March 1992

Character Recognition Using A Modular Spatiotemporal Connectionist Model

Thomas Fontaine
University of Pennsylvania

Lokendra Shastri
University of Pennsylvania

Recommended Citation

Thomas Fontaine and Lokendra Shastri, "Character Recognition Using A Modular Spatiotemporal Connectionist Model", . March 1992.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-24.

Character Recognition Using A Modular Spatiotemporal Connectionist Model

Abstract

We describe a connectionist model for recognizing handprinted characters. Instead of treating the input as a static signal, the image is scanned over time and converted into a time-varying signal. The temporalized image is processed by a spatiotemporal connectionist network suitable for dealing with time-varying signals. The resulting system offers several attractive features, including shift-invariance and inherent retention of local spatial relationships along the temporalized axis, a reduction in the number of free parameters, and the ability to process images of arbitrary length.

Connectionist networks were chosen as they offer learnability, rapid recognition, and attractive commercial possibilities. A modular and structured approach was taken in order to simplify network construction, optimization and analysis.

Results on the task of handprinted digit recognition are among the best report to date on a set of real-world ZIP code digit images, provided by the United States Postal Service. The system achieved a 99.1% recognition rate on the training set and a 96.0% recognition rate on the test set with no rejections. A 99.0% recognition rate on the test set was achieved when 14.6% of the images were rejected.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-24.

Character Recognition Using A Modular Spatiotemporal Connectionist Model

MS-CIS-92-24
LINC LAB 219

Thomas Fontaine
Lokendra Shastri

University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

March 1992

Character Recognition Using A Modular Spatiotemporal Connectionist Model [*]

Thomas Fontaine and Lokendra Shastri
Computer and Information Science Department
University of Pennsylvania
Philadelphia, PA 19104-6389

Abstract

We describe a connectionist model for recognizing handprinted characters. Instead of treating the input as a static signal, the image is scanned over time and converted into a time-varying signal. The temporalized image is processed by a spatiotemporal connectionist network suitable for dealing with time-varying signals. The resulting system offers several attractive features, including shift-invariance and inherent retention of local spatial relationships along the temporalized axis, a reduction in the number of free parameters, and the ability to process images of arbitrary length.

Connectionist networks were chosen as they offer learnability, rapid recognition, and attractive commercial possibilities. A modular and structured approach was taken in order to simplify network construction, optimization and analysis.

Results on the task of handprinted digit recognition are among the best reported to date on a set of real-world ZIP code digit images, provided by the United Stated Postal Service. The system achieved a 99.1% recognition rate on the training set and a 96.0% recognition rate on the test set with no rejections. A 99.0% recognition rate on the test set was achieved when 14.6% of the images were rejected.

1 Introduction

The applications of a device capable of handprinted document recognition are numerous, touching on such diverse areas as office automation, postal sorting, and print-to-voice transcription devices for the blind [15][18]. A subtask of handprinted document recognition is the recognition of handprinted words, which in turn involves the recognition of isolated handprinted characters. A device capable of recognizing handprinted characters, however, has remained elusive despite a multitude of efforts spanning nearly four decades of research. The failure to develop a working solution to the problem can best be attributed to the excess of variance inherent in handprinted character samples. Mechanical differences, such as the stylus used, the writing surface, and the scanner employed can create dramatically different images. Intra-author factors such as age, mood, and purpose of writing can affect the printing of a character, as can inter-author differences such as writing style, left or right handedness, and skew of print.

Character recognition schemes typically operate upon *static* character images, whereby an image is presented to a system as a time-invariant signal. An alternative viewpoint is to consider an image to be a time-varying signal which is presented to a system in a piecewise fashion over time. For example, consider a discrete image, say M rows by N columns. One could envisage a column-wise scan of the image in which a system receives the M pixels of data contained in column i of the image at time i. Thus, the static image would be converted into a *spatiotemporal* signal that extends over N time steps.

[*] This work was supported by grant MCS-83-05211 and ARO grants DAA29-84-9-0027 and DAAL03-89-C-0031.

It was suggested by Shastri [17] that in some visual recognition domains there may be inherent advantages in considering images as *spatiotemporal* signals. First, this would offer shift-invariance along the temporalized dimension. Second, local spatial relationships in the image along the temporalized dimension would be inherently retained and would not have to be learned. Next, because such a model would use one spatial dimension and one temporal dimension, it would be architecturally less complex than a similar model using two spatial dimensions. Finally, the model would be capable of processing arbitrarily long inputs along the scanning dimension, and hence, could be extended to recognize handprinted words.

Our work has focused on the problem of real-world handprinted digit recognition. Our focus was motivated by several factors. First, although handprinted digit recognition is a subproblem of character recognition, the task is still quite difficult due to the variance present in handprinted digit samples. Second, a digit recognition system has many useful real-world applications, such as ZIP code recognition and check balance recognition. Finally, voluminous, real-world, handprinted digit databases are widely available.

The variance inherent in handprinted digit samples and the availability of large handprinted digit databases suggest the utilization of a system capable of learning from examples. Connectionist networks offer a suitable framework, since they can be optimized using a variety of well-studied methods, discovering salient features in large sets of data. Moreover, a trained network is capable of rapid recognition and can be implemented on a single microchip, offering attractive commercial possibilities. Several recent approaches employing connectionist networks to recognize handprinted digits have achieved good results on real-world images [4][11][12].

Although the connectionist paradigm is often associated with the modeling of cognitive processes, we chose to utilize connectionist networks because they are well suited for the task of handprinted digit recognition. Our network was developed with emphasis on structure and modularity, which allowed the incorporation of domain knowledge, a reduction in the number of free parameters, and the simplification of error analysis. Since spatiotemporal data representation necessitates working within a framework capable of processing time-varying signals, spatiotemporal connectionist models were employed.

Section 2 describes the temporalization of visual images and the utilization of spatiotemporal connectionist models. Section 3 provides a description of the digit database and preprocessing steps used, while Section 4 elaborates on the network developed to recognize handprinted digits. Results are reported in Section 5. Section 6 describes some avenues for future work and concluding remarks are found in Section 7.

2 The Spatiotemporal Connectionist Approach

2.1 Shift-Invariance

A system capable of recognizing a character in various images, independent of the spatial positioning of the character, is said to be a *shift-invariant* system. Developing shift-invariant connectionist recognition systems has proven difficult [9][14]. Connectionist networks operate with a fixed number of input units. In pixel-level image recognition, the number and arrangement of input units typically correspond to the number and arrangement of pixels in the input image. Since a given character may appear at different spatial locations in different images, the relevant data may be assimilated by different sets of input units. One can see the difficulty in deriving shift-invariance using such an approach, since the desired result is to recognize the character regardless of which of these sets of input units is receiving the character.

Our basic spatiotemporal scheme operating on an M by N image requires M input units. At time step i, column i of the image is assimilated by the input units. Thus, N time steps are necessary to complete this column-wise scanning, with output emanating from the output layer at each time step. If the network is optimized such that a column of zero inputs (white space) yields no significant output and does not significantly change the state of the network, then the network output, summed over time, is independent of the spatial position of the character along the temporalized axis. Thus shift-variance along the temporalized axis falls out as a natural byproduct of this method of data temporalization and assimilation.

In an initial experiment to determine the efficacy of our scheme, the problem of discriminating between two toy patterns was inspected. Two 3 by 3 binary patterns representing a T and a C were used in conjunction with a 10 by 10 field of interest, yielding a total of 64 possible placements of each pattern within the field of interest. Of these possible placements, 20 of each pattern type were chosen as training images. The experiment was repeated 4 times with random training examples and an average accuracy of 99.1% was achieved on the entire set of images.

The shift-invariant properties of the approach were then inspected using a more elaborate T-C discrimination problem [8]. Two prototypical 11 by 11 binary patterns, representing a T and C, were used in conjunction with a 32 by 32 field of interest, yielding a total of 484 placements per pattern. The training set was comprised of 60 randomly chosen placements of each pattern. After training, the network recognized 96% of the remaining 848 test images. The network's ability to recognize handprinted T and C samples was then informally tested and the network was found to have the ability to accurately perform discrimination regardless of character size, location, and style, to a large extent. Skew and stroke thickness were not handled well. A typical sample of test images which were correctly identified is shown in Figure 1.

Although the problem of recognizing patterns invariant to shift in two dimensions is interesting, in actual character recognition systems shift-invariance is easily produced by special purpose hardware designed to find the bounding box of a character. As we shall see, however, invariance along the temporalized dimension is desirable when the more general problem of word recognition is considered.

2.2 Retention of Local Spatial Relationships

Consider a unit in the first hidden layer of a traditional (static) network. The activations received by this unit from units in the input layer are unlabeled levels of activation, and hence, this unit cannot determine which inputs come from spatially neighboring pixels and which do not. As far as this hidden unit is concerned, the input it receives from an image I is indistinguishable from the input it receives from an image I' obtained by permuting I. Now consider a hidden unit in a spatiotemporal model. The inputs to such a unit from two adjacent pixels (along the temporalized dimension) become available to the unit in adjacent time steps. Consequently, the spatial structure of the input (along the temporalized dimension) is made explicit to the hidden unit.

2.3 Reduction in Network Complexity

In the spatiotemporal scheme, a spatial dimension is effectively exchanged for a temporal dimension. As such, one can expect a decrease in system throughput due to the extra time taken to assimilate the image. On the other hand, a substantial decrease in the complexity of the network is achieved.

Static connectionist networks, implemented completely in parallel, can perform classification in constant time. In utilizing a temporal dimension, $O(N)$ time steps are needed for the entire image to propagate through the network where N is the size of the spatial dimension of the image which is temporalized. It is important to note, however, that if the throughput of the classificatory device is not the bottleneck of the entire recognition system, then the extra time necessary to process a temporalized image may be inconsequential, unless N is large. Typically, character recognition systems operate on relatively small values of N, such as $N = 20$.

Given an M by M image, a number of input and hidden units of only $O(M)$ is needed in the spatiotemporal scheme, as opposed to $O(M^2)$ in the static case. The more relevant factor, however, is that only $O(M^2)$ links are necessary between layers in the completely connected spatiotemporal case, whereas $O(M^4)$ are required in the completely connected static case.[1] During network training, the number of links in a network

[1] The number of links would be $\propto kM^2$, where k is the number of delays used on links. Typically, $k \ll M$. For example, in the system described in this paper, $k = 3$, while $M = 20$.

correspond to the number of free parameters in an unconstrained nonlinear optimization. The reduction from $O(M^4)$ to $O(M^2)$ free parameters can dramatically decrease the dimensionality of the optimization.

2.4 Extension to Word Recognition

A common and often warranted criticism of the connectionist approach is that a network must have a fixed number of inputs, and thus must process images of a fixed size. With such a constraint, it seems very unlikely that a connectionist network could be developed to recognize word images. This criticism assumes, however, that *static* data is being processed. The utilization of the temporal dimension allows a connectionist network to operate on images of arbitrary size along the temporal dimension.

Consider, for example, a word image with M rows and N columns, where $N \gg M$, given a lengthy word. By sending the data contained in column i into the system at time i, a left-to-right assimilation of the image takes place. In a simple case where characters are separated by white space, this scheme could allow a connectionist network to recognize entire words simply by noting the sequence of output unit activation peaks over time, since white space between neighboring characters is ignored. Although real-world word images will not always have convenient white space between adjacent characters, the spatiotemporal approach does relax the restriction of fixed-size inputs, allowing for the possibility of progress towards word recognition (see Section 6).

2.5 Utilization of Spatiotemporal Connectionist Networks

Spatiotemporal data representation mandates working within a framework designed for the processing of time-varying signals. Several connectionist approaches utilizing a temporal dimension have been proposed [5][19][20][23]. The connectionist model employed in this work was inspired by the *Temporal Flow Model*, or TFM, proposed by Watrous who developed the model to address the problem of speech recognition and achieved good results [23].

The TFM is characterized by arbitrary link connectivity, as well as a propagation delay associated with each link. These features can be employed to provide a rich mechanism with which to process time-varying signals. Since a static (feed-forward) network produces output which is strictly an instantaneous function of its input, such a network is incapable of providing context sensitivity along the temporal dimension. In contrast, the TFM provides context sensitivity by allowing temporal integration of signals as the image is processed. In the models described in this paper, such integration is performed in two ways:

- Unit Recurrencies. Network recurrencies are restricted to only those in which a unit feeds back on itself via a recurrent link, so that the output of a unit at time t depends not only upon signals from other units, but also upon its own output at time $t-1$.

- Inter-layer integration via propagation delays. Consider two units, one of which sends output and the other which receives the signal. Suppose we employ two links between the units, the first with an associated delay of 1 time unit, and the second with a delay of 2 time units. A signal emanating at time t will reach the receiving unit along the link with a delay of 2 at the same time the signal at $t+1$ arrives, via the link with a delay of 1, thereby providing the desired temporal integration of signals.

One can generalize this integration via propagation delays by allowing an arbitrary number of links with associated delays between units. The utilization of a number of delay links effectively allows a unit to compute a function over a spatial window. For example, if M links are impinging upon a unit, with a propagation delay of i associated with the ith link, then the unit computes a function of M different time slices, which is equivalent to computing a function over a spatial window of width M.

2.6 Optimization of Spatiotemporal Connectionist Networks

In a connectionist framework, optimization is typically performed by viewing each weight on a link in a network as an unconstrained variable in an error minimization problem. Training samples are propagated through a network, and a measure of error is generated by the amount of dispersion between the actual network output and the desired network output. Minimization of this error maximizes the network performance on the training set.

Although simple search methods for optimization [1] and stochastic minimization methods [16] exist, both paradigms are typically not well suited for optimizing connectionist networks due to the excessive number of operations needed during minimization. Deterministic gradient methods for minimization (eg, see [6]) are applicable, provided that the gradient is computable. Once a scheme to compute the gradient of a multilayer network was popularized [14], it was demonstrated that many known nonlinear optimization techniques could easily be applied to connectionist networks [21]. All optimization experiments performed in preparation of this paper were performed using GRADSIM, a connectionist optimization package which offers several classical deterministic gradient descent algorithms [22].

Since feed-forward networks operate in a static fashion, the quantity to be minimized is an instantaneous error generated from the difference between the output of the network and the desired output, summed over all training examples. Spatiotemporal networks, however, operate by generating varying outputs over time. Hence, a target output is necessary at each time step and this target sequence is known as the *target function*. The development of a suitable set of target functions is a temporal credit assignment problem. Although the selection of target functions for a particular problem can be guided by domain knowledge (eg, estimated probability distributions of gray scale mass in visual images), and by human intuition, selection is primarily experimental. The set of target functions used in the development of our system is described in Section 4.4.

3 The Dataset and Preprocessing

One must be cautious when comparing recognition results achieved using different databases. A standard database from which individual researchers can test their respective systems is desirable. Ideally, such a database should be widely available, voluminous, the number of authors should approach the number of images, the authors should be from a diverse background, and the authors should be unaware that their printing will be used to test a recognition device. The "United States Postal Service Office of Advanced Technology Handwritten ZIP Code Database (1987)" is such a database of unconstrained handprinted digits. This database was made available to the authors by the Office of Advanced Technology, United States Postal Service. It contains thousands of handprinted ZIP codes, scanned from letters passing into the Buffalo, New York, Post Office, and provides a reasonable basis for comparison of handprinted digit recognition devices.

Although the test set was originally comprised of 616 ZIP code images, only 540 images were ultimately used.[2] 59 images in the original test set contained ZIP+4 codes, for example, and were not used simply for bookkeeping purposes. Another 15 images contained dark lines running across each image, and were not used, since the focus of research was not on the preprocessing of such images. Lastly, an image containing only 4 digits was not used, and another which was improperly coded was also discarded. To produce isolated digit images, each ZIP code image was broken down into five individual digits by hand. A linear slice was made between consecutive digits as fairly as possible, without removing stray marks or extended strokes. In total, 5,450 digit images comprised the training set, while the test set contained 2,700 digit images.

Preprocessing of digit images can greatly augment recognition performance by normalizing certain variations. After the ZIP code images were binarized and segmented by hand, a low pass filter was applied to each digit image to remove pepper noise. The skew of each digit was then normalized using a method suggested

[2] ZIP codes with serial numbers from bd_0001 to bd_1000 and from bd_1600 to bd_2000 were used for training, while ZIP codes with numbers between bd_2001 and bd_2636 were used for testing.

by Bakis [2] (whereby the XY moment about the centroid is forced to zero), and each digit was scaled to fit in a 20x20 bounding box using a simple nearest neighbor method [10] (such that the aspect ratio of the image was preserved). Finally, the SPTA method of skeletonization [13] was employed to remove variation caused by differing thicknesses of writing styli and image quantizations. Examples of three binarized digit images before and after preprocessing are shown in Figure 2.

4 The Handprinted Digit Recognition System

With a connectionist approach, it is tempting to utilize a minimally structured network, relying on the power of optimization techniques to produce a suitable result. For relatively small or toy problems, such an approach may lead to reasonable success. In problem domains such as handprinted digit recognition which require networks of a larger scale, the number of free parameters becomes large. This generally precludes the possibility of starting with a minimally structured network and deriving a network which possesses the desired level of generalization. In view of this, we have developed our recognition system in a structured and modular manner.

4.1 Modularity

Our digit recognition system is comprised of ten individually trained Single Digit Recognition Networks, each of which is responsible for the detection of a particular digit. Each Single Digit Recognition Network consists of four Single Scan Networks, each of which assimilates data from a different "scan" of the image. A Single Scan Network is constructed from a number of adaptable connectionist layers, operating in conjunction with a number of pretrained Feature Detection Modules. A Feature Detection Module is in turn formed by the replication and tessellation of a pretrained Local Receptive Field.

A modular approach offers several advantages. By utilizing pretrained feature detectors, knowledge is incorporated into the system without increasing the dimensionality of the optimization. Network construction is simplified, since components can be trained on easier subproblems. Error analysis is also simplified, since errors occurring in a component can often be attributed to the improper functioning of one or more subcomponents. We attribute the success of our system to this modular approach.

4.2 Pretrained Spatiotemporal Feature Detection Modules

Certain simple characteristic features are inherent in many pattern recognition domains. Apropos to the problem of digit recognition, many of the Arabic numerals can be approximately written using four simple stylus strokes: horizontal, vertical, slash, and backslash. The simplicity and recurrence of these strokes suggests the utility of developing pretrained feature detection modules to recognize these features, which can then be integrated into a larger network (explicit extraction of these features dates back to at least the fifties, eg, see [3]).

4.2.1 The Local Receptive Field (LRF)

Since the prescence of horizontal, vertical, and diagonal strokes were considered to be the most relevant and easily measurable features to employ in digit recognition, a separate *Local Receptive Field* detection module, or LRF, was pretrained to detect each of these four features.

The generic LRF module is seen in Figure 3. It receives input over a spatial field of 4 inputs, a temporal field of 4 time steps, and consists of 4 input units, 4 hidden units, and a single output unit. Hidden unit n receives information from all input units, and utilizes n links from each input unit, with respective delays of $1, 2, \ldots, n$, effectively creating a spatial window of width n into the temporal signal. For example, the portion of the image which hidden unit 3 is allowed to view is shaded in Figure 3. The output unit receives signals from each of the hidden units via links with associated time delays of 1, and each unit utilizes a self-recurrent link, although not shown in the figure. As long as a feature to be detected by an LRF is present in its 4 by 4 receptive field, the LRF will emanate an output signal, albeit with a slight lag.

Each specific LRF to detect horizontal, vertical, slash, and backslash strokes was trained using the same generic LRF module, and thus different LRFs to detect different features vary only by the weights on the links, and not by the LRF topology. To demonstrate the training methodology, we present the process by which the "slash" LRF detector was produced.

- Generation of all possible 4 by 4 binary image yields 2^{16}, or 65,536 images. Of these, 10,000 were randomly chosen as a training set, and a number of "prototypical" slash images, constructed by hand, were added to this set to ensure proper concept formulation.
- Heuristics were developed to score the "goodness of slash" present in a 4 by 4 image, yielding a score between 0 and 1 for each image. Heuristic scores were based upon the XY moment about the centroid of the image, the connectivity of bits along the slash direction, and the number of bits present. The heuristics were tested extensively and invariably returned a score accordant with the score the authors would have assigned.
- The heuristics were used to automatically generate a target function for each training example. Each example was overlayed in the center of a 10 by 4 image of off-bits. The target function value for time t was derived by applying the heuristic scoring routine to the portion of the image comprised of rows t through $t+3$.
- The LRF network was optimized on the 10,000 images and their respective target functions using the BFGS optimization method (see [6] for example), until a low error was achieved.

Informal testing of all four LRF modules revealed that they were accurate and robust, performing as intended in all cases inspected.

4.2.2 The Feature Detection Module (FDM)

LRFs act locally. A well known technique for extending local feature detection devices to act upon larger fields of interest is to simply replicate them, and tessellate them as desired (see [9][14], for example). We refer to a group of identical and tessellated LRFs as a *Feature Detection Module*, or FDM, or sometimes simply as a *feature detector*. An example of an FDM using 3 LRFs, with an input unit *overlap* of 2 and covering a receptive field of 8 inputs, is seen in Figure 4.

Using FDMs, a particular topological feature to be detected which is present in a given image will essentially be converted into a number of LRF unit outputs. Spatial information along one dimension is retained since a feature detector is comprised of several spatially differing LRFs, each with its own output. Spatial information along the other dimension is encoded by the temporal sequence of firings of the LRFs.

A desirable trait of the feature detectors is their modularity. Each feature detector is composed from an LRF building block in a simple manner, and the number of useful feature detectors is limited only by the number of useful LRFs which can be developed. At a different level of modularity, the feature detection modules may easily be inserted into a network design. During optimization, the FDMs are masked out and not considered part of the optimization. This allows the incorporation of robust feature detectors which yield useful information without increasing the dimensionality of the optimization.

4.3 A Single Scan Network (SSN)

In the spatiotemporal approach discussed to this point, data from column i of the image is assimilated by the network at time i, effectively producing a column-wise scan of the image. One could, however, easily employ a variety of other directional scans, such as a row-wise scan in which row i is input into the system at time i. Given a feature detector to detect a feature with respect to a given scan direction, it is interesting to note that the same feature detector can be used to detect a rotation of the feature with respect to a different scan. For example, a horizontal stroke detector with respect to a column-wise scan can be used as a vertical stroke detector in conjunction with a row-wise scan.

Consider scanning an M by M image of an isolated digit using a left-to-right column-wise scan. Although important discriminatory information may be present in the rightmost columns of the image, this information is not detected by the network until the final time steps. It may be useful to employ multiple scans in a variety of directions, where each scan feeds information into a separate group of input units. The cost of additional scans, of course, is the increase in architectural complexity. Our digit recognition system employs a row-wise scan, a column-wise scan, a reverse-row-wise scan, and a reverse-column-wise scan of the image.

For each scanning direction, we utilized a *Single Scan Network*, depicted in Figure 5. Two pretrained feature detection modules, a horizontal and slash stroke detector, were employed, along with several unstructured hidden layers. Dashed links were pretrained and did not vary during optimization. We now present some implementation details of our Single Scan Networks.

- Each Feature Detection Module contained 9 Local Receptive Fields, each with an overlap of two input units.

- The first elastic hidden layer used 9 hidden units, each receiving information from 4 adjacent input units, such that the receptive field of each unit had an overlap of 2 with each of its neighbors—the same scheme which was used in developing the FDMs. Each hidden unit in the hidden layer, however, received signals from a given input unit via 3 links with associated delays of 1, 3, and 5, creating a staggered spatial window into the temporal signal. Since adjacent time slices often contain redundant information, a sampling of the signal at every other time step allows important features to be detected, while keeping architectural complexity relatively low.

- The second hidden layer used two banks of 6 units. Each unit in a bank received information over a local field of width 4 from both pretrained feature detectors and the previous hidden layer. The receptive fields of adjacent hidden units had an overlap of 3. All links used a delay of 1, constraining the units to develop features by combining signals which emanated from the previous layers at approximately equal times.

- All 12 units from the second hidden layer impinged on the output unit via links with delays of 1, 3, and 5, again spatializing the temporal signals in a staggered fashion.

- Each hidden and output unit used a self-recurrent link.

- All variable links were assigned an initial random weight in the range of $(-0.05, 0.05)$.

- Units computed a sigmoidal function of their summed input, yielding an output bounded between 0 and 1.

4.4 A Single Digit Recognition Network

Information from each scan is processed independently and concurrently by the four SSNs, with the output of each SSN being passed to a final output layer consisting of a single output unit. We refer to this complete network as a Single Digit Recognition Network, depicted in Figure 6. Although each SSN could have been optimized independently before aggregation, we chose instead to optimize the Single Digit Recognition Network modules directly.

Each Single Digit Recognition Network was optimized to recognize a single digit class, and reject all others. The target function employed for a negative example was a constant function of 0.05 over time, while the target function of a positive example was sigmoidal, rising from a target of 0.05 to 0.95 by the end of the input. This scheme produces *pessimistic* recognition networks. A network will essentially remain inactive until some redeeming feature is discovered in the image which provokes confidence that the image belongs to the digit class represented by the network.

Each digit network was optimized separately using the BFGS algorithm (see [6], for example), until a mean squared error (MSE) of .002 was reached. MSE is calculated by summing errors generated by all image presentations, and normalizing for the number of images. The error for each image is taken to be the squared pointwise difference between network response and desired response, integrated over time, and then time-normalized. The "2" digit network optimization was creepingly slow after reaching an MSE of .0025 and was terminated due to time constraints.

4.5 The Complete Digit Recognition Network

After each of the ten Single Digit Recognition Networks was trained to recognize its respective digit, all networks were combined to produce the final handprinted digit recognition network. The aggregate network is comprised of 17,108 links (3,888 of which belong to pretrained feature detectors), and 1,290 units (360 of which belong to pretrained detectors).

4.6 Significance of Modularity

In retrospect, several levels of modularity can be seen in our system. Simple LRFs are replicated and tessellated to produce the Feature Detection Modules. FDMs, in conjunction with unstructured hidden layers, are combined in a hierarchically ordered fashion to yield a Single Scan Network, capable of analyzing the image with respect to a given scan. Several SSNs, one for each desired scan, are juxtaposed and connected to an output unit to derive a Single Digit Recognition Network, ultimately responsible for the detection of a particular digit class. After each of the ten Single Digit Recognition Networks are optimized to recognize their respective digit class, they are amassed to yield the final digit recognition network.

This modularly hierarchical approach allows us to incorporate domain knowledge into the network while simplifying optimization. In addition, the usage of single scan network modules and separate digit recognition modules facilitates error analysis. If a misclassification occurs, it can typically be attributed to lack of activation of the correct digit class network, or an excess of activation from an incorrect digit class network. Many of the digit recognition modules need not undergo inspection for a given classification error, since they performed in the desired manner. Typically, only one digit recognition network will have caused an error, resulting in a 90% reduction in the area of the network which needs to be analyzed. Analysis of the digit recognition network in error is also simplified, since the network is comprised of four SSN modules, each of which may be investigated separately.

5 Results

To test the accuracy of a Single Digit Network, we used a classification criterion based on hypothesizing that a pattern was a member or was not a member of the digit class represented by the network. Classification was made by choosing the hypothesis which yielded the smaller error, where the error was taken to be the distance of the network response to the (positive or negative) target function.

Table 1 shows the percentage of the 5,450 training images correctly classified by each separate digit classification network after training. The difference in percentages from row to row is due to testing of networks at different Mean Squared Errors, which are indexed in the first column. Table 2 is similarly indexed, but shows the accuracy of the networks upon presentation of the test set, comprised of 2,700 digit images which the networks had not been trained on. The performance on the test set, particularly at a MSE .002 level, compares favorably with the results of the training set, indicating a good level of generalization.

MSE	0	1	2	3	4	5	6	7	8	9
.002	99.9	99.6	na	99.9	99.8	99.9	99.8	100	99.7	99.8
.003	99.5	99.6	99.6	99.5	99.7	99.5	99.3	99.4	99.3	99.6
.004	99.1	99.6	99.1	99.2	99.3	99.4	98.9	98.8	98.7	99.3
.005	98.9	99.6	98.7	98.2	98.8	99.3	98.5	97.9	98.4	99.2
.006	98.8	99.6	98.4	98.0	98.8	98.8	98.5	97.9	98.0	98.0

Table 1: Individual Digit Network Training Results

MSE	0	1	2	3	4	5	6	7	8	9
.002	98.9	99.7	na	98.7	98.6	98.5	99.3	99.1	98.8	98.6
.003	98.9	99.7	98.1	98.3	98.5	98.8	98.9	98.9	98.7	97.1
.004	98.2	99.6	97.6	97.2	98.3	98.4	98.9	98.7	98.2	97.4
.005	98.0	99.5	97.1	97.1	98.2	98.2	98.5	98.8	97.7	96.5
.006	98.1	99.5	97.0	96.4	98.3	98.0	98.1	98.1	97.3	96.5

Table 2: Individual Digit Network Testing Results

After all ten separate digit recognition networks were combined to yield a single network with ten output units, a different classification criterion was needed. A basic winner-take-all scheme was chosen, in which the classification decision was made by choosing the class corresponding to the output unit which generated the highest time-normalized integrated activation.

Since it is often of great practical importance to assess the performance of a recognition system by deriving the percentage of test images that must be rejected as unclassifiable in order to force the error rate to 1% on the remaining images, a rejection criterion was also defined. Considering time-normalized integrated activation, let A_h be the highest activation of the ten output units, and let A_s be the second highest activation. We defined a measure of classification confidence, C, as:

$$C = \frac{1 - A_s}{1 - A_h} \tag{1}$$

Since $A_h, A_s \in (0,1)$, and $A_s \leq A_h$, we have $C \geq 1$. Clearly, a larger C indicates a more confident classification. Our rejection criterion was then defined such that for some $\epsilon > 0$, if $C < (1 + \epsilon)$, then the image was rejected as being unclassifiable.

Figure 8, depicts the output of our system, in response to a typical set of ten digit images. The plot was generated by GRADVIEW, a graphical interface to GRADSIM [22] and shows the output unit response, over time, upon assimilation of a set of test images.

Using the winner-take-all classification criterion described above, the final network achieved a recognition rate of 99.1% on the training set of 5,450 training images. On the test set of 2,700 images, an accuracy of 96.0% was obtained with no rejections. In order to force the network down to only a 1% error rate, 14.6% of the images needed to be rejected, with a rejection ϵ of 0.47. These results compare very favorably to other results which have been reported using samples drawn from the same database, as seen in Table 3.

It should be noted that no network postprocessing was performed. Further optimizations can be made, and the network can be massaged to increase performance in a number of ways. Even without postprocessing, however, classification results clearly validate the ability of spatiotemporal connectionist networks to perform well in the domain of handprinted digit recognition.

Author	Year	Raw Recognition	Rejections to 1% Error
Denker, et al [4]	1989	94%	14%
Le Cun, et al [11]	1990	95.4%	9%
Fontaine and Shastri	1992	96.0%	14.6%

Table 3: Recent Results on the USPS ZIP Code Database

6 Future Work: Word Recognition

The spatiotemporal approach may have other very useful applications which have not been fully explored. Shift-invariance along the temporalized dimension is a byproduct of the approach, and is certainly an interesting area for future work. Temporalizing a spatial dimension also allows for processing of images of arbitrary size along the temporalized dimension. A natural application of this feature is word recognition. We are investigating word recognition by utilizing a hybrid model consisting of two spatiotemporal connectionist networks governed by a procedural controller. The first spatiotemporal connectionist network, dubbed the Coarse Recognition Device, is responsible for coarsely estimating segmentation boundaries between characters. A second network, called the Refined Recognition Device, is very similar to the handprinted digit network which was described in this paper. It is responsible for accepting or rejecting estimations made by the Coarse Recognition Device by attempting to classify portions of the image as characters. Both networks are governed by a traditional procedural controller, capable of fusing signals emanating from the two connectionist networks while incorporating systematic domain knowledge. We are inspecting the application of such a scheme to the problem of handprinted ZIP code recognition [7].

7 Concluding Remarks

We have presented an alternative approach to handprinted character recognition in which an image is processed by a spatiotemporal connectionist network over time.

Connectionist networks are elastic, offer expeditious recognition after training, and possess the attractive feature of being implementable on a single microchip. By employing spatiotemporal networks to process temporalized images, shift-invariance and retention of local spatial relationships along the temporalized dimension is achieved, while yielding a significant reduction in network complexity over static connectionist networks. Perhaps the greatest advantage of image temporalization, however, is that the size of the input data is no longer restricted along the dimension being temporalized, allowing for the possibility of extending the model to word recognition.

Modularity and structure were stressed in the development of our model in order to incorporate domain knowledge in the form of pretrained feature detectors, reduce the number of free parameters, and simplify error analysis. We applied our model to a the problem of handprinted digit recognition, and achieved good results on a difficult, real-world database. Further improvements can be made.

We feel that the spatiotemporal approach to visual pattern recognition can be of great utility in certain domains and offers several interesting avenues for future work.

Acknowledgements

The authors wish to thank the United States Postal Service, Gary Herring of the Office of Advanced Technology, and John Hull of SUNY Buffalo for providing us with the ZIP code database.

References

[1] P. R. Adby and M. A. H. Dempster. *Introduction to Optimization Methods.* London: Chapman and Hall, 1974.

[2] R. Bakis, N. Herbst, and G. Nagy. An experimental study of machine recognition of hand-printed numerals. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):119–132, 1968.

[3] J. S. Bomba. Alpha-numeric character recognition using local operators. In *Proceedings of the East Joint Computer Conference*, pages 218–224, 1959.

[4] J. S. Denker, W. R. Gardner, H. P. Graf, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, H. S. Baird, and I. Guyon. Neural network recognizer for hand-written zip code digits. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1, pages 323–331. Morgan Kaufmann, 1989.

[5] J. L. Elman. Finding structure in time. Technical Report CRL 8801, University of California, San Diego, Center for Research in Language, 1988.

[6] R. Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, 1980.

[7] T. Fontaine. *A Hybrid Procedural-Connectionist Word Recognition System.* Thesis Proposal, University of Pennsylvania, 1991.

[8] T. Fontaine and L. Shastri. Spatiotemporal connectionist character recognition, May 1990. Working Paper.

[9] K. Fukushima, S. Miyake, and T. Ito. Neocognitron: a neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 13:826–834, 1983.

[10] H. S. Hou. *Digital Document Processing.* John Wiley and Sons, 1983.

[11] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 396–404. Morgan Kaufmann, 1990.

[12] G. Martin and J. Pittman. Recognizing hand-printed letters and digits. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 405–414. Morgan Kaufmann, 1990.

[13] N. J. Naccache and R. Shinghal. SPTA: a proposed algorithm for thinning binary patterns. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume SMC-14, pages 409–418, 1984.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representation by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. Cambridge, MA: MIT Press, 1986.

[15] J. Schürmann. Reading machines. In *Proceedings of the International Conference on Pattern Recognition*, pages 1031–1044, 1982.

[16] G. E. Hinton T. J. Sejnowski. Learning and relearning in Boltzmann machines. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 282–317. Cambridge, MA: MIT Press, 1986.

[17] L. Shastri. Personal communication, April 1989.

[18] C. Suen, M. Berthod, and S. Mori. Automatic recognition of handprinted characters—the state of the art. *Proceedings of the IEEE*, 68:469–487, 1980.

[19] R. S. Sutton. Learning to predict by the methods of temporal differences. In *Machine Learning 3*, pages 9–44. Boston: Kluwer, 1988.

[20] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 328–339, 1989.

[21] R. Watrous. Learning algorithms for connectionist networks: applied gradient methods for nonlinear optimization. Technical Report MS-CIS-87-51, University of Pennsylvania, June 1987.

[22] R. Watrous. GRADSIM: a connectionist network simulator using gradient optimization techniques. Technical Report MS-CIS-88-16, University of Pennsylvania, March 1988.

[23] R. Watrous. *Speech Recognition Using Connectionist Networks.* PhD thesis, University of Pennsylvania, 1988.

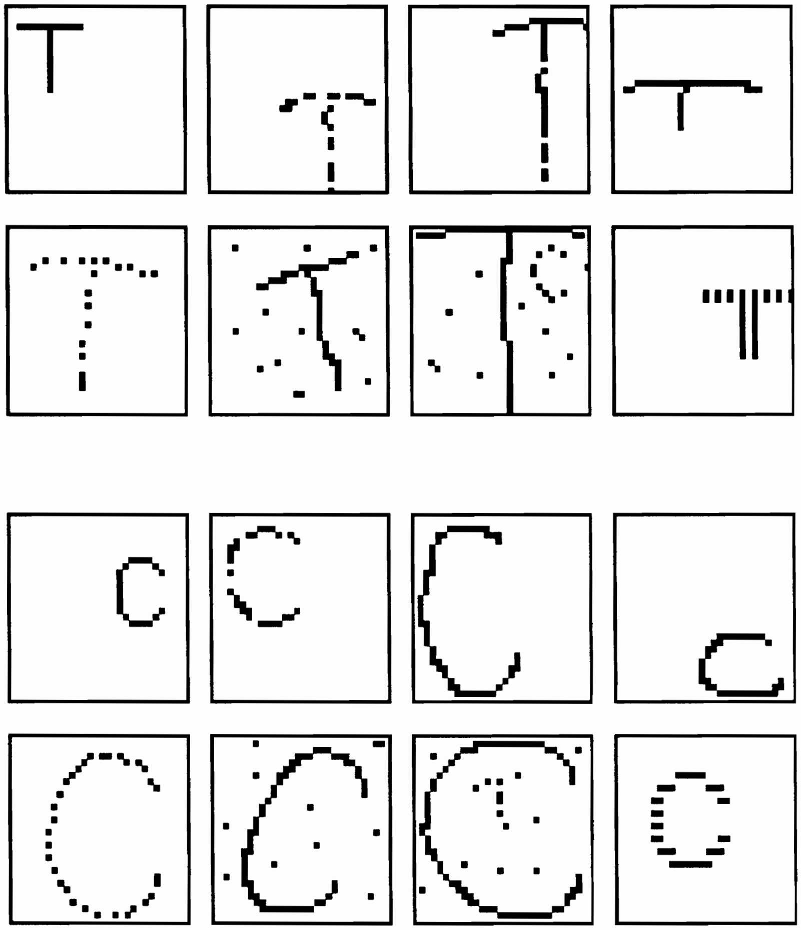

Figure 1: Some examples of T and C images correctly classified by an earlier spatiotemporal model.

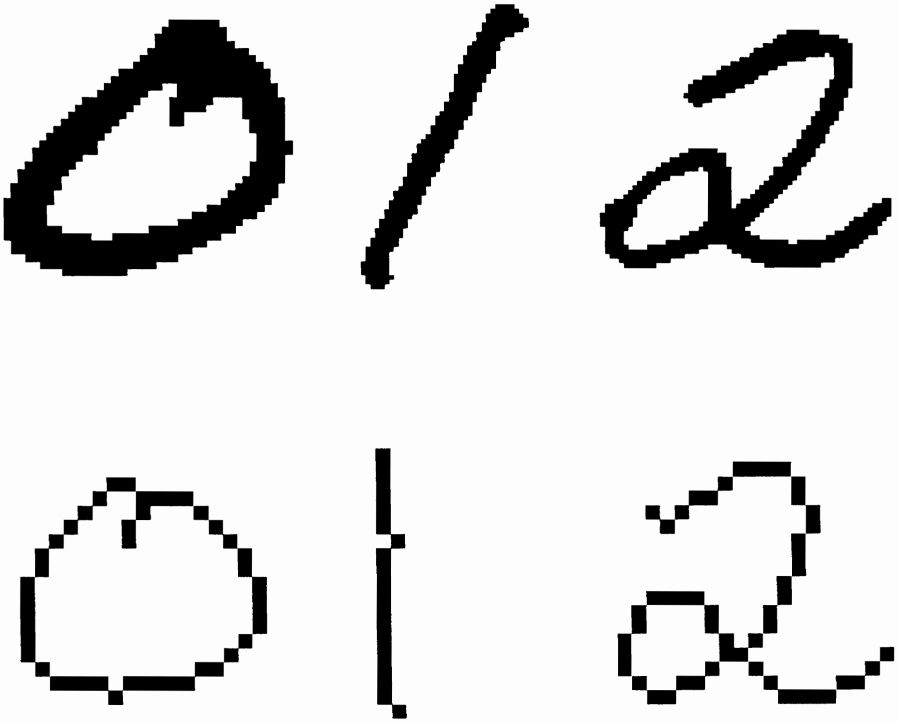

Figure 2: Some examples of binarized digit samples (top row), and their corresponding preprocessed versions after skew-normalization, scaling, and skeletonization.

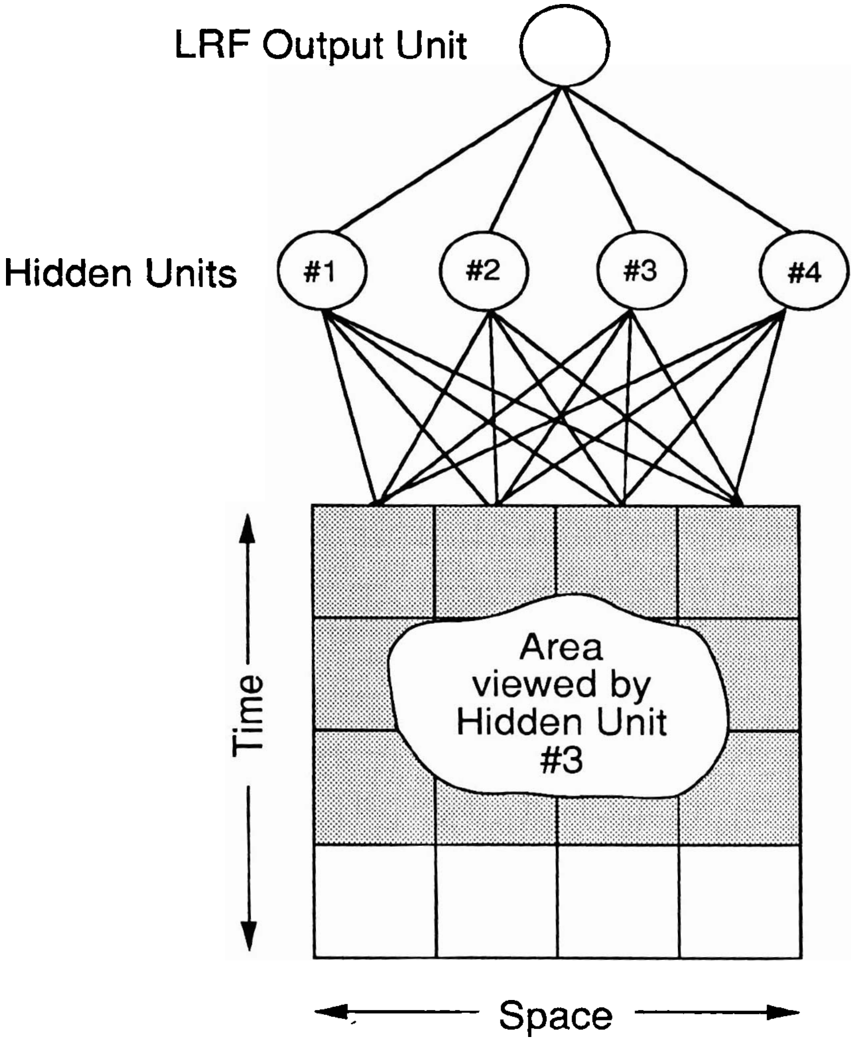

Figure 3: A generic Local Receptive Field (LRF) module. Hidden Unit n is able to view a spatial field of width 4, and a temporal window of width n. In effect, the temporal signal is spatialized, and the output unit computes a function of a purely spatial field. Unit #3, for example, is allowed to view the shaded portion of the image.

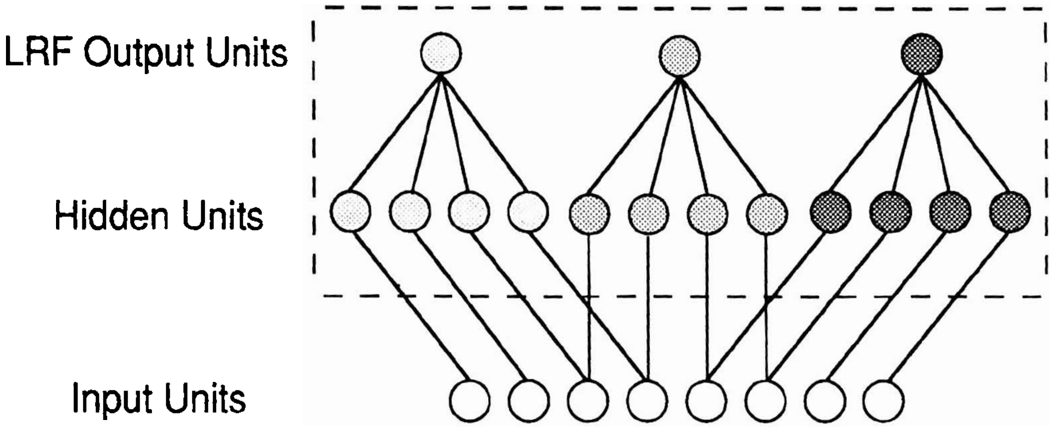

Figure 4: A generic Feature Detection Module (FDM). The FDM is composed of 3 LRFs, tessellated such that adjacent LRFs share 2 input units. The dashed box demarcates the entire FDM.

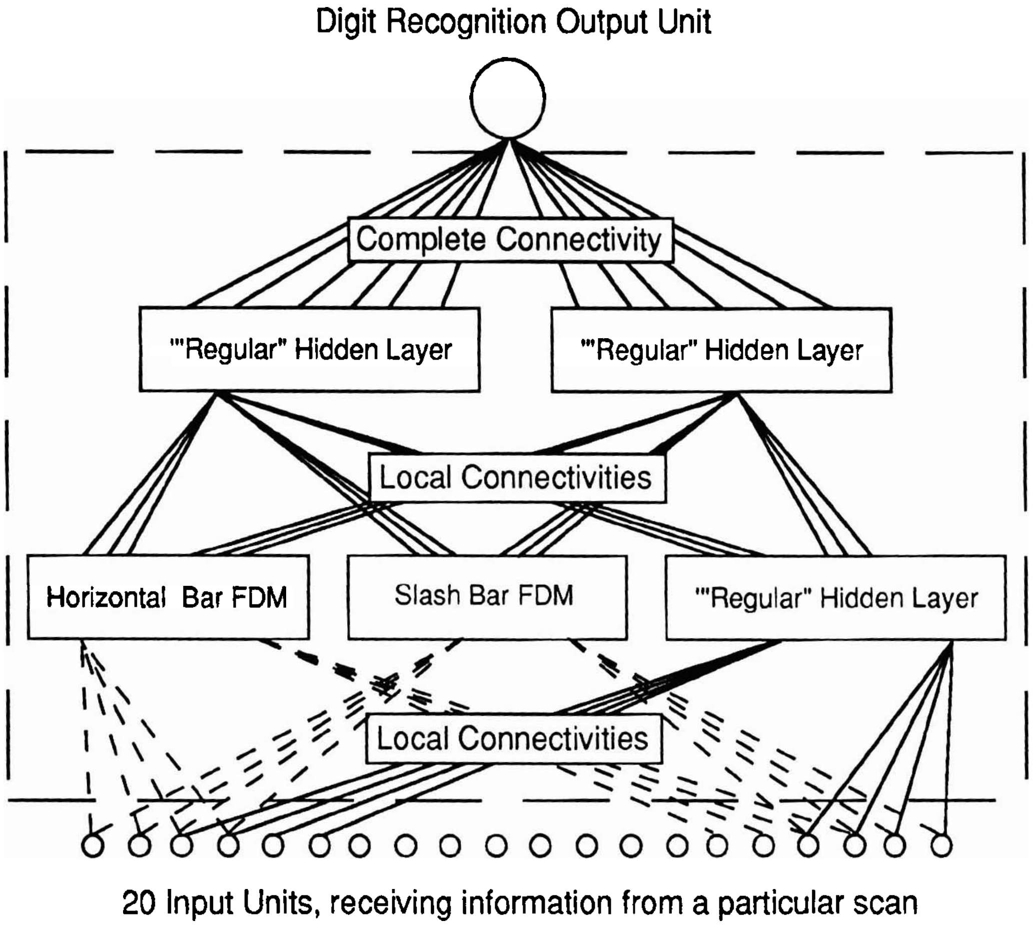

Figure 5: A Single Scan Network (SSN) Module. The input units, 20 in this case, pass information along links which are either frozen, if they are part of a pretrained FDM (dashed lines), or trainable, if they are "regular" links (solid lines). A local hierarchical structure is used to detect higher order features as information propagates towards the output unit. The dashed box demarcates the entire SSN, and is seen replicated in Figure 5.

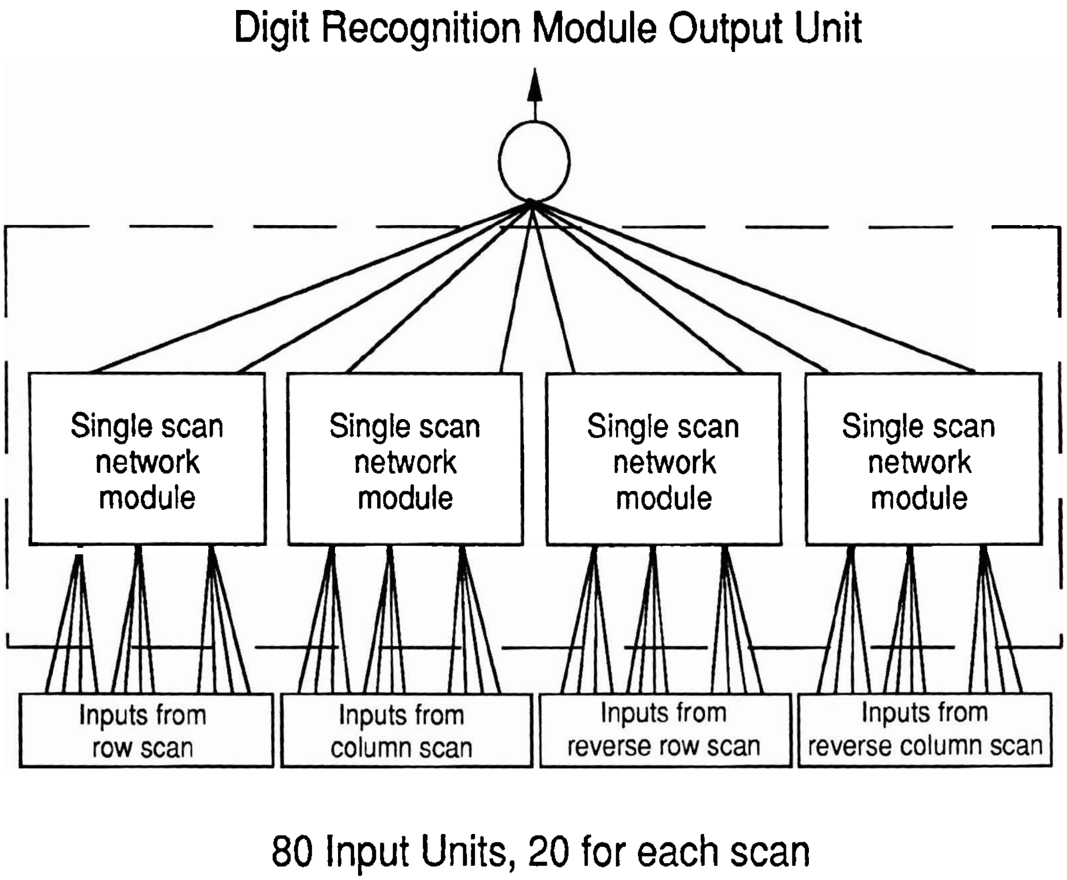
Digit Recognition Module Output Unit
Single scan network module
Single scan network module
Single scan network module
Single scan network module
Inputs from row scan
Inputs from column scan
Inputs from reverse row scan
Inputs from reverse column scan
80 Input Units, 20 for each scan

Figure 6: A Single Digit Recognition Module. 80 input units are used, aligned in 4 banks of 20, receiving information from 4 scans. Information from each scan is processed independently in separate SSNs, and the information is combined at the output level. The dashed box delimits the digit recognition module, utilized repeatedly in Figure 6.

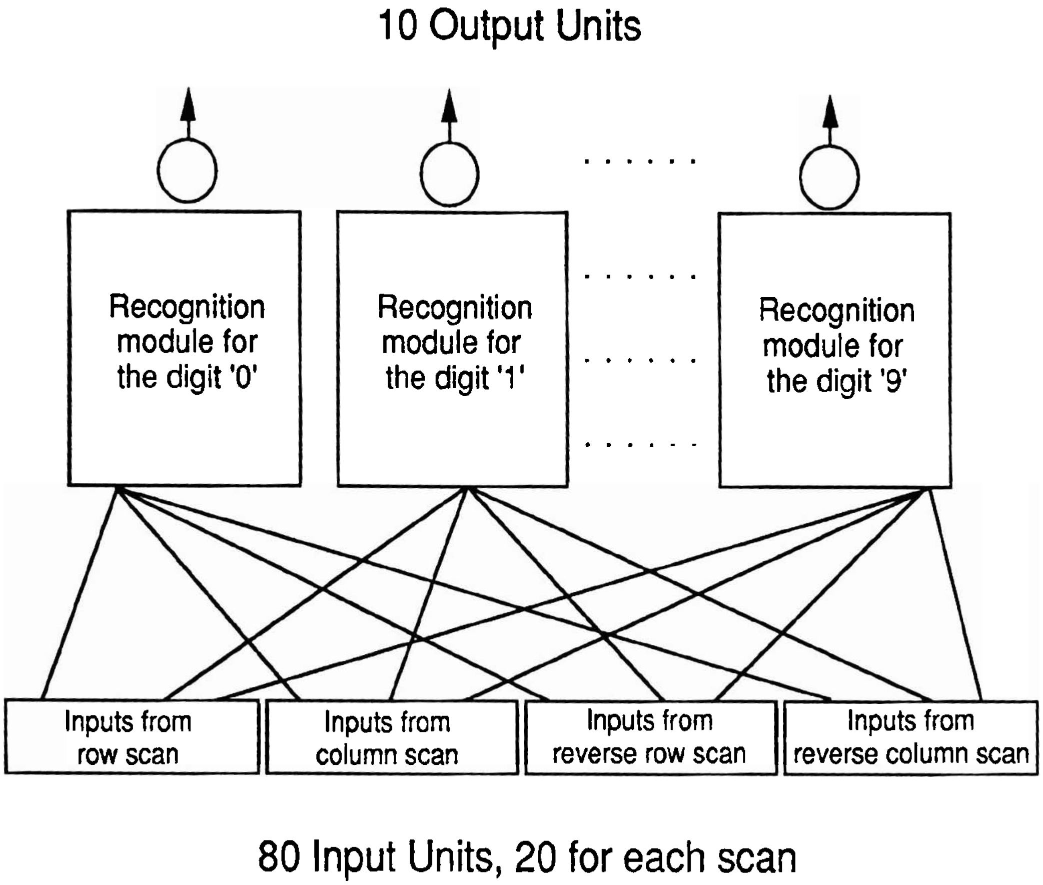
10 Output Units
Recognition module for the digit '0'
Recognition module for the digit '1'
Recognition module for the digit '9'
Inputs from row scan
Inputs from column scan
Inputs from reverse row scan
Inputs from reverse column scan
80 Input Units, 20 for each scan

Figure 7: A Complete Digit Recognition Network. 10 single digit recognition modules are coupled together, each of which separately processes the shared input from 4 scans.

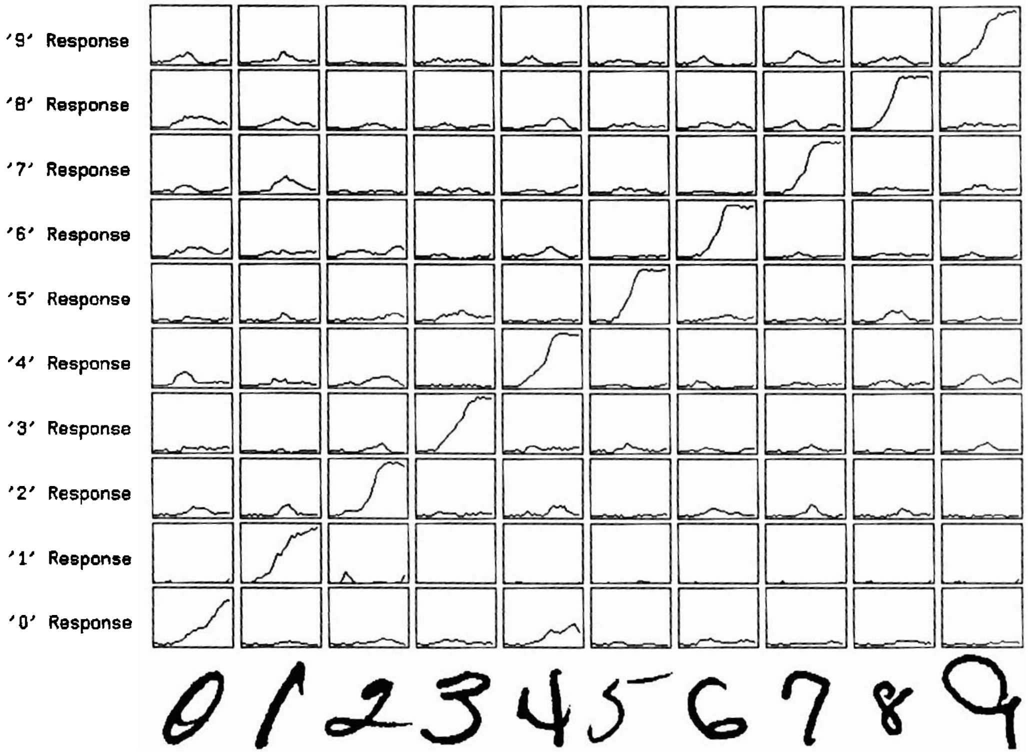

Figure 8: Digit recognition system output unit response, over time, to a typical set of real-world ZIP code digit images. Each digit provokes the highest response in the desired output unit.