AN ARTIFICIAL NEURAL NETWORK BASED SEGMENTATION ALGORITHM FOR OFF-LINE HANDWRITING RECOGNITION

M. BLUMENSTEIN AND B. VERMA
School of Information Technology
Faculty of Information and Communication Technology
Griffith University, Gold Coast Campus
PMB 50, Gold Coast Mail Centre, Qld 4217 Australia
E-mail: {M.Blumenstein, B.Verma}@eas.gu.edu.au

Artificial Neural Networks (ANNs) have been successfully applied to Optical Character Recognition (OCR) yielding excellent results. This paper presents a method for segmentation of difficult handwriting with the use of conventional algorithms in conjunction with ANNs. The segmentation algorithm is heuristic in nature detecting important features which may represent a prospective segmentation point. An Artificial Neural Network is subsequently used to verify the authenticity of the segmentation points found by the algorithm. The C programming language, the SP2 supercomputer and a SUN workstation were used for the experiments. The algorithm has been tested on real-world handwriting obtained from the CEDAR database. Some preliminary experimental results are presented in this paper.

1 Introduction

Artificial Neural Networks have proven to be successful in many areas of Pattern Recognition[1]. Many researchers have also applied the excellent generalisation capabilities offered by ANNs to the recognition of characters[2,3]. However, there have only been a handful of researchers using ANNs for the segmentation of printed and cursive handwriting[4,5]. As is mentioned in the literature[6,7], segmentation plays an important role in the overall process of handwriting recognition. Unfortunately, not only is it a vital process but it is also one that has not achieved very accurate results.

Many researchers have used conventional and heuristic algorithms for the segmentation of handwritten words[8,9]. Researchers such as Eastwood et al.[4], and Martin et al.[5], have obtained promising results with the use of ANN-based methods. This research attempts to integrate both conventional and intelligent methods for the segmentation of difficult printed and handwritten words.

A simple heuristic segmentation algorithm is presented which finds segmentation points in printed and joined handwritten words. The algorithm checks for minima in cursive handwriting along with other important features to find segmentation points between characters. An Artificial Neural Network trained with valid segmentation points from a database of scanned, handwritten words is used to assess the correctness of the segmentation points found by the algorithm.

The remainder of the paper is broken down into 4 sections. Section 2 briefly describes the proposed techniques and algorithm, Section 3 provides experimental results, a discussion of the results follows in Section 4, and a conclusion is drawn in Section 5.

2 Proposed Techniques

The following section addresses the steps that were required to preprocess and segment handwritten words using the aforementioned algorithm and an ANN. The techniques used were: 1. Binarisation and segmentation using a heuristic algorithm, 2. Manual segmentation, 3. Preprocessing and training of the ANN, and 4. Testing the heuristic algorithm. An overview of the segmentation process for Steps 3 and 4 is provided in Figure 2.

2.1 Binarisation and segmentation using a heuristic algorithm

Before any segmentation or processing could take place, it was necessary to convert the images into binary representations of the handwriting. A method[10] used in a previous paper was employed for this purpose. A simple heuristic segmentation algorithm was implemented which scanned handwritten words for important features to identify valid segmentation points between characters. The algorithm first scanned the word looking for minimas or arcs between letters, common in handwritten cursive script. In many cases these arcs are the ideal segmentation points, however in the case of letters such as “a”, “u” and “o”, an erroneous segmentation point could be identified. Therefore the algorithm incorporated a “hole seeking” component which attempted to prevent invalid segmentation points from being found.

If an arc was found, the algorithm checked to see whether it had not segmented a letter in half, by checking for a “hole”. Holes, are found in letters which are totally or partially closed such as an “a”, “c” and so on. If such a letter was found then segmentation at that point did not occur. Finally, the algorithm performed a final check to see if one segmentation point was not too close to another. The heuristic algorithm is briefly explained in Figure 1.

Step 1. Average character size for the current word is calculated, by scanning for segregated characters and noting their width and height

Step 2. If a column of pixels exists, check its properties. Else go to Step 9

Step 3. If the pixel density is zero, then segmentation point found. Go to Step 2

Step 4. Check either side of point to verify whether it is located in a valley or minima. If so, go to Step 5. Else return to Step 2

Step 5. Calculate how many columns have been passed since the last segmentation point

Step 6. If the number of columns is greater than the average size of the character, go to Step 7. Else go back to Step 2

Step 7. Check to see whether the point is part of a partially enclosed or totally enclosed character

Step 8. If Step 7 is false, then segmentation point found. Repeat by going to Step 2

Step 9. End of segmentation procedure

Figure 1. Segmentation Algorithm

2.2 Manual segmentation

To train the ANN with both accurate and erroneous segmentation points, the output from the heuristic segmentation algorithm was used. It was necessary to manually separate the coordinates provided by the algorithm into “good” and “bad” segmentation categories and save them to a file. These coordinates were then processed by a further step to produce a training file for the ANN.

2.3 Preprocessing and training of the ANN

A preprocessing program was used to locate the coordinates produced in the previous step and to extract the segmentation points from each binarised word. These segmentation points were then used to create a training file which was presented to a multi-layer feedforward Neural Network trained with the backpropagation algorithm. The structure of the ANN contained between 295 and 335 inputs, 20 to 30 hidden units and 1 output.

2.5 Testing the heuristic algorithm

After training of the ANN, handwriting used for testing was applied to the heuristic segmentation algorithm. Again, many segmentation points were found (some of which were incorrect). All segmentation points were then presented to the ANN to verify the accuracy of the heuristic algorithm's output. The ANN presented output indicating which segmentation points were correct and which were incorrect. Finally, only correct segmentation points were left and extra, non-legitimate points were removed.

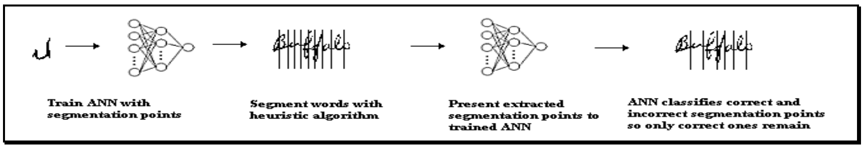
Train ANN with segmentation points
Segment words with heuristic algorithm
Present extracted segmentation points to trained ANN
ANN classifies correct and incorrect segmentation points so only correct ones remain

Figure 2. Segmentation Process

3 Experimental Results

3.1 Handwriting Database

For preliminary experimentation of the method detailed in Section 2, we used samples of handwriting located on the CD-ROM released by CEDAR[11]. The handwritten words used, were city names actually sampled from envelopes collected from the U.S. Postal Service (Figure 3).

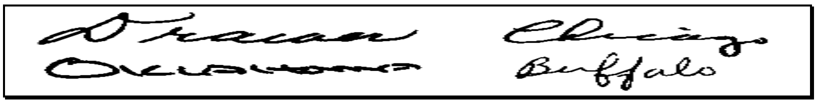

Figure 3. Handwriting Samples Used for Training/Testing

3.2 Implementation and Experimentation

Implementation of the segmenter and preliminary experiments were performed on a SUN workstation. Results for experiments conducted, testing handwritten words and zipcodes are presented in

Table 1. The average recognition rates displayed in Table 1 were calculated by adding all the recognition rates for each word/zipcode and dividing the total by the number of words/zipcodes.

Table 1. Results experiments involving words and zipcodes

	Total No. of Words/ Zipcodes	Total No. of Seg. Points	Avg. Rec. Rate for Training Set [%]	Avg. Rec. Rate for Test Set [%]
Words	6	74	100	100
	13	143	100	65.11
Zipcodes	14	103	100	85.12

4 Discussion of Results

As mentioned earlier, many researchers have used various techniques for the segmentation of characters in handwritten words. Segmentation accuracy rates of above 90% were achieved by Lee et al.[12], however the authors were only dealing with printed alphanumeric characters. Srihari et al.[6] obtained segmentation accuracies of 83% for handwritten zip codes (no alphanumerics). Finally, experiments conducted by Eastwood et al.[5], segmenting cursive handwriting produced a 75.9% accuracy rate. The authors used an ANN-based technique for segmentation with 100,000 training patterns. On average our segmentation accuracy for both experiments involving words (Table 1) was just over 76%. Although our experiments were only preliminary, our results compare favorably with those of other researchers. In further work a much larger database of segmentation points shall be used for training the ANN, which should increase segmentation accuracy rates.

5 Conclusions

A preliminary heuristic segmentation system used in conjunction with a multi-layered feedforward neural network has been presented. Preliminary experiments were conducted on real-world handwritten words. The ANN-based segmentation system produced promising results even with some very difficult handwriting. This research is still in progress, and many improvements to the heuristic algorithm shall be explored to improve the identification of segmentation points. Different ANN settings shall also be pursued to increase final classification accuracy of segmentation points.

6 References

1. A. D. Kulkarni, 1994, *Artificial Neural Networks for image understanding*, (New York: Van Nostrand Reinhold)
2. C. Y. Suen, R. Legault, C. Nadal, M. Cherier, and L. Lam, 1993, Building a New Generation of Handwriting Recognition Systems, *Pattern Recognition Letters*, **14**, 305-315.
3. S-W. Lee, 1996, Off-Line Recognition of Totally Unconstrained Handwritten Numerals Using Multilayer Cluster Neural Network, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, 648-652.
4. G. L. Martin, M. Rashid, J. A. Pittman, 1993, Integrated Segmentation and Recognition through Exhaustive Scans or Learned Saccadic Jumps, *International Journal of Pattern Recognition and Artificial Intelligence*, **7**, 187-203.
5. B. Eastwood, A. Jennings, A. Harvey, 1997, A Feature Based Neural Network Segmenter for Handwritten Words, *ICCIMA'97, Australia*, 286-290.
6. S. N. Srihari, 1993, Recognition of Handwritten and Machine-printed Text for Postal Address Interpretation, *Pattern Recognition Letters*, **14**, 291-302.
7. M. Gilloux, 1993, Research into the New Generation of Character and Mailing Address Recognition Systems at the French Post Office Research Center, *Pattern Recognition Letters*, **14**, 267-276.
8. R. M. Bozinovic, S. N. Srihari, 1989, Off-Line Cursive Script Word Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **11**, 68-83.
9. H. Fujisawa, Y. Nakano, K. Kurino, 1992, Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis, *Proceedings of the IEEE*, **80**, 1079-1092.
10. M. Blumenstein, B. Verma, 1997, A Segmentation Algorithm used in Conjunction with Artificial Neural Networks for the Recognition of Real-World Postal Addresses, *ICCIMA'97, Australia*, 155-160.
11. CEDAR CD-ROM 1, Center of Excellence for Document Analysis and Recognition, State University of New York, Buffalo, http://www.cedar.buffalo.edu/Databases/CDROM1/.
12. S-W. Lee, D-J. Lee, H-S. Park, 1996, A New Methodology for Gray-Scale Character Segmentation and Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, 1045-1051.