A Segmentation Algorithm used in Conjunction with Artificial Neural Networks for the Recognition of Real-World Postal Addresses

Michael Blumenstein & Brijesh Verma, *Member of IEEE & IASTED*
Faculty of Engineering and Applied Science
Griffith University, Gold Coast Campus
PMB 50, Gold Coast Mail Centre, Qld 4217 Australia
E-mail: {M.Blumenstein, B.Verma}@eas.gu.edu.au

ABSTRACT- Artificial Neural Networks (ANNs) have been successfully applied for pattern recognition, speech recognition, control and other real world problems. This paper presents a method for segmentation of printed and difficult handwritten postal addresses. The segmentation algorithm is used to prepare raw training data for use with an Artificial Neural Network. The C programming language, the SP2 supercomputer and a SUN workstation were used for the experiments. The algorithm has been successfully tested on real world handwritten postal addresses. Some experimental results are presented in this paper.

Keywords: Artificial Neural Networks, Pattern Recognition, Segmentation, Classification, Postal Address Recognition.

1. INTRODUCTION

ANNs have been successfully used in such areas as pattern recognition (Khotanzad & Lu 1991), medical applications (Zheng et al. 1994), fingerprint analysis (Kulkarni 1994) and signature verification (Han & Sethi 1996) to name just a few. Research has also been conducted regarding the recognition of Postal Addresses using ANNs. However, before an address may be recognised, a large amount of preprocessing must first occur. This paper shall discuss one of the most important techniques employed prior to recognition : Segmentation.

Segmentation has been recognised by (Gilloux 1993) as being a crucial task for the recognition of addresses. It is also a very difficult task, especially when dealing with handwritten addresses. (Srihari 1997) mentions that good segmentation techniques are essential for zip code segmentation in postal address recognition. Srihari's research conveys the idea that there is still room for improvement with regards to segmentation techniques. Finally (Eastwood et al. 1997), also recognise the importance of segmentation in postal address recognition and even in other more difficult applications. The authors also show that satisfactory results may be achieved when segmenting difficult handwriting. In following sections, a conventional algorithm for segmentation shall be presented and discussed.

Sample handwritten characters and addresses were collected and stored in a large database. A preliminary postal address recognition system was developed that could preprocess, segment, and classify the handwritten addresses acquired. An ANN was used for the actual classification, along with other conventional methods.

The organisation of the remainder of the paper is as follows : section 2 discusses the proposed techniques, section 3 details the experimental results, section 4 discusses the results and finally a conclusion is drawn in section 5.

2. PROPOSED TECHNIQUES

There are many steps that need to be taken before handwritten words or characters on an envelope or a page can be recognised by a computer. The 5 stages of our experiments were as follows: 1. Scanning, 2. Binarisation, 3. Segmentation, 4. Preprocessing and 5. ANN recognition techniques. Figure 2 depicts the complete system.

2.1 Scanning

The handwritten characters and addresses were acquired from various students and faculty members around the university. Their handwriting was sampled on A4 sized paper. These handwritten characters were scanned using a flatbed Macintosh scanner and a Macintosh personal computer. The scanned images were saved in Tagged Image Format (TIF). Later, the images were moved across platforms to an IBM PC and converted into a Windows Bitmap format using Paint Shop Pro Version 3.11.

2.2 Binarisation

The Windows Bitmaps were then uploaded onto the SP2 Super Computer in Brisbane. Using a program originally implemented for the recognition of Hindi characters (Verma 1995), each Windows Bitmap was converted into a binary bitmap representation of the handwriting. In this form, segmentation and preprocessing could take place more easily.

2.3 Segmentation

A segmentation program was implemented using the algorithm presented in Figure 1. It was used to separate the individual lines and characters of the handwritten postal addresses. The algorithm payed particular attention to separating hand-printed characters which were “touching” due to poor scanner resolution. The program first looked for characters which were not touching (separated by columns of zeros). If the search could not find a clear breakpoint, it tried to find a point where the density of pixels was sparse. An algorithm was devised especially for this purpose, which searched for a decrease of black pixel density. When a point like this was found the program monitored either side of the point to make sure that it was in between an area of high pixel density. The end result was a file of characters of all different sizes. A simple preprocessing technique was then required to normalise the character matrices. The segmentation algorithm can be best explained in terms of 10 steps, as shown in Figure 1 below:

Step 1.	Average character size for the current address is calculated, by scanning for segregated characters and noting their width and height
Step 2.	If a column exists, check its pixel density. Else go to Step 10
Step 3.	If the pixel density is zero, then segmentation point found. Bypass ensuing columns with pixel density zero, until the beginning of the next character is encountered. Go to Step 2
Step 4.	If previous or next column's pixel density is greater than current column, go to Step 5. Else return to Step 2
Step 5.	Calculate how many columns have been passed since last segmentation point
Step 6.	If the number of columns is smaller than the average size of the character, go to Step 2
Step 7.	If any ensuing columns have a pixel density of zero, go to Step 2
Step 8.	Check if average pixel density of previous and ensuing columns is greater than that of the proposed point
Step 9.	If Step 8 is true, then segmentation point found. Repeat by going to Step 2
Step 10.	End of segmentation procedure

Figure 1. Segmentation Algorithm

2.4 Preprocessing

A simple normalisation technique was employed to create a file of character matrices of the same height and width. First, the largest character in the set was found. The rest of the characters were then padded horizontally and vertically with "0's" to produce character matrices of equal size.

2.5 ANN Recognition Techniques

Feedforward, multi-layered neural networks (Verma 1995), were used for recognition. An Error Backpropagation (EBP) algorithm was used to train the neural networks. The segmented and preprocessed characters were presented to the neural networks as multiple sets of 36 character training pairs (A-Z, 0-9). The ANNs were trained with many different initial values for weights, momentum and learning rate. The architectures of the neural networks were as follows :

For printed characters, the number of input neurons was 345, and the number of outputs was 36. The number of hidden units varied from 10 to 16. For handwritten characters, the number of input neurons was 456, the number of outputs was 36, and the number of hidden units varied from 12 to 100.

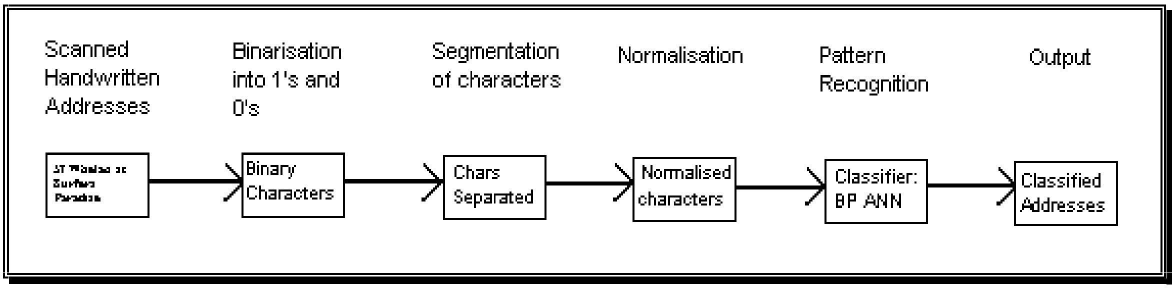

Figure 2. A system for postal address recognition.

3. EXPERIMENTAL RESULTS

3.1 Character Database

The largest database of characters consisted of 4248 characters written by 11 different students and faculty members. A test database was also created consisting of 396 characters. After segmentation and preprocessing, the dimensions of the character matrices were 15x23 and 19x24 for the databases used. Some samples of the characters used are shown in Figure 3.

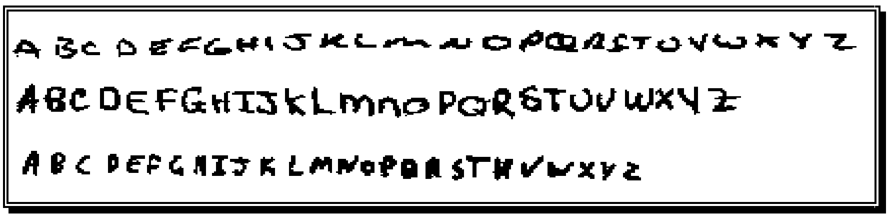

Figure 3. Samples of training characters

Figure 4. Test samples for segmentation algorithm

3.2 Implementation and Experimentation of Segmentation Program

The experiments were executed on the SP2 Super Computer. The first part of experimentation involved the segmentation and preprocessing of the postal addresses. The segmentation program implemented was written in C. It accepted the raw file of bits representing the postal address and output a raw file of various sized character matrices. These were then normalised using a preprocessing system. The general algorithm has already been discussed above in Section 2.3. Finally, a file was output consisting of character matrices appropriate for use with a classification system. An example presenting the segmentation of a handwritten address is presented in Figures 5(a) and 5(b).

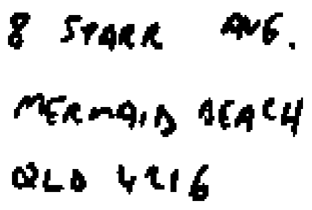

Figure 5(a) Address before Segmentation

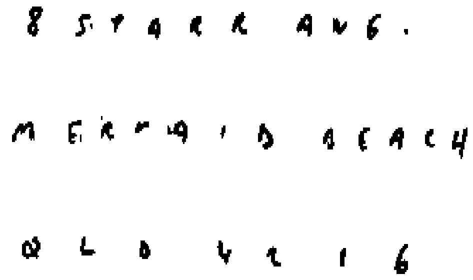

Figure 5(b) Address After Segmentation

3.3 Experimentation involving character recognition

Character recognition experiments were first conducted to ascertain which ANN variable settings would generate the best results for further experiments. The best settings and results are first displayed for printed characters (Table 1) and then subsequently for handwritten characters (Table 2). For the results presented in Table 1, 288 characters were used for training the ANN, and 72 characters were used for testing. 8 different fonts were used for training of the ANN: Times New Roman, Arial, Book Antiqua, Bookman Old Style, Calisto MT, Century Gothic, Century School Book, Comic Sans MS. Subsequently, 2 different fonts were used for testing: Courier, News Gothic MT.

For the results in Table 2, 4248 handwritten characters were used for training and 396 were used for testing. The final column in each table presents classification rates for both training and test sets. Classification rates for the training set reflect results attained by presenting characters the ANN was trained with. Conversely, results presented for the test set refer to classification rates of characters the ANN had never seen before.

Table 1. Results for Printed Characters

Hidden Units	No. of Iterations	Learning Rate	Momentum	Classification Rate [%]	
				Training Set	Test Set
12	1000	0.1	0.1	100	80.56
16	250	0.1	0.1	98.96	81.94
16	300	0.1	0.1	99.65	84.72
16	325	0.1	0.1	99.65	83.33

Table 2. Results for Handwritten Characters

Hidden Units	No. of Iterations	Learning Rate	Momentum	Classification Rate %	
				Training Set	Test Set
60	500	0.1	0.1	95.41	50.00
80	500	0.1	0.1	97.69	56.31
100	500	0.1	0.1	98.73	58.59

3.4 Experimentation with Real-World Postal Addresses

We scanned some printed and handwritten postal addresses for testing the proposed system. Some samples are shown in Figure 4. We used the best neural network structure for classification. Table 3 presents results for the recognition of 10 printed postal addresses using 10 different fonts. Table 4 presents seven of the best results for handwritten postal address recognition out of the 40 addresses tested.

Table 3. Results for printed postal addresses

	Recognition Rate of Address (%)
Address 1 (Arial 9pt)	97.56
Address 2 (Book Antiqua 9 pt)	88.89
Address 3 (Bookman Old Style 9pt)	94.87
Address 4 (Calisto MT)	97.62
Address 5 (Century Gothic 9pt)	85.11
Address 6 (Century Schoolbook 9pt)	97.22
Address 7 (Comic Sans 9pt)	92.11
Address 8 (Courier 9pt)	83.33
Address 9 (News Gothic 9pt)	90
Address 10 (Times New Roman 9pt)	94

Table 4. Results for handwritten postal addresses

	Recognition Rate of Address (%)
Address 1	50
Address 2	60
Address 3	57.43
Address 4	51.28
Address 5	56.76
Address 6	68.75
Address 7	52.78

4. DISCUSSION OF RESULTS

4.1 Segmentation

The conventional segmentation algorithm proved to be quite successful for both printed and handwritten addresses. Satisfactory segmentation was even achieved with difficult and touching handwritten characters. Unfortunately, the algorithm had a number of limitations:

1) Difficulties in segmenting characters with large variations in size

2) Problems with segmenting extremely untidy handwriting

3) Problems with segmentation of non-horizontal addresses

The first limitation affects the algorithm if the size and especially the width of characters vary largely. As can be seen in the algorithm, the average character width plays a large part in checking for an appropriate segmentation point. If a particular character is too large or small, than an erroneous segmentation point may be created as it does not conform to the measurements of other characters.

The second limitation only impedes the segmentation process when dealing with incredibly illegible or untidy handwriting. For example, if two characters are extremely tightly coupled the algorithm may not be able to detect an effective segmentation point.

The final limitation is prominent if a particular address is written in such a way that the lines of the address end up straying heavily away from the horizontal. In most cases, if the lines are slightly straying up or down across the envelope the segmentation process will not be impeded. Other problems may occur if words making up the address are written on a slant, and characters overlap. In future research, rotation and slant correction shall be employed to decrease the severity of these problems.

4.2 Classification

As can be seen clearly from the above tables, the results vary quite substantially. The best and most promising results for printed character recognition could be found when using 12-16 units in the hidden layer, very low values for η and α (0.1 and 0.1), and a number of iterations between 250 and 1000. When using a lower number of iterations for the 12 hidden unit configuration, our ANN performed poorly. For handwritten character recognition, the difficulty of the task could be reflected in the quality of results attained for the test set. Nonetheless, the best settings were found to be between 60 and 100 units in the hidden layer, very low values for η and α (0.1 and 0.1), and 500 iterations.

Although on average the classification percentages were quite high for the training sets, lower recognition rates were being continually received for the test sets. This could be attributed to the difficult nature of the handwriting samples used for experimentation.

Tables 3 and 4 show results for the classification of printed and handwritten postal addresses. For printed addresses, the recognition rates are extremely high. Perfect classification rates are obtained when using fonts the ANN has been trained with. High results are also attained for the recognition of addresses the ANN had not been previously trained with. Finally, handwritten address recognition rates were presented. Although being substantially lower, the results were very impressive as the postal addresses were made up of difficult handwriting which the ANN had not been presented with previously.

5. CONCLUSION

A preliminary intelligent system based on a multi-layered feed forward neural network, using the EBP algorithm, has been developed to recognise handwritten postal addresses. The overall system included a successful segmentation algorithm which could deal with simple and difficult handwriting. Over 4000 real world handwritten characters were used to train the ANN, varying the number of hidden units, the learning rate and the momentum. For printed characters the results showed that by keeping the number of units in the hidden layer low, and using a low learning rate and momentum, good recognition rates could be achieved. It was also found that when using a large database of handwritten characters, better results could be achieved by increasing the number of hidden units. The system was tested on a number of real world printed and handwritten postal addresses, the results obtained were very promising. This work is still in progress, and methods are being devised to hopefully increase recognition rates. Further research shall use a more complex segmentation algorithm, feature extraction techniques, slant correction and rotation. This shall ensure that problems with touching and very difficult handwriting are significantly reduced.

REFERENCES

Al-Emami, S., Usher, M., 1990, 'On-Line Recognition of Handwritten Arabic Characters' in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12, No 7. July. pp 704-709.

Brault, J-J., Plamondon, R., 1993, 'Segmenting Handwritten Signatures at their Perceptually Important Points' in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 15, No. 9. September. pp 953-963.

Denker et al., 1989, 'Neural Network Recognizer for hand-written zip code digits' in Neural Information Processing Systems, Vol. 1, Denver, Morgan-Kaufmann, pp 396-493.

Eastwood, B., Jennings, A., Harvey, A., 1997, 'A Feature based Neural Network Segmenter for Handwritten Words' in Proceedings of ICCIMA '97, Gold Coast, pp 286-290.

Gilloux, M., 1993, 'Research into the new generation of character and mailing address recognition systems at the French post office research center' in Pattern Recognition Letters, Vol. 14, No. 4. April. pp 267-276.

Han, K., Sethi, I. K., 1996, 'Handwritten Signature retrieval and Identification' in Pattern Recognition Letters, Elsevier Science B. V., pp 83-90.

Khotanzad, A., Lu, J., 1991, 'Shape and Texture Recognition by a Neural Network' in Artificial Neural Networks in Pattern Recognition, Sethi, I. K., Jain, A.K., Elsevier Science Publishers B. V., Amsterdam, Netherlands.

Kulkarni, A. D., 1994, Artificial Neural Networks for image understanding, Van Nostrand Reinhold, New York.

Le Cun, Y., 1990, 'Handwritten Digit Recognition with a Back-Propagation Network' in Neural Processing Systems, Vol. 2, Denver, Morgan-Kaufmann, pp 323-331.

Srihari S., N., 1997, 'Handwriting Recognition Technology and its integration into mail processing', in Proceedings of ICCIMA '97, Gold Coast, pp 17.

Verma B., 1995, 'Handwritten Hindi Character Recognition Using RBF and MLP Neural Networks' in IEEE ICNN'95, Perth, pp. 86-92.

Zheng, B., Qian, W., Clarke, L., 1994, 'Multistage Neural Network for Pattern Recognition in Mammogram Screening' in IEEE ICNN, Orlando, Vol. 6, pp 3437-3448.