LOGPOLAR SAMPLING AND NORMALIZATION BASED ON BOUNDARY CROSSING FOR HANDWRITTEN NUMERALS RECOGNITION

Yuh-Fwu Guu and Behrouz Peikari

Electrical Engineering Department, Southern Methodist University
Dallas, Texas 75275

ABSTRACT

This paper presents a new logpolar sampling procedure for recognition of handwritten numerals. It is shown that this approach requires less computation than the logpolar sampling method employed by Duren and Peikari [1]. Furthermore, in addition to the ability of transforming rotational variation to translational variation, it can also reduce the scale variation. This logpolar sampling is used as a pre-processing stage in conjunction with various neural network structures. The results show that it can be used with a two layered sparsly connected neural network to obtain a better recognition rate than previous works [1, 2]. A normalization method based on boundary crossings is also introduced, it is shown that it requires even less computations than the logpolar sampling method and has the ability of reducing the deformation effect found in handwritten characters. Over 16500 character samples are used in conducting the experiments, recognition rates of 96.24% and 95.91% (96.9697% at 374th training epoch) are obtained using logpolar sampling and normalization method respectively with a 4:1 training/testing partition.

1. INTRODUCTION

Optical character recognition under the category of single character recognition can be divided into two major groups: printed character and handwritten character recognition. The primary concern in dealing with these two categories are different because in the printed character recognition, characters of a certain class are of the exact geometric shape within a particular font, such as the industry standard OCR-A and OCR-B. The possible variations are size, orientation, and noise content. In dealing with handwritten character recognition however, there are no "standard" characters as in the printed case. The characters under consideration could have all the variations associated with size, orientation, noise content, and deformation.

This paper presents two new methods employed in dealing with recognition of handwritten numerals. Concerns are on the pre-processing methods used to reduce the deformation effect. Logpolar sampled characters as inputs to the neural networks are first investigated, followed by the normalization method. The effect of the structure of the neural network classifiers are also investigated. Results obtained using combinations of the pre-processing methods and the network structures are presented

2. LOGPOLAR SAMPLING

Schwartz [3] suggested that the mapping of retinal space onto the striate cortex can be characterized as a logarithmic conformal mapping. This retinal mapping can be described as a space-variant cortical magnification factor which is inversely proportional to the retinal eccentricity and can be approximated by a logarithmic polar coordinate transform. Fischer [4] suggested that if both the retinal space and the striate cortex were treated mathematically as complex planes, the retinal mapping can be approximated by the complex logarithm and the transformation of the visual field into its neural representation can be modeled via the logarithmic mapping function as in Equation 1. The term w and z are both complex numbers and they represent the points in the retinal and cortical space respectively.

$$w = \log z \tag{1}$$

Logpolar sampling takes a function $f(x, y)$ defined at N by N grid points and transforms it to logarithmic polar coordinates around the centroid defined as:

$$(x, y) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{2}$$

where,

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \tag{3}$$

M radial vectors each starting at the centroid outward are defined with angles:

$$\Phi(v) = \frac{2\pi v}{M} \qquad v = 0, 1, \ldots, M-1 \tag{4}$$

0-7803-2431-5/95 $4.00 © 1995 IEEE

M circular sampling path with exponentially increasing radii were taken according to the equation:

$$r(u) = \left(\frac{N}{2}\right)^{\frac{u}{(M-1)}} \qquad u = 0, 1, \ldots, M-1 \quad (5)$$

The character image is represented by a function $f(x,y)$ and is sampled at those $M \times M$ points located at the coordinate $(x(u,v), y(u,v))$, where the M radial axises intersect with the M circular sampling paths. This sampling process can be described as a transformation from (x,y) space to (u,v) space represented as the functions defined in Equation 6 and Equation 7.

$$x(u,v) = x_0 + r(u)\cos\Phi(v) \qquad (6)$$

$$y(u,v) = y_0 + r(u)\sin\Phi(v) \qquad (7)$$

In comparison with the research done by Duren and Peikari [1] which requires a large amount of computations by taking sampling paths that were varied in proportional to the zeroth order moment of the character. The logpolar sampling technique has the advantage of reducing the number of computations required.

3. CHARACTER NORMALIZATION

One approach to the handwritten numeral recongition problem is to normalize the deformed character in the 2-D space. The strokes of the characters appeared more linearly after the normalization process. Yamada, et al [5] succesfully applied line density normalization to the recognition of Chinese characters. They defined line densities as the number of strokes in the horizontal and vertical scans of the character. The main difference between Chinese characters and Arabic numerals is that Chinese characters contain more strokes in general. In order to obtain a better normalization, the line densities are re-defined using boundary crossing as shown in Equation 8.

$$\begin{aligned} C_x(x) = \sum_{y=1}^{N} \{ [1 - f(x,y)]\, f(x,y-1) \\ + f(x,y)\,[1 - f(x,y-1)] \} \\ x = 1, 2, \cdots, N \\ C_y(y) = \sum_{x=1}^{N} \{ [1 - f(x,y)]\, f(x-1,y) \\ + f(x,y)\,[1 - f(x-1,y)] \} \\ y = 1, 2, \cdots, M \end{aligned} \qquad (8)$$

The line densities at a particular pixel location defined in the two-dimensional plane are denoted as D_x and D_y for the x and y coordinates respectively as shown in Equation 9. It can be thought of as the average number of one-to-zero and zero-to-one transitions in the $x-y$ plane. The terms S_x and S_y represent the total number of one-to-zero and zero-to-one transitions with respect to vertical and horizontal scans as given in Equation 10.

$$D_x = \frac{S_x}{M}; \qquad D_y = \frac{S_y}{N} \qquad (9)$$

and

$$S_x = \sum_{x=1}^{M} C_x(x); \qquad S_y = \sum_{y=1}^{N} C_y(y) \qquad (10)$$

The new pixel location (x', y') is selected according to the line density functions such that the line densities at this new location (x', y') are no less than the line densities at the original location (x, y). Equation 11 shown below gives the coordinate of this new location of the pixel (x', y').

$$\begin{aligned} (x', y') &= (min\ x', min\ y') \\ &= \left(min \left\{ x' | \sum_{k=1}^{x'} C_x(k) \geq x D_x \right\} \right. \\ &\qquad \left. , min \left\{ y' | \sum_{k=1}^{y'} C_y(k) \geq y D_y \right\} \right) \end{aligned} \qquad (11)$$

The assumption made here is that the line densities of all pixels are assumed to be the same. We can think of this as a way of re-sampling the original character by forcing the sampling points fall onto a new location at which the line densities are equalized in both x and y directions. The result is that the character image becomes linearly distributed in the two dimensional image plane and the deformation effect is reduced considerably.

4. NEURAL NETWORK CONFIGURATION

Two types of neural networks were used in the experiments. One is the fully connected network, and the other is the sparsely connected network. A fully connected network is referred to as the kind of network in which each and every one of the nodes in the first hidden layer is connected to each and every one of the nodes in the input layer. Connections between any other adjacent layers are still fully connected in all cases. Four fully connected networks with hidden

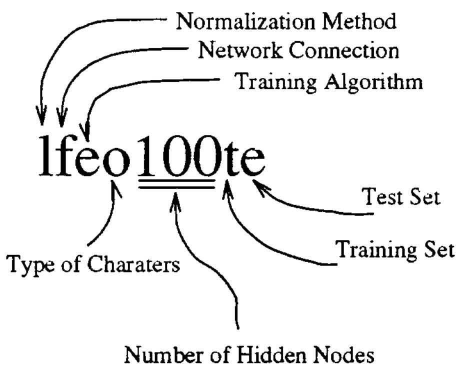

Figure 1: Notation Used for the Tables

nodes of 10, 27, 50, to 100 were used. These four networks were trained and tested using characters processed via the proposed method. The advantage of fully connected networks is that they have more memories compared to the sparsely connected networks. As a result, they require more storage for the weights and consequently, the time required for training these networks is greater than that of the sparsely connected networks. Five sparsely connected networks were tested, two of them are single-layered and the others are two-layered networks.

Figure 1 introduces an example of the notation used in the experiments for the figures and tables.

If there exist a character before the character "f", it represents the normalization method used. A letter "l" represents normalization method proposed by [5], and a letter "d" represents the modified normalization method. The second character represents the type of network connection. A letter "f" represents a fully connected network, and a letter "s" represents a sparsely connected network. The third character represents the training method used for the neural network. A letter "e" represents the error back-propagation method. The fourth character represents the type of pre-processing done on the characters. A letter "o" represents the original characters, and a letter "l" represents the log-polar sampled characters. The number after the pre-processing represent the number of total hidden nodes The second character from the end represents the training character set used and the last character represents the test character set used. In both characters, a letter "e" represents the smaller character set which consists of 330 characters in each of the 10 classes and a letter "t" represents the larger character set which consists of 1325 characters in each of the 10 classes.

Table 1 introduces the notations used in describing the network connection of sparsely connected networks. The left most number denotes the number of input nodes which is 1024 from the 32 by 32 input image for all the networks, and the right most number denotes the number of output nodes which is 10 from the 10 output classes for all networks. The numbers in between denote the number of nodes in the hidden layer(s) starting from the left as the first hidden layer to the right as the second hidden layer. A ":" separates each adjacent layer, and a pair of "(" and ")" in the layer denotes a sparse connection to the layer before it.

Table 1: Notations for Sparsely Connected Neural Network Structures

Notation	Network Structure
1024:(32xM):10	A three-layer network with $(32 \times M)$ nodes in the hidden layer. The 1024 nodes in the input layer are grouped into 32 groups with 32 nodes in each group and each group is connected to a different group of M nodes in the hidden layer. All $(32 \times M)$ nodes in the hidden layer are fully connected to the 10 output nodes.
1024:(32xM):N:10	A four-layer network with $(32 \times M)$ nodes in the first hidden layer and N nodes in the second hidden layer. The 1024 input nodes are grouped into 32 groups with 32 nodes in each group and each group is connected to a different group of M nodes in the first hidden layer. The $(32 \times M)$ first hidden layer nodes are fully connected to the N nodes in the second hidden layer. All N nodes in the second hidden layer are fully connected to the 10 output nodes.

The major problem of fully connected networks is the number of weights required. Although the number of hidden nodes is proportional to the learning capability in general, it is not desirable to have a large amount of weights because the memory required grows as the number of weights increases. One way of avoiding the use of large number of weights is to use sparse connections in the network. It reduces the number of weights required in the network and results in improved network performance as well as better resistance to problems caused by neuron saturation.

5. EXPERIMENTAL INVESTIGATION

Back-propagation training algorithm was used to train the networks, and the training was limited to 150 epoches unless otherwise specified. A total of 16550 characters were divided into two sets. The training set contains 10 classes of handwritten numerals from 0 to 9 with 1325 isolated characters in each class, and the test character set contains also 10 classes with 330 characters in each class. This is so partitioned in order to compare with the results reported in [1, 2]. This database of handwritten Arabic numerals was kindly supplied by Recognition Equipment, Inc.

Methods using Yamada's line density definition and the proposed boundary crossing based normalization were both implemented for comparison, and the results show that the boundary based method performs better than the line crossing method. Results obtained using original characters as inputs to the neural networks are also given in the tables.

5.1. Fully Connected Networks

Table 2 gives the best results obtained for the fully connected networks using the methods proposed earlier. It can be seen from this table that the logpolar sampling, and the normalization methods with 100 hidden nodes both yield a better recognition rate compared with the method proposed in [1, 2].

Table 2: Best Recognition Rates (%) for All Methods with 13250 Training & 3300 Test Samples

Method	Number of Hidden Nodes			
	10	27	50	100
feo	81.2121	88.5152	91.3636	94.1818
fel	88.6667	92.9091	94.5152	95.0606
lfeo	86.0606	91.5455	93.6970	94.8485
dfeo	89.7576	93.4545	94.0000	95.1818

5.2. Sparsely Connected Networks

It was pointed out that sparsely connected network structure requires less weights in the network. It reduces not only the memory requirement but also the training time as well. Additional gain is also obtained on the recognition rates. Table 3 compares the recognition rates obtained for the sparsely connected networks including the single hidden layered and the multiple hidden layered networks with the use of 13250 training and 3300 test characters.

Table 3: Best Recognition Rates (%) for Sparsely Connected Networks with 13250 Training & 3300 test Characters

Type of Networks	Recognition Rates (%)		
	Orig.	LP	Norm.
1024:(32x10):10	88.1212	94.1212	95.4545
1024:(32x20):10	88.9697	94.6667	95.5152
1024:(32x5):32:10	91.7879	96.1515	95.6364
1024:(32x10):10:10	86.8182	93.6667	93.2424
1024:(32x10):32:10	92.7576	96.2424	95.9091

6. CONCLUSIONS

This paper presented methods for the neural network based handwritten numeral recognition. These methods have the ability of reducing the deformation effect. The experiments show that these methods yield better recognition rate than in [1, 2], and the memory requirement are also reduced by using the multi-layer, sparsely connected network. The computation requirement is also favorable.

1. REFERENCES

[1] R. W. Duren and B. Peikari, "A new neural network architecture for rotationally invariant object recognition," *Proc. of the 34th Midwest Symposium on Circuits and Systems*, May 1991, Monterey, Ca.

[2] A. Khotanzad and J. H. Lu, "Classification of invariant image representations using a neural network," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, volumn 38, no. 6, pp. 1028–1038, June, 1990.

[3] E. L. Schwartz, "Computational anatomy and functional architecture of striate cortex: a spatial mapping approach to perceptual coding," *Vision Res.*, pp. 645–669, 1980.

[4] B. Fischer, "Overlap of receptive field centers and representation of the visual field in the cat's optic tract," *Vision Res.*, pp. 2113–2120, 1973.

[5] H. Yamada, K. Yamamoto, and T. Saito, "A nonlinear normalization method for handprinted kanji character recognition-line density equalization," *Pattern Recognition*, no. 9, pp. 1023–1029, 1990.