Neural-based Solutions for the Segmentation and Recognition of Difficult Handwritten Words from a Benchmark Database

M. Blumenstein[1] and B. Verma[1,2]

[1]School of Information Technology
Griffith University-Gold Coast Campus
PMB 50, Gold Coast Mail Centre
QLD 9726, Australia
Telephone: +61 7 5594 8738
Fax: +61 7 5594 8066
E-mail: {m.blumenstein, b.verma}@gu.edu.au

[2]Department of Computer Engineering
and Computer Science
University of Missouri-Columbia
Columbia, MO 65211, USA
Telephone: +1 573 8846464
Fax: +1 573 8828318
E-mail: bverma@ece.missouri.edu

Abstract

A new intelligent segmentation technique is proposed that may be used in conjunction with a neural classifier and a simple lexicon for the recognition of difficult handwritten words. A heuristic segmentation algorithm is initially used to over-segment each word. An Artificial Neural Network (ANN) trained with 32,034 segmentation points is then used to verify the validity of the segmentation points found. Following segmentation, character matrices from each word are extracted, normalised and then passed through a global feature extractor after which a second ANN trained with segmented characters is used for classification. These recognised characters are grouped into words and presented to a variable-length lexicon that utilises a string processing algorithm to compare and retrieve words with highest confidences. This research provides promising results for segmentation, character and word recognition.

1. Introduction

Researchers have utilised many different approaches for both the segmentation and recognition tasks of handwritten word recognition. Only a few have utilised ANNs for the segmentation of cursive words [1,2]. Even fewer have detailed their findings for the segmentation process of their system. Most researchers tend to measure the success of their system by their findings from the character or word recognition phases. As is mentioned in [3], segmentation plays an important role in the overall process of handwriting recognition. There is still a need to compare results for the segmentation of handwriting using benchmark databases. Cursive word segmentation deserves particular attention since it has been acknowledged as the most difficult of all handwriting segmentation problems [4].

In the proposed word recognition system, heuristic and intelligent methods are used for the segmentation of real-world, handwritten words. Following segmentation, character matrices are extracted from the words and classified. Finally, to show how the segmentation technique may possibly be used in the context of an overall system, a lexicon is used to match each set of recognised characters (each set represents a single word) to potential correct words. The entire system is shown in Figure 1.

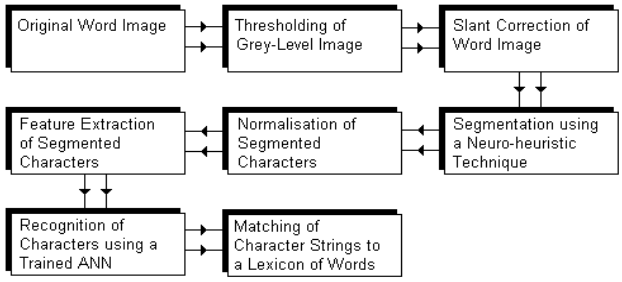

Figure 1. Complete handwriting recognition system

2. Proposed segmentation technique

The segmentation process is briefly explained in the following sections and is illustrated in Figures 2 and 3. A more detailed description can be found in [2].

2.1 Overview of the heuristic algorithm

Prior to segmentation, we employed some simple preprocessing techniques. We first converted each grey level word image into a binary format using Otsu's thresholding algorithm [5]. We then employed a slant detection and correction technique [6].

For both training and testing phases, a heuristic, feature detection algorithm is used to locate prospective segmentation points in handwritten words. Each word is

inspected in an attempt to locate characteristics representative of segmentation points. Six major operations are executed to perform segmentation. 1. Average character width of the word is estimated. 2. Upper and lower word contours are examined to enable the location of possible ligatures. 3. Histograms of vertical pixel density are calculated. Minima in the histograms are used to further confirm the location of possible segmentation points in each word. 4. Words are also scanned for "holes". These regions are marked as being inappropriate to accommodate possible segmentation points. 5. Clusters of proximate segmentation points are analysed and are reduced in number so that only small collections of more likely points representing a particular area may exist. 6. Segmentation points are forced in areas of a word that have a sparse distribution of segmentation points. The result is a set of over-segmented words that await ANN verification.

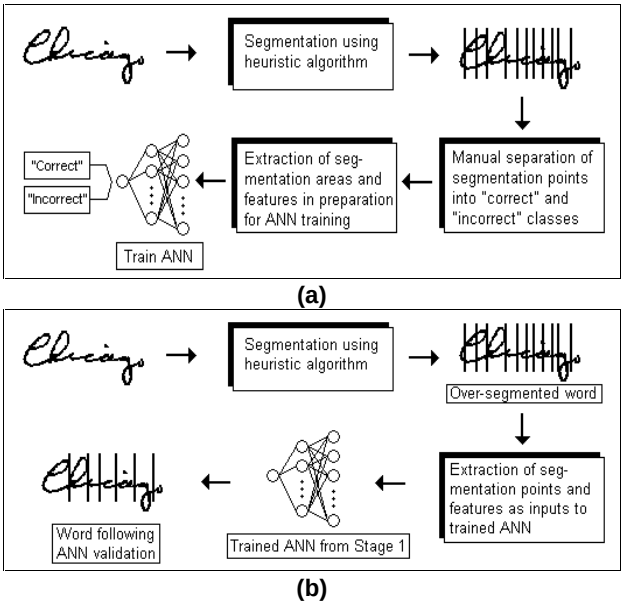

Figure 2. Proposed segmentation technique (a) Stage 1: training phase (b) Stage 2: testing phase

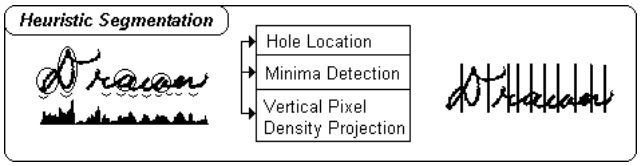

Figure 3. Some steps in heuristic segmentation

2.2 Training phase of the segmentation technique

Prior to ANN training, the heuristic feature detector is used to segment all words that shall be required for the training process. The segmentation points output by the heuristic feature detector are manually analysed so that the x-coordinates can be categorised into "correct" and "incorrect" segmentation point classes. For each segmentation point, a matrix of pixels representing the segmentation area is extracted and stored in an ANN training file. Each matrix is first normalised in size, and then significantly reduced in size by a simple feature extractor. The feature extractor breaks the segmentation point matrix down into small windows of equal size and analyses the density of black and white pixels. Therefore, instead of presenting the raw pixel values of the segmentation points to the ANN, only the densities of each window are presented. As an example, if a window exists that is 4x4 in dimension, and contains 6 black pixels, then a single value of 0.38 (Number of pixels/16) is written to the training file to represent the value of the window. Accompanying each matrix the desired output is also stored in the training file (0.1 for an incorrect segmentation point and 0.9 for a correct point) ready for ANN training.

2.3 Testing phase of the segmentation technique

Following ANN training, the words used for testing are also segmented using the heuristic, feature-based algorithm. This time there is no manual processing. The segmentation points are automatically extracted and are fed into the trained ANN. The ANN then verifies which segmentation points are correct and which are incorrect. Finally, upon ANN verification, each word used for testing should only contain valid segmentation points.

3. The recognition of segmented characters

Another area of the handwriting recognition domain that has not received sufficient attention is the comparison of researchers' results for segmented character recognition utilising benchmark handwritten word databases. Following the technique described in Section 2, character segments were extracted from each word and then recognised by a classifier.

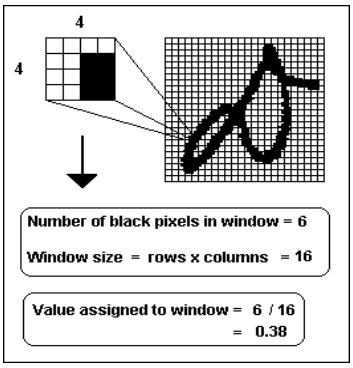

Figure 4. A window of 4x4 in dimension is extracted from a character matrix

Using the segmentation points generated for training in Section 2.2, segmented characters were extracted to train a backpropagation neural network. The extracted characters were first normalised and then reduced in size by the global feature extraction technique detailed in Section 2.2 and Figure 4. Characters used for testing were extracted using the same procedure. Following neural network training, segmented test characters were passed through the ANN and were classified.

4. Recognition of words using a simple lexicon

A variable sized lexicon of words was implemented to recognise all words used for testing. The lexicon was solely implemented to indicate how the segmentation technique could be used as part of a fully operational handwriting recognition system. It must be noted therefore that our research was mainly focussed on producing an accurate segmentation component, not to produce a highly accurate word recogniser.

Each recognised character set from the previous section (representing a single word), was used to test the lexicon. The lexicon used a simple string comparison algorithm, which first matched each character of each lexicon word to the characters in the "test" word being examined. The number of correct characters was noted. In further processing, information such as the order of the characters found in each test word and the length of each test word, were compared to those of all lexicon words. Each word in the lexicon was given a confidence rating for every test word depending on the number of matching characters found and the number of characters that appeared in the correct sequence: See Figure 5 below.

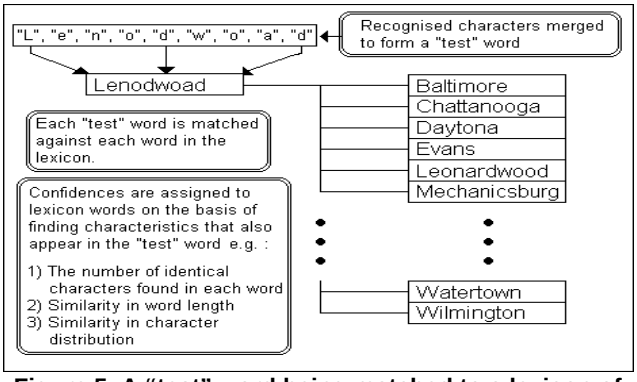

Figure 5. A "test" word being matched to a lexicon of words

5. Experimental results

For experimentation of the techniques detailed in Sections 2 to 4, handwritten words from the CEDAR benchmark database [7] were used. In particular we used all the words contained in the "BD/cities" directory of the CD-ROM.

5.1 Segmentation results

All segmentation experiments were conducted using an ANN trained with the backpropagation algorithm. Table 1 shows the top experimental results for the verification of segmentation points by the ANN. Many experiments were performed varying settings such as the number of iterations, the number of hidden units, momentum and learning rate. For each experiment the number of inputs remained the same: a 14x3 matrix of pixel densities (42 inputs). The number of outputs was always set to 1. Table 1 shows the top results obtained when the ANN was trained with 32,034 training patterns (correct and incorrect segmentation points). The number of testing patterns was 3162.

Table 1. Results for validation of segmentation points using 32034 training patterns

Classification rate for test set	Classification rate [%] test set
2568/3162	81.21

5.2 Segmented character recognition results

The character recognition experiments were also conducted using a backpropagation neural network. The number of characters used for training and testing respectively, were 15297 and 1212. The number of outputs was 52 representing 26 uppercase characters (A-Z) and 26 lowercase characters (a-z). The results obtained for character recognition are presented in Table 2 and are divided into two categories. Results are presented for experiments which distinguished and which did not distinguish between uppercase and lowercase characters.

Table 2. Character recognition results using 15297 training patterns (a) Case Sensitive Experiments (CSE) and (b) Non-Case Sensitive Experiments (N-CSE)

	Classification rate for test set	Classification rate [%] test set
(a) CSE	680/1212	56.11
(b) N-CSE	709/1212	58.50

5.3 Word recognition results

Following character recognition, sets of characters comprising words were presented to lexicons of size 10, 50 and 100 words. Word test sets of size 40, 148 and 211 were presented to the lexicons. Both words contained in the lexicon and words used for testing were randomly selected for the experiments. Top word recognition results for each lexicon size are presented in Table 3. The value "N" ranges between 2 and 10, and indicates whether the correct word was located in the top 2, 5, or 10 choices suggested by the lexicon.

Table 3. Word recognition results

Lexicon size	Recognition rates for top N choices [%]		
	N=2	N=5	N=10
10	100	100	N/A
50	66.67	71.43	85.71
100	50	65	70

6. Discussion of results

6.1 Classification of segmentation points

The neuro-heuristic algorithm achieved a recognition rate of 81.21% for identification of 3162 segmentation point patterns. Other results in the literature for segmentation of handwritten words include: Eastwood *et al.* [1]: 75.9%, Han and Sethi [8]: 85.7% and Yanikoglu and Sandon [9]: 97%. Although Yanikoglu and Sandon's results are very high, it must be noted that they did not use a benchmark database of real-world unconstrained words for their experiments. The results for segmentation achieved in this research compare favourably with other researchers.

6.2 Classification of segmented characters

Results obtained by researchers for segmented character recognition are still not as high as those for the recognition of handwritten numerals. Top researchers [10,11] have obtained classification rates ranging from 67-80% on samples from the CEDAR CD-ROM. The experimental procedures matching closest to those described in this research, are that of Yamada and Nakano [11]. Their results for case sensitive, segmented character recognition was 67.8%. The top result presented in Table 2 is just above 56%. Our results are slightly lower, however it is important to note that in their research, Yamada and Nakano used more training samples, and long recognition times were recorded due to the algorithms used. Following ANN training, the classifier in our research recognised over 1000 characters in 2-3 seconds. Therefore, taking into account factors such as speed and simplicity our classification method has also generated favourable results.

6.3 Overall word recognition

The results obtained for overall word recognition were not significantly high. The recognition rates were only high for the smallest lexicon of words, however as the lexicon increased in size the recognition rate dropped suddenly. For lexicons of size 50 and 100, the recognition rates for the top 2 to 10 choices were reasonable, however top choices were quite low. Recognition at the word level was not given significant attention because our research mainly focussed on character segmentation. For future experiments, we shall use a more effective lexicon-based approach for the word recognition stage to improve overall word recognition results.

7. Conclusions

An intelligent segmentation technique has been presented in this paper, producing good results. It was used to segment difficult cursive and printed handwritten words from the CEDAR database. A segmented character recogniser has also been presented as part of an overall handwriting recognition system. Considering the speed and simplicity of the system, results presented for character recognition and word recognition are favourable. The main focus of the research presented in this paper was the segmentation of handwritten words. It has been noted that there are very few researchers that have published their segmentation results for handwritten word recognition in the context of a complete system. Therefore it is hoped that further research can be dedicated to improving and to comparing results for this very important procedure.

References

[1] B. Eastwood, A. Jennings and A. Harvey, "A Feature Based Neural Network Segmenter for Handwritten Words", *ICCIMA '97, Gold Coast, Australia*, 1997, pp. 286-290.

[2] M. Blumenstein and B. Verma, "A New Segmentation Algorithm for Handwritten Word Recognition", *IJCNN '99, Washington, U.S.A.*, 1999.

[3] R. G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 18, 1996, pp. 690-706.

[4] Y. Lu and M. Shridhar, "Character Segmentation in Handwritten Words – An Overview", *Pattern Recognition*, Vol. 29, 1996, pp. 77-96.

[5] N. Otsu, "A threshold selection method from gray level histograms", *IEEE Trans. Systems, Man and Cybernetics*, Vol SMC-9, 1979, pp. 62-66.

[6] R. M. Bozinovic and S. N. Srihari, "Off-Line Cursive Script Word Recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, 1989, pp. 68-83.

[7] J. J. Hull, "A Database for Handwritten Text Recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 16, 1994, pp. 550-554.

[8] K. Han and I. K. Sethi, "Off-line Cursive Handwriting Segmentation", *ICDAR '95, Montreal, Canada*, 1995, pp. 894-897.

[9] B. Yanikoglu and P. A. Sandon, "Segmentation of Off-line Cursive Handwriting using Linear Programming", *Pattern Recognition*, Vol. 31, 1998, pp. 1825-1833.

[10] F. Kimura, N. Kayahara, Y. Miyake and M. Shridhar, "Machine and Human Recognition of Segmented Characters from Handwritten Words", *ICDAR '97, Ulm, Germany*, 1997, pp. 866-869.

[11] H. Yamada and Y. Nakano, "Cursive Handwritten Word Recognition Using Multiple Segmentation Determined by Contour Analysis", *IEICE Trans. On Information and Systems*, Vol. E79-D, 1996, pp. 464-470.