypotheses to be evaluated by the word recognizer. To keep the system's capabilities as general as possible, evaluation of word hypotheses is done using four kinds of word recognizers. Two of the word recognizers use hidden Markov models trained on printed and cursive words, respectively. The third is a word recognizer designed using the feature correspondence matching approach described earlier. The fourth recognizer is based on segmentation followed by evaluation using a dynamic programming technique. Since all of these modules return their top few choices and the associated confidence measures, a voting scheme and reject curve are used for deciding on the top choice and its new confidence measure.

The recognized words are structured into phrases using a lexicon-directed dynamic programming approach. For the case of matching of street names, the phrase lexicon is generated using aliases, abbreviations, suffixes, and prefixes of street names. The results of experiments with 1200 handwritten mail-pieces containing street address information indicated acceptable performance.

MCC: The MCC approach is based on an ISR technique using NN's [35], [32]. The network uses a sliding window and attempts to recognize a character only if the window is centered on it. If the window center is between two characters, a NOT-CENTERED condition is declared. The training set is prepared by a human who clicks at the horizontal center of each digit in sequence using a mouse button, and associates a digit label with each such center. The target output is generated during training as follows: When the center position of the window is within two pixels of the center of a character, the target value of the character's output node is set at the maximum with the target values of the NOT-CENTERED mode and all other characters set at the minimum. When the center position of a window is within two pixels of the halfway point between two character centers, the reverse situation holds. The NN is a two-hidden-layer MLP network, with local shared connections in the first hidden layer, and local connections in the second hidden layer. It has been observed that the weight-sharing techniques used to reduce memory requirements often yielded poor results.

This approach has been extended to handle handprint digits and characters [32]. An added feature to the approach described above is the introduction of a saccadic scan. This is accomplished by training an MLP net to estimate the distance to the next character on the right so that at run-time, the system can navigate along a character field by effectively jumping over the spaces between characters. This effectively reduces the number of forward passes required compared to the exhaustive scanning scheme [35]. In addition to the saccade system, when the recognition of handwritten characters is considered, postprocessing techniques that use dictionaries result in significant reductions in the field error rate.

KODAK: It has been observed [33] that the MCC system using saccades worked well on nicely spaced characters, but if the localization was on intercharacter space, then the saccade could move to the previous character and classify it the second time. The reason given for this is the discontinuities in weight values in the first processing layer lead to significant changes in input that get fed to the NN when even small shifts are present in the input image. The saccade network in the MCC system has the burdensome task of making saccades as well as classification of characters using a single complicated network. The KODAK system appears to be an improvement over the MCC system using the following modifications: i) projections of Gabor-basis sets are used as inputs instead of raw input image windows, and ii) the task of positioning and classification are done by two separate NN's.

The positioning network accepts as inputs size-normalized subfields of disconnected groups of characters, with a spacing of six pixels and with two black rows appended at the top and bottom. The resulting image is processed by a feature extraction layer that basically performs projection onto Gabor basis sets. The input layer is followed by three hidden layers and an output layer with 12 outputs. As the input window slides to each pixel position in the input field, the output layer produces 12 activations corresponding to the chosen pixel position in 12 consecutive windows. Using a weighted average of these activations and some post processing, a waveform with peaks characterizing the positions of characters is produced. Based on extensive testing with the NIST data base, KODAK reports that 91% of all characters were positioned to within a pixel. Comparisons with the performance of the MCC saccade system show the superior performance of the positioning network.

The classifier network is similar to the positioning network, but has only two hidden layers, and an output layer with ten outputs for digits. To account for the errors in positioning the characters, the classifier network was trained and tested on five copies of input fields corresponding to shifts of up to and including two pixels on either side of the center pixel. Comparison to the saccade system shows improved performance.

An interesting part of this work [33] is the system-level comparison of performance with the SACCADE, NESTOR, AEG, and IBM systems. Using a small test data set (240 fields with 1492 characters) from IRS forms, the authors conclude that the KODAK system performed better than the others.

Other applications of NN's include modifications to the Neocognitron [36] and a booster technique for improving the accuracy of NN approaches [37].

III. NIST's New MLP Neural Network

In previous work on character and fingerprint classification [38], probabilistic neural networks (PNN) were found to be superior to MLP networks in classification accuracy. This comparison was conducted using global Karhunen–Loéve (KL) transforms as the feature set. In later work, [39], combinations of PNN and MLP networks were found to be equal to PNN in accuracy and to have superior error-reject performance.

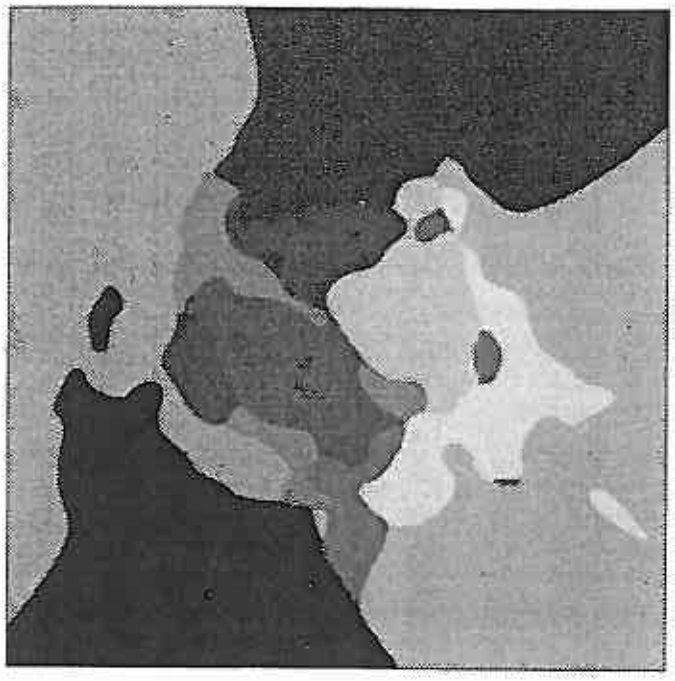

Fig. 1. Digit classifications generated by a PNN classifier using the first two KL components in a region centered on (0.0).

These results were achieved by using 45 PNN networks to make binary decisions between digit pairs and combining the 45 outputs with a single MLP. This procedure is much more expensive than conventional MLP training of a single network and uses much more memory.

Analysis of the results of the binary decision network [39] for isolated digit recognition showed that the feature space used in the recognition process had a topological structure whose local intrinsic dimension [40] was 10.5, although the global KL transform dimension was approximately 100. For binary decision machines, typical feature set sizes required for good accuracy were 20 to 28, significantly fewer than the number of features required by the global problem for MLP's, 48 to 52. The number of features, 20 to 28, needed to make binary decision machines discriminate between digits was larger than the intrinsic dimensionality, 10.5. Similar tests showed a comparable structure in the fingerprint feature data. The low local dimensionality explains some of the difficulty of these problems; the MLP is being used to approximate a complex fractal object, the set of decision surfaces, which has a typical local dimension of 10.5, embedded in a space of much larger dimension. Since the domain of each prototype in the PNN network is local, the PNN can more easily approximate surfaces with this topology. Fig. 1 shows a typical PNN decision surface and Fig. 2 shows a typical MLP decision surface. See [38, Fig. 8] for additional examples of this type of local structure in PNN-based recognition.

The local nature of PNN decision surfaces also explains why MLP's have better error-reject performance. It was shown in [41] that the error-reject curve decreases most rapidly when binary choices are made between classes. The decision surfaces in Fig. 1 are such that, as the radius of a test region

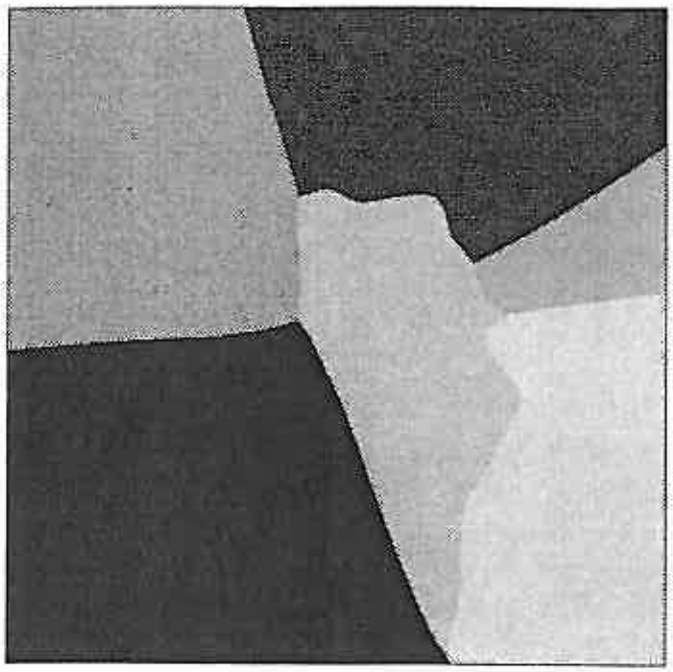

Fig. 2. Digit classifications generated by a MLP classifier using the first two KL components in a region centered on $(0, 0)$.

expands, multiple class regions are intersected, flattening the error-reject curve. Simpler class decision surfaces result in better error-reject performance. Thus, the shape of the error-reject curve can be used to assess the complexity of decision surfaces.

In this section, we show that three modifications to the conjugate gradient (SCG) method discussed in [42] will allow a three-layer MLP to approximate the required decision surface with zero-reject error similar to PNN and k-nearest-neighbor (KNN) methods. The error-reject performance is better than the best binary decision method discussed in [39]. This indicates that the resulting decision surfaces are both the most accurate and the simplest approximations to the character and fingerprint classification problem yet found.

A. MLP Problems Addressed

There are several difficulties that hamper the training, or limit the usefulness, of MLP's. We summarize some of them briefly.

1) Given a three-layer MLP with N_i input neurodes, N_h hidden neurodes, and N_o output neurodes, training is done by minimizing the error or objective function, E. There are generally many local minima of E, and each of them occurs $N_h!$ times because of symmetry (all the permutations of the hidden layer nodes). Finding the global minimum is unlikely. (Though since E is suboptimal, the global minimum is unlikely to be the best one.)
2) If the nodes of the network are sigmoidal, the Jacobian of E with respect to weights tends to be ill-conditioned [43], causing problems with the convergence of the optimization process. Some of the weights can change by large amounts without changing E much.
3) Near a local minimum of E, the local dimension is likely to be much smaller than N_i. A fully connected MLP is too large, but only locally. This also causes problems with the convergence of the optimization process. Since different regions of the network are active for different input samples, we can't just reduce N_i or N_h.
4) The data are often inherently low precision. In our OCR example, the 7480 training samples are binary images of digits; averages over the training set can take on at most 7480 distinct values, and thus are known only to $\log_2 7480$ bits, about 13.
5) The training data are sparse in the input space, which for discussion may have dimension as high as 128. In our OCR example, we have approximately 2^{13} training samples, so that $2^{13} \ll 2^{128}$. Our input space is woefully undersampled.
6) The training data, and therefore the boundaries of the classes in the N_i-dimensional input space, are more cloudlike than ball-like.

B. Modifications to Network Optimization

There are two time scales associated with the dynamics of neural network training. The shorter time scale is associated with the calculation of feedback signals within an iteration (the "inner iterations") and the longer time scale is associated with the sequence of optimization iterations (the "outer" iterations). The dynamics can be used to analyze the sequence of dynamic systems generated as the optimization iteration proceeds.

On the short time scale, in any single iteration the structure of the network is fixed. The dynamics involves the application of the feature vectors as driving inputs. Since the feature vectors provide a forcing function, this is not an equilibrium system, but a problem where the training data drives the network to different stationary states, with error-correcting feedback to alter the weights to bring the network to a minimum in the recognition error.

On the longer time scale is the optimization iteration process. At each time step of the outer iteration the connectivity of the network is changed by Boltzmann pruning. On this time scale structural stability is the dominant dynamic process. The goal of the outer iteration process is to approach a steady state where the effect of network changes minimizes the feedback error over the training set and thus an overall steady state is achieved.

The link between the study of time-dependent network performance and nonlinear network optimization is provided by the analysis of structural stability. We looked directly at the qualitative properties expected in systems of this kind [44]–[46] and altered the training procedure to take the expected dynamic behavior into account. This analysis used the dynamical systems approach to provide us with qualitative information about the phase portrait of the system during training rather than a statistical representation of the weight space of the MLP network.

We effectively reduce the dimension of the problem using the center manifold approach [47]. This approach is similar to the Lyapunov–Schmidt technique [48] which reduces the dimension of the system from n to the dimension of the center manifold, which in numerical calculations is equal to the number of calculable nonzero eigenvalues of the Jacobian. Since the number of weights in the typical network is approximately 10^4 and the number of bits in the feature data is approximately 13, direct numerical methods for calculation of the eigenvalues from the linearized dynamics are very poorly conditioned. The center manifold method has the advantage over the Lyapunov method in that the reduced problem still is a dynamical system with the same dynamic properties as the original system. The reduction in dimension is implemented using the Boltzmann machine in coordination with the SCG learning algorithm.

The reduced problem after application of the center manifold method is still an SCG system. This approach can reduce the error independent of the content of the particular sample distribution and the size of training data. The result is a saving in training time and improvement in performance without analysis of those network components which make minimal contributions to the learning process.

Based on insights obtained from system dynamics, three constraints were integrated into the SCG optimization. Each of these modifications alters the dynamics of the training process in a way that simplifies the form of the decision surfaces, which as stated earlier, have a global dimension of about 100 with a local dimension of about 10.5. Understanding the topology of this space is useful for developing improved training methods based on dynamics. All of these modifications take place in the inner loop of the SCG optimization.

1) Weight Regularization: The first constraint is weight regularization, which has been shown to act as a smoother when the NN training process is treated as an approximation problem [49]. Regularization decreases the volume of weight space used in the optimization process by adding an error term which is proportional to the sum of the squares of the weights. The effect is to create a parabolic term in the error function that is centered on the origin, reducing the average magnitude of the weights. A scheduled sequence of regularization values is used that starts with high regularization and decreases until no further change in the form of the error-reject curve is detected. At present this sequence is manually terminated but automatic termination using a validation set is possible. Constraining the network weights reduces the number of bits in the weights and therefore the amount of information contained in the network, allowing a simplification in the network structure.

For our OCR problem, as stated earlier, the local feature dimension of features is about 10.5, and the large networks used here have 96 KL features. That is, the training examples sample well a feature subspace of dimension 10.5. Full sampling, at least one point in every hyper-quadrant of the feature space would require 2^{96} samples. The undersampling increases roughly as the power of the difference in the global and the local dimension. For the OCR problem, this is $96 - 10.5$ so that the undersampling increases at least as fast as the volume calculated from the feature space radius. The large difference in local and global dimensionality indicates that adding a practical number of training examples (that might expand the feature set volume somewhat but would not alter the local or global dimensionality of the feature set) is unlikely to produce a significant improvement in network performance. Rather than feebly attempting to fill the ever-expanding volume with new training examples, we apply regularization to constrain the weight space.

2) Sine Activation: The second insight we can obtain from dynamics is that, as we expand the Jacobian within the optimization, any very small nonlinear terms will be unimportant compared to the noise in real training data. For the data used here, the KL transform normalizes the input signals to a uniform dynamic range; the NN is not needed to perform this task. The calculation of the KL transform involves, for the OCR problem, calculation of a mean image value for every location in the image. Since the values used to calculate the mean are binary, the mean for an N-example training set can only take on N values and is only known to $\log_2 N$ bits. Even with perfect training data, the KL features have this finite precision. Ambiguous characters, or characters that are well formed but which have noisy edge pixels, add additional uncertainty, giving uncertainty throughout the dynamic calculation. Small changes in outputs in network dynamics caused by training data noise are indistinguishable from small changes caused by real but small dynamic terms.

The usual form for the activation function for NN's is a sigmoidal or logistic function. This function has small derivatives of all orders for large positive or negative values of the input signal, since all derivatives of the activation function approach zero exponentially. Therefore, the Jacobian of the inner iteration is singular within computational error [43]. When the Jacobian has large numbers of near-zero eigenvalues, the optimization is dominated by center manifold dynamics [44], [47]. Changing the activation function to a sinusoidal function creates a significant change in the dynamics of the training since the activation function has significant derivatives for all possible input signals. Using sinusoidal activation functions improves network training dynamics and results in better error-reject performance and simpler networks. The resulting sinusoidal-based networks train in a comparable amount of time and are similar in size to those trained using sigmoidal activation functions.

3) Boltzmann Weight Pruning: The third insight provided by dynamic stability considerations involves the dynamics of small weights near the origin in weight space. In weight space, the stability analysis of these variables involves small real eigenvalues which will be dominated by weight oscillations. The dynamics of these weights are associated with dynamic processes that have small real parts and are therefore near the center manifold. For the OCR problem, for example, the dynamics near the center manifold result partly from an attempt by the optimization process to make fine character distinctions on large signals well above the noise level and partly from small weights interacting with the noise. Since these two effects are indistinguishable and create very complex dynamics, it is better to set the near-zero weights to zero, forcing these dynamic processes out of the training.

The design and implementation of most NN architectures are based on an analysis of the size and content of the network training data. In character classification applications, the training sets are large and the local and global ranks of the feature data are very different. The complex structure of the training data requires large networks, in the order of 10^4 weights and 10^2 nodes. If the training process is treated as a dynamic system with the weights as independent variables, the Jacobian would have at least 10^8 terms.

Boltzmann methods have been used as a statistical method for combinatorial optimization and for the design of learning algorithms [50], [51]. This method can be used in conjunction with a supervised learning method to dynamically reduce the size of these large networks [52], whereas other methods requiring the direct analysis of the weights and/or their topology are numerically intractable. The strategy used in this research is to remove the weights using Boltzmann criteria during the training process. Information content is used as a measure of network complexity for evaluation of the resulting network.

In the earlier work [52], the SCG method is used as a starting network for the Boltzmann weight pruning algorithm. The initial network is fully connected. The pruning is carried out by selecting a normalized temperature, T, and removing weights based on a probability of removal

$$P_i = \exp\left(-w_i^2/T\right). \tag{1}$$

The values of P_i are compared to a set of uniformly distributed random numbers on the interval $[0, 1]$. If the probability P_i is greater than its random number, w_i is set to zero. The process is carried out for each iteration of the SCG optimization process and is dynamic. If a weight is removed it may subsequently be restored by the SCG algorithm; the restored weight may survive if it has sufficient magnitude in subsequent iterations.

Boltzmann pruning has two effects on the training process. First, it takes small dynamic components that have small real eigenvalues near the center manifold and places them on the center manifold. This simplifies training dynamics by reducing weight space dimension, and removes initial local minima, thus avoiding the problems associated with dependence on random starting weights in a traditional SCG optimization. Second, Boltzmann pruning keeps the information content of the weights bounded at values which are equal to or less than the information content of the training set. When KL features are derived from binary images, the significance of the feature is no greater than the number of significant bits in the mean image value. Boltzmann pruning forces this constraint on the weights.

When Boltzmann pruning was used in [52], detailed annealing schedules were used to insure convergence of the training process. When regularization is combined with pruning, an annealing schedule is not needed and pruning can proceed concurrently with the regularization process, starting at the beginning of the training process. The cost of pruning is only a small extra computation associated with the weight removal.

C. Training the New MLP

When the new MLP learning algorithm is invoked, it performs a sequence of steps. Each step does either training, or testing:

1) Training: A set of patterns is used to train (optimize) the weights of the network. Each pattern consists of a feature vector, along with either a class or a target vector. A feature vector is a tuple of floating-point numbers, which typically has been extracted from some natural object such as a handwritten character. A class denotes the actual class to which the object belongs, for example the character of which a handwritten mark is an instance. The network can be trained to become a classifier: it trains using a set of feature vectors extracted from objects of known classes. Or, more generally, the network can be trained to learn, again from example input-output pairs, a function whose output is a vector of floating-point numbers, rather than a class; if this is done, the network is a sort of interpolator or function-fitter. A training step finishes by writing the final values of the network weights as a file. It also produces a summary file showing various information about the step, and optionally produces a longer file that shows the results the final (trained) network produced for each individual pattern.

2) Testing: A set of patterns is sent through a network, after the network weights are read from a file. The output values, i.e. the hypothetical classes (for a classifier network) or the produced output vectors (for a fitter network), are compared with target classes or vectors, and the resulting error rate is computed. The program can produce a table showing the correct classification rate as a function of the rejection rate.

A good strategy for training the MLP on a new classification problem is to first work with a single training/testing session, surveying different combinations of parameter settings until a reasonable amount of training is achieved within the first 50 iterations, for example. This typically involves using a relatively high value for regularization and varying the number of hidden nodes in the network. For handprint character classification, the number of hidden neurodes should be equal to or greater than the number of input KL features.

Within this process of surveying parameters, it is also important to determine a value of temperature that causes a reasonable amount of Boltzmann weight pruning. Different levels of temperature are surveyed, typically incrementing or decrementing by powers of ten, until approximately 10% of the total number of weights are being pruned. The higher the temperature, the more severe the pruning. For handprint character recognition, a temperature of 10^{-4} works well.

Once reasonable training and pruning are achieved, these parameters should remain fixed, and successive sessions of training/testing are performed according to a schedule of decreasing regularization. For handprint character classification it works well to specify about 50 iterations for each training session, and to use a regularization factor schedule starting at 2.0 and decreasing to 1.0, 0.5, 0.2, 0.1, 0.01, and 0.001 for each successive training session. This process of multiple training/testing sessions initiates MLP training within a reasonable solution space, and enables the machine learning

to refine its solution so that convergence is achieved while maintaining a high level of generalization by controlling the dynamics of constructing well-behaved decision surfaces. The intermediate testing sessions allow one to evaluate the progress made on an independent testing set, so that a judgment can be made as to whether incremental gains in training have reached diminishing returns.

D. Comments

The insights gained from system dynamics combine to yield a training method which is smoothed on the exterior large, weight, region by regularization. At the same time the weights are pruned in the interior, small weight, region by Boltzmann pruning. The combination of these two methods greatly restricts the active region of weight space. Weight removal simplifies the problem by reducing the number of degrees of freedom. Restricting the range of weight values to a spherical shell near to but excluding the origin matches the significance of the network to the significance of the training data. In the region between constraints, dynamic stability is enhanced by choosing activation functions with nonzero derivatives throughout the space.

It would be interesting to do experiments which would allow the various modifications to the optimization strategies to be tested separately. In general, this has proved to be impossible because the methods only achieve convergence for the large networks used here when both pruning and regularization are used concurrently. The interaction between amount of regularization and the pruning temperature are such that changing the effective temperature over a range between $T = 10^{-3}$ to $T = 10^{-5}$ changes the number of network weights by a factor of 2.76, from 8653 to 3134, but does not significantly alter the reject error performance. Changing the minimum regularization factor from 2.0 to 0.01 typically reduces the zero reject error by a factor of two over the entire temperature range. These results are discussed in more detail in reference [53].

Changing the form of the activation function from sinusoidal to sigmoidal improved the reject error performance of the network by a factor of two at zero reject to a factor of four at the 15% reject level. Again the reader is referred to [53] for more details.

After this extended discussion, one may wonder if this type of recognition problem is unique to OCR problems. The authors would argue that it is not. The characters used in this work are normalized to 32×32 binary images containing 1024 pixels. Characters form a very small subset of all 2^{1024} possible images. The point of regularization is to confine the training dynamics to a region of weight space where the images are near the subset of images which are meaningful characters. At the other extreme, we argue that the subset of character images is not dense in feature space and that many noisy images (that are near character images but are not easily recognizable characters) are found in large training sets. The complex dynamics associated with the noisy images are pruned away by the Boltzmann process. These properties are not unique to character recognition. The difficulty of matching images to incomplete feature sets should not be surprising since patterns generated or identified by an image algebra [54] are a small subset of the patterns which could be generated with arbitrary generators and links in any image algebra.

IV. NIST Form-Based Handprint Recognition System

The new MLP described in Section III has been integrated into an end-to-end test-bed system at NIST. The NIST Form-Based Handprint Recognition System (referred to hereafter as the NIST system) has been designed to follow the general guidelines presented in [55]. This system was designed to process the HSF forms distributed in *NIST Special Database 19* (SD19) [56], and it is capable of processing every one of the 3669 forms in the data base.

A. The Application

Successful application of OCR technology requires more than system integration [55]. State-of-the-art solutions still require customization and tuning to the problem at hand. This being true, an operational system is largely defined by the details of the application.

The NIST system is a public domain software test-bed designed to read the handprinted characters written on Handwriting Sample Forms (HSF). An example image of a completed HSF form is shown in Fig. 3. This form was designed to collect a large sample of handwriting to support handprint recognition research. A CD-ROM named *NIST Special Database 19* (SD19), containing 3669 completed forms scanned in binary at 11.8 pixels per millimeter (300 pixels/in), is publicly available [56]. This data set also contains over 800 000 segmented and labeled character images from these forms.

The NIST system is designed to read all but the top line of fields on the form. The system processes the 28-digit fields and the randomly ordered lowercase and uppercase alphabet fields along with the handprinted paragraph of the Preamble to the U.S. Constitution at the bottom of the form.

B. System Components

Fig. 4 contains a diagram that illustrates the organization of the components of the NIST system. Generally speaking, each of these components has many possible algorithmic solutions. Therefore, the NIST system is designed in a modular fashion so that different methods can be evaluated and compared within the context of an end-to-end system.

A considerable amount of processing must take place in order to reliably isolate the handprint on a form. The front end of the system (from Load Form Image to Segment Text Line(s)) is comprised of a sequence of complex algorithms based primarily on image processing and statistical techniques. Once isolated, segmented character images are normalized. This is typically viewed as a preprocessing task for the NN classifier. The NIST system applies the KL transform to convert character images into feature vectors of floating point numbers. These feature vectors are classified using the new NIST MLP. The last step, spell-correction of the

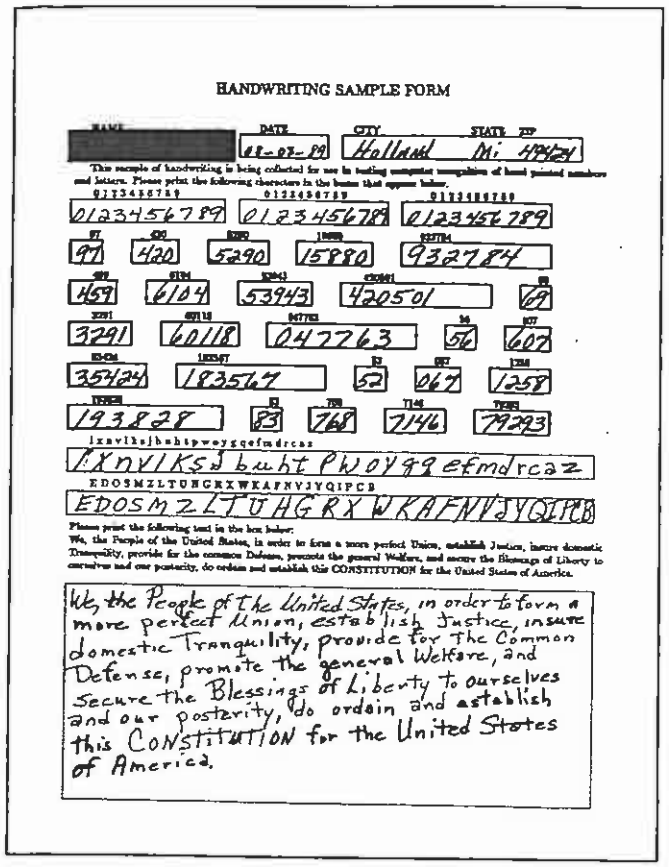

Fig. 3. Example HSF form from NIST Special Database 19.

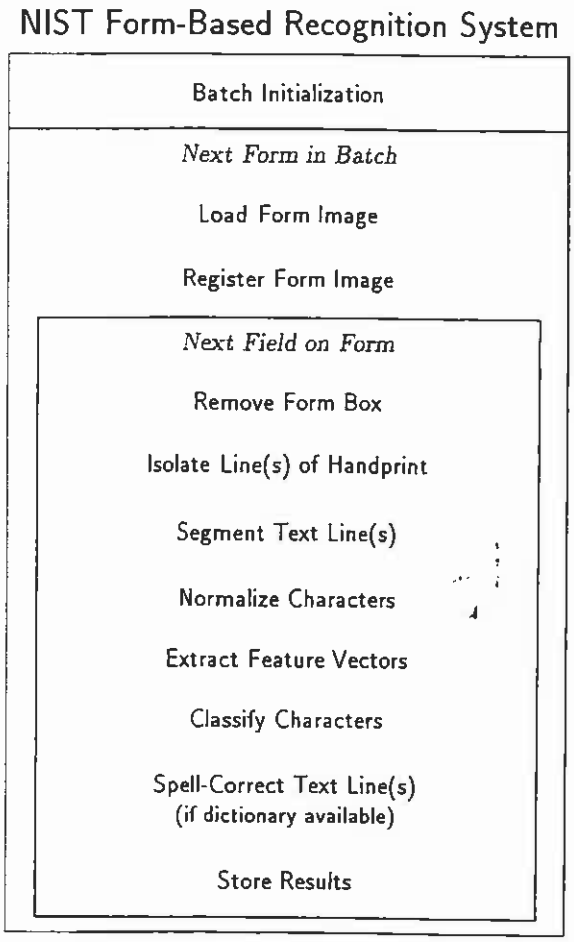

Fig. 4. Organization of functional components within the NIST system.

classified characters from the Preamble paragraph, falls under postprocessing.

1) Batch Initialization: The NIST system is a noninteractive batch processing system designed to process one or more images of completed HSF forms with each invocation. This first step loads all the precomputed items required to process a particular type of form (in this case HSF forms). These items include basis functions used for feature extraction and NN weights for classification. There are four types of fields on the HSF form: numeric, lowercase, uppercase, and the Preamble paragraph. Each type of field requires a separate set of basis functions and NN weights.

2) Load Form Image: For each form in the batch, the NIST system reads a CCITT Group 4 compressed binary raster image from a file on disk, decompresses the image in software, and passes the bitmap along with its attributes on to subsequent components.

3) Register Form Image: The form must be registered or aligned so that the fields in the image correspond to a prototypical template of field coordinates. The NIST system uses a generalized method of form registration that automatically estimates the amount of rotation and translation in the image without any detailed knowledge of the form [57]. By using this general registration technique, new form types can be trained automatically. A prototypical form is scanned, its rotational distortion is automatically measured and removed, and the positions of dominant lines are stored for future registrations.

4) Remove Form Box: After registration, coordinates of each field are used to extract field subimages. Given a field subimage, the black pixels corresponding to the handwriting must be separated from the black pixels corresponding to the form. This is a difficult task as a black pixel can represent handwriting, the form, or the overlap of both. As all the fields on the HSF form are represented by boxes, the NIST system uses a general algorithm that locates the box within the field subimage, and intelligently removes the sides so as to preserve overlapping characters [58].

5) Isolate Line(s) of Handprint: Line isolation is difficult for multiple-line responses such as the handprinted paragraph of the Preamble at the bottom of the HSF form. There are no lines provided in this paragraph box to guide the writer, nor are there any instructions as to how many words should be written on a line. As a result, the baselines of the writing significantly fluctuate, which makes tracking the lines of handprint difficult.

The NIST system uses a bottom-up approach to isolating the lines of handprint within a paragraph. The technique starts by decomposing the paragraph into a set of connected components. Each component is represented by a point at its geometric center. To reconstruct the handprinted lines of text, a nearest neighbor search is performed left-to-right and top-to-bottom through the two-dimensional (2-D) pattern of points [59]. Having located piecewise-linear trajectories of the text, the tops and bottoms of linked components are interpolated

and smoothed forming line bands. These bands form a spatial map, and all the components in the image are sorted into their respective lines in correct reading order according to their overlap and/or proximity to these bands [60].

6) Segment Text Lines: The segmentation method used in the NIST system is image-based and does not use the over-segmentation or integrated recognition techniques. Connected components are used as first-order approximations to single and complete characters. Building on the utility of connected components, the segmentation method uses a simple adaptive model of writing style [61]. Using this model, fragmented characters are reconstructed, multiple characters are split, and noise components are identified and discarded. Visual features are measured (the width of the pen stroke and the heights of the characters) and used by fuzzy rules, making the method robust. The segmentor performs best when applied to single-line responses, and even better when the fields are numeric.

7) Normalize Characters: The recognition technique used by the NIST system falls under the category of feature-based NN pattern classification. The segmented character images vary greatly in size, slant, and shape. The segmented character images are size-normalized by scaling the image either up or down so that the character tightly fits within a 20×32 pixel region. The stroke width is also normalized using simple morphology. Image normalization is performed to deal with the size and slant of writing, leaving to the recognition process primarily the task of differentiating characters by variation in shape. Upon normalization, each character is centered in a 32×32 pixel image.

8) Extract Feature Vectors: At this point, each character is represented by 1024 binary pixel values. The KL transform is applied to these binary pixel vectors in order to reduce dimensionality, suppress noise, and produce optimally compact features (in terms of variance) for classification [12].

A training set of normalized character images is used to compute a covariance matrix which is diagonalized using standard linear algebra routines, producing eigenvalues and corresponding eigenvectors. This computation is expensive, but is done once off-line, and the top n ranked eigenvectors are stored as basis functions and used subsequently for feature extraction. Feature vectors of length 128 are used in the NIST system; each coefficient in the vector is the dot product of an eigenvector with the (1024-pixel vector of the character being classified less the global mean vector).

9) Classify Characters: Once segmented characters are represented by feature vectors, a host of pattern classification techniques can be applied. We have used the training method described in Section III that produces MLP networks with performance equal to or better than PNN for character recognition [53]. This is achieved with a single three-layer network by making fundamental changes in the network optimization strategy as discussed in Section III. A network of size $128 \times 128 \times 10$ was used to classify digits, and networks of size $128 \times 128 \times 26$ were used to classify alphabetic characters. On handprinted digit classification problems, these modifications improve error-reject performance by factors between two and four and reduce network size (with pruned weights set to zero) by 40–60%.

To classify a character, the appropriate eigenvectors (or basis functions) and MLP weight matrices must be loaded into memory. As mentioned earlier, this is accomplished during batch initialization. Using the eigenvectors, the normalized image is transformed into a feature vector. The feature vector is then presented to the MLP network. The result is an assigned classification along with a confidence value.

10) Spell-Correct Text Line(s): The only field on the HSF form that has any linguistic information is the Preamble field. The Preamble contains 38 distinct words all of which are included in a dictionary. Words are parsed from each line of raw classifications by applying the dictionary as described in [59]. This process identifies words within the character stream while simultaneously compensating for errors due to wrong segmentations and classifications. The limited size of the dictionary helps offset the burden placed on this process.

11) Store Results: For the numeric and randomly ordered alphabet fields, the NIST system outputs for each segmented character an assigned class and its associated confidence as determined by the MLP classifier. When processing the Preamble paragraph, the system produces a sequence of spell-corrected words as output.

V. Evaluation of Neural Network-Based OCR Systems

This section presents NN-based results from the First OCR Systems Conference and from the NIST Form-Based Handprint Recognition System.

It is important to understand that the accuracy of the trained NN OCR system will be strongly dependent on both the size and the quality of the training data. Many common test examples used to demonstrate the properties of pattern recognition systems contain on the order of 10^2 examples. These examples show the basic characteristics of the system but provide only an approximate idea of the system accuracy. As an example, the first version of an OCR system was built at NIST using 1024 characters for training and testing. This system has an accuracy of 94%. As the sample size was increased the accuracy initially dropped as more difficult cases were included. As the test and training sample reached 10 000 characters the accuracy began to slowly improve. The poorest accuracy achieved was with sample sizes near 10^4 and was 85%. The sample sizes discussed in this paper are well below the 10^5 character sample size which we have estimated is necessary to saturate the learning process of the NIST system [12].

A. NN Results from First OCR Systems Conference

The task of this conference was to recognize isolated handprinted characters reporting both an assigned class and a confidence value. The test was comprised of images from 58 646 digits, 11 941 uppercase letters, and 12 000 lowercase characters. For the purposes of discussion, we will focus on results from only the digits [62].

Table I lists 11 of the 40 different systems used in the conference for recognizing digits. The table is divided into three part: systems using both NN-based feature extraction and classification, systems using non-NN types of feature

TABLE I
METHODS OF FEATURE EXTRACTION AND CLASSIFICATION USED IN FIRST OCR SYSTEMS CONFERENCE

System	Features	Classification
	Neural Net	
ATT_2	receptor fields	MLP
Hughes_1	neocognitron	
Nestor	neocognitron	MLP
Symbus	raw	self-Org. NN
	Hybrid	
ERIM_1	morphological	MLP
Kodak_2	Gabor	MLP
NYNEX	model	MLP
NIST_4	K-L	PNN
	Non Neural Net	
Think_1	template	distance maps
UBOL	rule based	KNN
Elsagb_1	shape func.	KNN

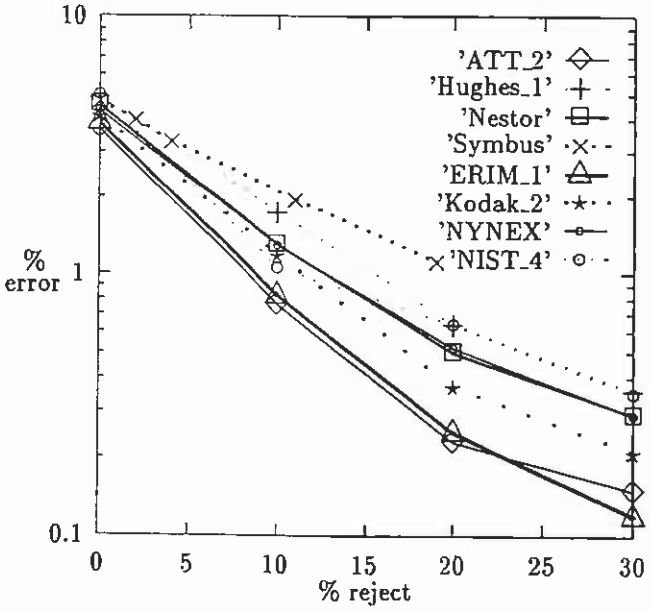

Fig. 5. Rejection versus error rates for NN-based systems.

extraction in conjunction with a NN-based classifier, and systems that are strictly non-NN.

Fig. 5 plots the error reject curves from digit recognition results for the first eight NN-based systems listed in the table. Fig. 6 contains results from the three non-NN systems listed in Table I.

There is no significant difference between the results of the three non-NN systems. The results from the NN systems however vary greatly. One can conclude from these graphs that the statistical techniques are somewhat constrained to the same solution space, while it is possible to utilize NN's to obtain superior results. On the other hand, the variation in NN results demonstrates that one cannot expect to simply apply any NN paradigm and get satisfactory results. Careful consideration must be given to the method of feature extraction, the NN architecture, and the learning paradigm to be applied. It is interesting to note that all of the best performing systems used

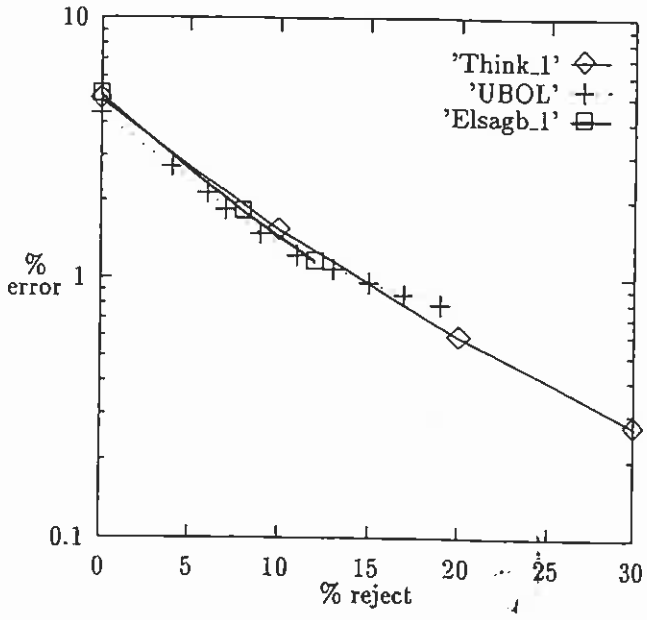

Fig. 6. Rejection versus error rates for non-NN systems.

MLP-based NN's. These observations are consistent with the experiments conducted by NIST when developing the new MLP training algorithm described in Section III.

For example, in an independent study [53], a common set of training and testing feature vectors were used to compare PNN to two different MLP-base digit classifiers. Both MLP's were trained using the techniques discussed in Section III. The MLP's differed in the type of activation transfer function they each used. The first MLP was trained using traditional sigmoidal transfer functions, while the second MLP was trained using sinusoidal functions. In the study, KL features from 7480 training and 23 140 testing image of digits were used. These error reject curves are plotted in Fig. 7. By applying sinusoidal transfer functions, the performance of the MLP is enhanced to where it performs better than PNN.

B. NIST System's Accuracies and Error Rates

This sections evaluates the performance of the NIST Form-Based Handprint Recognition System. An end-to-end system for handprint OCR has correlated modules. The results presented at the two Census Systems Conferences demonstrate the difficulty of deducing end-to-end system performance from component characteristics even when the same organization designed and constructed the components. This is the case because interactions between components are complex, highly nonlinear, nonlocal, noncommutative, and nonadditive [55].

In order to compile statistics on accuracy and error rates, the NIST system was run across the forms in SD19 and recognition results were stored to files. Recognition system classifications were stored in *hypothesis* files, and their associated confidence values were stored in *confidence files*. These files were then processed using the NIST Scoring Package [63], and performance statistics were compiled at the character, field, and word levels.

The results presented in this section comprise one of the largest published experiments of its kind, and it is reproducible

TABLE II
HSFSYS2 ACCURACIES AND ERROR RATES FOR DIGIT FIELDS ACROSS SD19. PART (A) REPORTS CHARACTER-LEVEL STATISTICS AND (B) REPORTS FIELD-LEVEL STATISTICS

	hsf_0	hsf_1	hsf_2	hsf_3	hsf_4	hsf_6†	hsf_7	hsf_8	Total
(A) Characters									
Correct	96.6%	96.5%	96.1%	97.2%	93.7%	97.3%	96.3%	96.9%	96.3%
	62772	62731	62486	75804	60933	63144	62615	8817	459302
Substituted	2.8%	2.9%	3.0%	2.4%	5.5%	2.1%	3.0%	2.5%	3.1%
	1816	1871	1972	1891	3575	1367	1920	229	14641
Inserted	0.7%	0.7%	0.7%	0.7%	0.8%	0.5%	0.6%	0.3%	0.7%
	454	487	425	571	489	318	422	25	3191
Deleted	0.6%	0.6%	0.8%	0.4%	0.8%	0.6%	0.7%	0.6%	0.6%
	412	398	542	305	492	359	465	54	3027
Total	65000	65000	65000	78000	65000	64870	65000	9100	476970
(B) Fields									
Correct	86.3%	86.7%	86.2%	88.6%	77.5%	89.6%	86.5%	88.2%	86.0%
	12084	12139	12068	14881	10855	12517	12105	1728	88377
Total	14000	14000	14000	16800	14000	13972	14000	1960	102732

†*Segmented character images from the writers in this partition were used to train the neural network classifiers.*

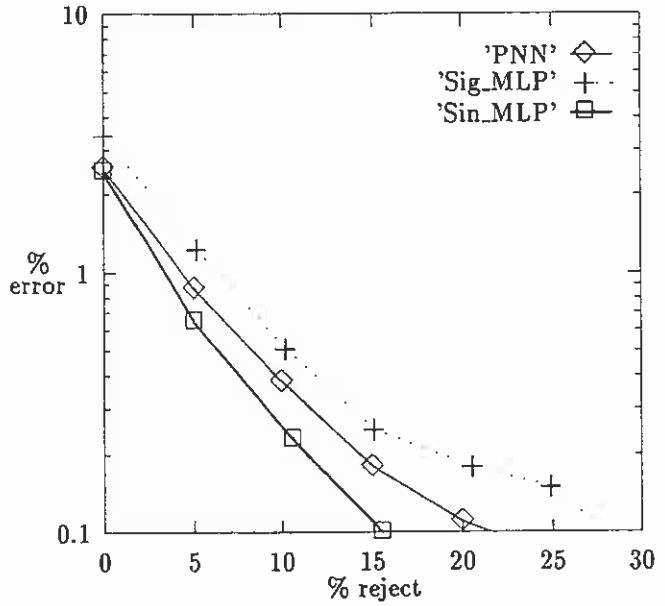

Fig. 7. Error reject graph for PNN, sigmoidal MLP, and sinusoidal MLP for classification of digits.

by purchasing the SD19 data base from NIST. In all, sample handwriting from 3669 writers was tested and a total of 109 200 words and 667 758 characters were recognized and scored. Training samples were extracted from specific writer partitions and used to train the new MLP-based character classifiers off-line. From the 667 758 characters, 109 719 were used in training. In the case of digits, the writers in hsf_6 (61,094 characters) were used in the training set, and in the case of upper and lowercase, writers in both hsf_4 and hsf_6 (totaling 24 420 uppercase characters and 24 205 lowercase characters) were used.

Table II lists the digit recognition results of running the new NIST system on all the forms in SD19. The system achieves a digit recognition accuracy of 96.3% and it recognizes 86% of the digit fields entirely correctly. The system is capable of registering every form in SD19, with only ten fields rejected due to poor image quality. The system recognizes uppercase characters with an accuracy of about 90%, lowercase characters at 80%, and the words in the Preamble paragraph at 64%.

Notice that the accuracies and error rates reported in Table II were compiled from the system processing the entire set of forms in SD19. This is our only handprint character database, so characters from a few of the database's partitions were required to train the MLP classifier off-line. Specifically, the partitions hsf_4 and hsf_6 were used for training as described earlier.

Comparing the system's results on hsf_6 to other partitions, it is interesting to see that the inclusion of hsf_6 in the classifier training does have a small influence. With digits, the system is 97.3% correct on hsf_6 whereas the results on hsf_3 are almost as good at 97.2%, and the other partitions (with the exception of hsf_4) range between 96% and 97%. The writers in hsf_4 are from a different population and are known to be statistically more difficult to recognize [64]. These small differences (particularly for the digits) demonstrate that the MLP character classifier is doing a reasonably good job at generalizing on writers it has not seen during its off-line training.

1) Rejection versus Error Rate: Using a machine for OCR in many ways complements the performance of humans. Machines are very efficient in doing tasks that are primarily repetitive and reflexive, whereas humans quickly fatigue under these conditions. Humans, on the other hand, are very adept at performing tasks requiring higher-level reasoning, and as a result they provide more robust but much slower solutions to complex problems. Accounting for these differences, successful recognition systems allow the machine to perform the bulk of the work, and on an exception basis, humans are used to resolve ambiguities and potential errors. This is accomplished through rejection mechanisms that automatically route low-confidence machine decisions to humans for verification.

This section compares the rejection ability of the new NIST recognition system to to its older counterpart described in

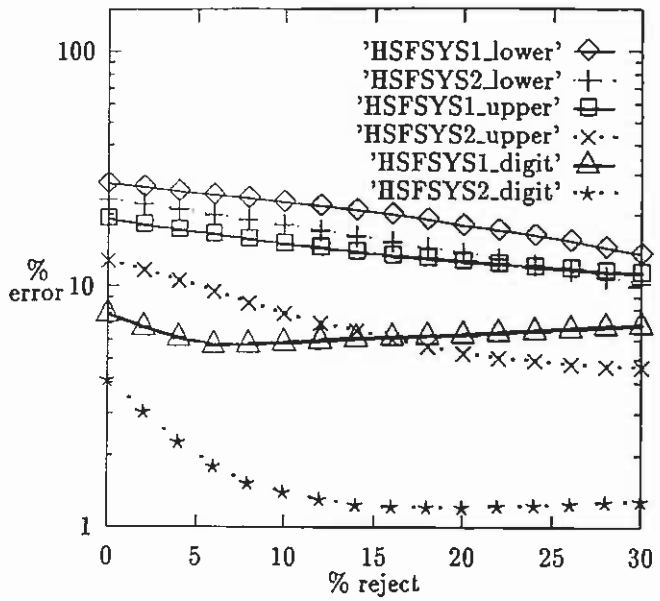

Fig. 8. Rejection versus error rates for digit, upper, and lowercase recognition between HSFSYS1 and HSFSYS2.

[13]. The older system is named HSFSYS1 and uses an optimized PNN to classify characters. The new NIST system is named HSFSYS2, and it not only uses the new MLP neural network for classification, but it employs a new writer-adaptive segmentor based on connected components [61].

The graph in Fig. 8 plots error versus rejection rates with error plotted on a logarithmic scale. Results are shown for both HSFSYS1 and the new system HSFSYS2, and the results are broken out by digit, upper, and lowercase recognition. In general, as the number of rejected character classifications increases, the error rate on the remaining accepted (or nonrejected) classifications decreases, and accuracy improves. Also, the impact of rejection on accuracy tapers off as more and more characters are rejected. In the figure, the bottom two curves represent the performance of the new and old systems on recognizing characters in the numeric fields on the HSF forms. With no rejection, HSFSYS2 has an error rate near 4%, and HSFSYS1 has an error rate over 7.5%. As the number of rejected digit classifications is increased, the error rate proceeds to drop, but HSFSYS2 falls at a significantly faster rate than does HSFSYS1. The difference in the slope of the two digit curves confirms the robustness of the MLP classifier used in HSFSYS2 over the PNN classifier used in HSFSYS1. The digit error rate of HSFSYS2 continues to drop to nearly 1.2% at 15% rejection. One concludes from these results, that in terms of recognizing numeric fields, the new NIST recognition system has half the error-rate of the original system for a moderate rejection level.

The differences between the two systems is less dramatic with uppercase and lowercase recognition. The middle two curves in Fig. 8 correspond to the results of recognizing the uppercase alphabet fields on the HSF forms. The HSFSYS2 curve does fall off slightly faster than does HSFSYS1's, but the distance between the curves is not as large as that of the digit curves. With no rejection, HSFSYS2 has an error rate of almost 13% and HSFSYS1 just over 19%. The two lowercase curves are even closer to each other, and they maintain pretty much the same relative distance across the range of rejections plotted. This emphasizes that lowercase recognition is still the most difficult for the NIST systems. The small increase in separation between these two curves as rejection increases can be attributed to a combination of two factors. First, the decision surfaces trained within the MLP classifier for lowercase are much more complex than those of uppercase, and the decision surfaces for uppercase are more complex than those of digits [53]. Second, the challenges remaining in the system that are affecting accuracy lie primarily in components other than the classifier. Otherwise, the relative slopes in the upper and lowercase curves would more closely resemble those of the digit classifications.

VI. Summary and Conclusions

In this paper, we have discussed NN handprint character recognition in considerable detail. The NIST public domain system has been presented and, since the source code is available, even more detailed analysis of algorithm performance is possible.

Analysis of NIST, ERIM, MCC, and other systems submitted to the two OCR systems conferences allow us to draw several important conclusions. NN's are the most widely used and most accurate methods of handprint character recognition. Previous work has shown that PNN's were the most accurate of the NN methods tested but with proper training MLP's are as accurate and have better rejection characteristics.

Since 1991 when the first OCR systems conference was held, major improvements in OCR performance have been achieved. The NIST version 2 public domain OCR system can achieve an average character recognition accuracy, including all error sources (such as segmentation) that is comparable to the best character accuracy achieved on isolated characters in 1991. These results were obtained using an integrated end-to-end testing method. The NIST version 2 system was tested on all of Special Database 19. This is the largest public test of handprint OCR done to date and shows that, with proper form design and system integration, economically useful rates of recognition are possible using only public domain algorithms.

We also conclude that with existing algorithms the accuracy of NN OCR systems is more dependent on overall system design and NN training set size than on the type of NN algorithm used.

References

[1] K. Han and I. K. Sethi, "Off-line cursive handwriting segmentation," in *Proc. Third Int. Conf. Document Analysis and Recognition*, Aug. 1995, pp. 894–896.

[2] A. Negi, K. S. Swaroop, and A. Agarwal, "A correspondence based approach to segmentation of cursive words," in *Proc. Third Int. Conf. Document Analysis and Recognition*, Montreal, P.Q., Canada, Aug. 1995, pp. 1034–1037.

[3] J. T. Favata and S. N. Srihari, "Off line recognition of handwritten cursive words," in *Proc. Conf. Machine Vision Applications in Character Recognition and Industrial Inspection*, San Jose, CA, SPIE, 1992, vol. 1661, pp. 224–234.

[4] S. N. Srihari, "Name and address block reader system for tax form processing," in *Proc. Third Int. Conf. Document Analysis and Recognition*, Montreal, P.Q., Canada, 1995, pp. 5–10.
[5] R. A. Wilkinson *et al.*, "The first optical character recognition systems conference," Tech. Rep. NISTIR 4912, Nat. Inst. Standards Technol., Aug. 1992.
[6] J. Geist *et al.*, "The second census optical character recognition systems conference," Tech. Rep. NISTIR 5452, Nat. Inst. Standards Technol., May 1994.
[7] J. S. Denker, *et al.*, "Neural network recognizer for hand-written zip code digits," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 323–331.
[8] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems*, vol. 2, D. S. Touretzky, Ed. Morgan Kaufman, 1990, pp. 598–605.
[9] G. L. Martin and J. A. Pittman, "Recognizing hand-printed letters and digits using backpropagation learning," *Neural Comput.*, vol. 3, pp. 258–267, Summer 1991.
[10] G. Seni, R. K. Srihari, and N. Nasrabadi, "Large vocabulary recognition of on-line handwritten cursive words," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 757–762, 1996.
[11] O. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition: A survey," *Pattern Recognit.*, vol. 29, pp. 641–662, 1996.
[12] P. J. Grother, "Karhunen Loéve feature extraction for neural handwritten character recognition," in *Proc. Conf. Applications of Artificial Neural Networks III*, Orlando, FL, Apr. 1992, vol. 1709, pp. 155–166.
[13] M. D. Garris *et al.*, "NIST form-based handprint recognition system," Tech. Rep. NISTIR 5469, Nat. Inst. Standards Technol., July 1994.
[14] M. D. Garris, R. A. Wilkinson, and C. L. Wilson, "Analysis of a biologically motivated neural network for character recognition," in *Proc. Conf. Analysis of Neural Network Applications*, May 1991, pp. 160–175.
[15] A. Shustorovich, "A subspace projection approach to feature extraction: The two-dimensional Gabor transform for character recognition," *Neural Networks*, vol. 7, pp. 1295–1301, 1994.
[16] G. E. Hinton and C. K. I. Williams, "Adaptive elastic models for hand-printed character recognition," in *Advances in Neural Information Processing Systems*, vol. 4, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. Morgan Kaufmann, 1992, pp. 512–519.
[17] C. K. I. Williams, M. D. Revow, and G. E. Hinton, "Using a neural net to instantiate a deformable model," in *Advances in Neural Information Processing Systems*, vol. 7, G. Tesauro, D. S. Touretzky, and T. Leen, Eds. Morgan Kaufmann, 1995, pp. 965–972.
[18] *CEDAR CDROM 1. Handwritten Cities, States, ZIP Codes, Digits, and Alphabetic Characters Database*, United States Postal Service Office of Advanced Technology, 1992.
[19] I. Guyon, V. N. Vapnick, B. E. Boser, L. Y. Botton, and S. A. Solla, "Structural risk minimization for character recognition," in *Advances in Neural Information Processing Systems*, J. E. Moody, S. J. Hanson, and R. P. Lippmann Eds. San Mateo, CA: Morgan Kauffman, vol. 4, pp. 471–479, 1992.
[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 318–362.
[21] Y. LeCun, *et al.*, "Back propagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, pp. 541–551, 1989.
[22] H. Schwenk and M. Milgram, "Transformation invariant autoassociation with application to handwritten character recognition," in *Advances in Neural Information Processing Systems*, vol. 7, G. Tesauro, D. S. Touretzky, and T. Leen, Eds. San Mateo, CA: Morgan Kaufmann, 1995, pp. 991–1006.
[23] P. Simard, Y. Le Cun, and J. S. Denker, "Efficient pattern recognition using a new transformation distance," in *Advances in Neural Information Processing Systems*, vol. 5, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds. San Mateo, CA: Morgan Kaufmann, 1993, pp. 50–58.
[24] O. Matan, *et al.*, "Multi-digit recognition using a space displacement neural network," in *Advances in Neural Information Processing Systems*, vol. 4, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA: Morgan Kaufmann, 1992, pp. 488–495.
[25] J. D. Keeler, D. E. Rumelhart, and W. K. Leow, "Integrated segmentation and recognition of hand-printed numerals," in *Advances in Neural Information Processing Systems*, vol. 3, R. P. Lippmann, J. E. Moody, and D. S. Touretzky, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 557–563.
[26] J. D. Keeler and D. E. Rumelhart, "A self-organizing integrated segmentation and recognition neural net," in *Advances in Neural Information Processing Systems*, vol. 4, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA: Morgan Kaufmann, 1992, pp. 496–503.
[27] S. Haykin, *Neural Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[28] S. W. Lee and E. J. Lee, "Integrated segmentation and recognition of connected handwritten characters with recurrent neural network," in *Proc. Third Int. Conf. Document Analysis and Recognition*, Montreal, P.Q., Canada, Aug. 1995, pp. 413–416.
[29] D. J. Hepp, "Recognition of handprinted and cursive words by finding feature correspondences," in *Proc. Conf. Document Recognition*, San Jose, CA, 1994, pp. 47–58.
[30] M. J. Ganzberger *et al.*, "Matching database records to handwritten text," in *Conf. Document Recognition*, San Jose, CA, 1994, vol. 2181, pp. 66–75.
[31] G. L. Martin, "Centered-object integrated segmentation and recognition of overlapping handprinted characters," *Neural Comput.*, vol. 5, pp. 419–429, May 1993.
[32] G. L. Martin and J. Talley, "Recognizing handwritten phrases from U.S. Census forms by combining neural networks and dynamic programming," *J. Artif. Neural Networks*, vol. 2, pp. 167–194, 1995.
[33] A. Shustorovich and C. W. Thrasher, "Neural network positioning and classification of handwritten characters," *Neural Networks*, vol. 6, pp. 685–693, 1996.
[34] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: MIT Press, 1979.
[35] G. L. Martin, "Centered-object integrated segmentation and recognition for visual character recognition," in *Advances in Neural Information Processing Systems*, vol. 4, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA: Morgan Kaufmann, 1992, pp. 504–511.
[36] K. Fukushima, "Connected character recognition with a neural networks," in *Proc. Int. Conf. Document Analysis and Recognition*, Montreal, P.Q., Canada, Aug. 1995, pp. 240–242.
[37] H. Drucker, R. Schapire, and P. Simard, "Boosting performance in neural networks," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, pp. 705–719, 1993.
[38] J. L. Blue, *et al.*, "Evaluation of pattern classifiers for fingerprint and OCR applications," *Pattern Recognit.*, vol. 27, pp. 485–501, 1994.
[39] C. L. Wilson, P. J. Grother, and C. S. Barnes, "Binary decision clustering for neural network based optical character recognition," *Pattern Recognit.*, vol. 29, pp. 425–437, 1996.
[40] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.
[41] L. K. Hansen, C. Liisberg, and P. Salamon, "The error-reject tradeoff," computer reprint, 1995.
[42] J. L. Blue and P. J. Grother, "Training feed forward networks using conjugate gradients," in *Proc. Conf. Character Recognition and Digitizer Technologies*, San Jose, CA, Feb. 1992, pp. 179–190.
[43] S. Saarinen, R. Bramley, and G. Cybenko, "Ill-conditioning in neural network training problems," *SIAM J. Sci. Comput.*, vol. 14, pp. 693–714, 1993.
[44] J. Carr, *Applications of Centre Manifold Theory*. Berlin, Germany: Springer-Verlag, 1981.
[45] M. W. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra*. New York: Academic, 1974.
[46] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Berlin, Germany: Springer-Verlag, 1983.
[47] J. Sijbrand, "Properties of center manifolds," *Trans. Amer. Math. Soc.*, vol. 289, pp. 431–469, June 1985.
[48] S. N. Chow and J. K. Hale, *Methods of Bifurcation Theory*. Berlin, Germany: Springer-Verlag, 1982.
[49] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Comput.*, vol. 7, pp. 219–269, 1995.
[50] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Botlzmann machines," *Cogn. Sci.*, vol. 9, pp. 147–169, 1985.
[51] F. L. Alt, "Digital pattern recognition by moments," in *Optical Character Recognition*, G. L. Fischer *et al.*, Eds. McGreger & Werner, 1962, pp. 159–179.
[52] O. M. Omidvar and C. L. Wilson, "Information content in neural net optimization," *J. Connection Sci.*, vol. 6, pp. 91–103, 1993.
[53] C. L. Wilson, J. L. Blue, and O. M. Omidvar, "Training dynamics and neural network performance," *Neural Networks*, vol. 10, pp. 907–923, 1997.
[54] U. Grenander, *General Pattern Theory*. Oxford, U.K.: Oxford Univ. Press, 1993.
[55] C. L. Wilson, *et al.*, "Design, integration, and evaluation of form-based handprint and OCR systems," Tech. Rep. NISTIR 5932, Nat. Inst.

Standards Technol., Dec. 1996.
[56] P. J. Grother, "Handprinted forms and characters database," Tech. Rep. Special Database 19, HFCD, Nat. Inst. Standards Technol., Mar. 1995.
[57] M. D. Garris and P. J. Grother, "Generalized form registration using structure-based techniques," in *Proc. Fifth Ann. Symp. Document Analysis and Information Retrieval*, Las Vegas, NV, Apr. 1996, pp. 321–334.
[58] M. D. Garris, "Intelligent form removal with character stroke preservation," in *Proc. Conf. Document Recognition III*, San Jose, CA, Feb. 1996, pp. 321–332.
[59] ——, "Unconstrained handprint recognition using a limited lexicon," *Tech. Rep. NISTIR 5310*, Nat. Inst. Standards Technol., 1993.
[60] ——, "Teaching computers to read handprinted paragraphs," Tech. Rep. NISTIR 5894, Nat. Inst. Standards Technol., Sept. 1996.
[61] ——, "Component-based handprint segmentation using adaptive writing style model," Tech. Rep. NISTIR 5843, Nat. Inst. Standards Technol., June 1996.
[62] C. L. Wilson, "Effectiveness of feature and classifier algorithms in character recognition systems," in *Proc. Conf. Character Recognition Technologies*, D. P. D'Amato, Ed., San Jose, CA, 1993, vol. 1906.
[63] M. D. Garris and S. A. Janet, "NIST scoring package user's guide release 1.0," Tech. Rep. NISTIR 4950, Nat. Inst. Standards Technol., Oct. 1992.
[64] P. J. Grother, "Cross validation comparison of NIST OCR databases," in *Proc. Conf. Character Recognition Technologies*, D. P. D'Amato, Ed., San Jose, CA, 1993, pp. 296–307.

Michael D. Garris received the B.S. degree in computer science from Clarion University of Pennsylvania in 1986 and the M.S. degree in computer science with an emphasis on artificial intelligence, image processing, and artificial neural networks, from The Johns Hopkins University, Baltimore, MD, in 1991.

For nine of the last 11 years, he has been a Researcher at the Information Technology Laboratory's Visual Image Processing Group, National Institute of Standards and Technology (NIST), Gaithersburg, MD, developing and evaluating methods of handwriting recognition, automated forms processing, and document analysis and retrieval. He served as the principal integrator and has created numerous algorithms for the NIST public domain Form-Based Handprint Recognition System.

Charles L. Wilson (M'67–SM'85) has worked in various areas of computer modeling ranging from semiconductor device simulation (for which he received a Department of Commerce gold medal in 1983), and computer aided design, to neural network pattern recognition at Los Alamos National Laboratory, AT&T Bell Laboratories, and for the last 18 years at the National Institute of Standards and Technology, Gaithersburg, MD. He is currently the manager of the Information Technology Laboratory's Visual Image Processing Group. His current research interests are in application of nonlinear dynamics to neural network methods, pattern recognition methods for OCR, fingerprints, face recognition, image compression, digital video, and standards used to evaluate recognition systems.

James L. Blue received the A.B. degree in physics and mathematics from Occidental College, Los Angeles, CA, and the Ph.D. degree in physics from the California Institute of Technology, Pasadena.

He currently heads the Information the Technology Laboratory's Mathematical Modeling Group, National Institute of Standards and Technology, Gaithersburg, MD. Previously, he did research in semiconductor devices, mathematical modeling, and numerical methods at AT&T Bell Laboratories. His current interests include computer algorithms and simulation, micromagnetics, neural networks, and optoelectronics.