ENHANCING NEURAL CONFIDENCE-BASED SEGMENTATION FOR CURSIVE HANDWRITING RECOGNITION

Chun Ki Cheng, Xin Yu Liu, Michael Blumenstein and Vallipuram Muthukkumarasamy

School of Information Technology, Griffith University-Gold Coast Campus
PMB 50, Gold Coast Mail Centre, QLD 9726, Australia
E-mail: M.Blumenstein@griffith.edu.au

ABSTRACT

This paper proposes some directions for enhancing a neural network-based technique for automatically segmenting cursive handwriting. The technique fuses confidence values obtained from left and center character recognition outputs in addition to a Segmentation Point Validation output. Specifically, this paper describes the use of a recently proposed feature extraction technique (Modified Direction Feature) for representing segmentation points and characters to enhance the overall segmentation process. Promising results are presented for Segmentation Point Validation and cursive character recognition on a benchmark dataset. In addition, a new methodology for detecting segmentation paths is presented and evaluated for extracting characters from cursive handwriting.

1. INTRODUCTION

The problem of automated handwriting recognition has persisted for many decades. Active research still continues to produce a satisfactory solution for recognizing off-line cursive handwriting. The motivating factors include commercial applications and scientific progress in an age-old artificial intelligence problem. One of the main impediments for progress has been the inherent variability in handwritten material [1].

Handwriting recognition itself is a mechanical process that transforms graphical human handwritten scripts into symbols that are stored on a computer system in the form of ASCII code or Unicode. One of the major problems in recognizing unconstrained cursive words is the process of segmentation [2],[3]. Segmentation is the process of separating the characters in a word, so that they may be used to assist in final word interpretation. Some systems use the method of over-segmentation to dissect the word at many intervals into primitives. The term "primitive" refers to an entire character or character components. Following initial over-segmentation, various techniques may be used to correctly assemble the primitives using contextual processing to recognise entire words. The removal of incorrect segmentation points from over-segmented words is still a difficult problem. A solution to this problem would guarantee a higher success rate for handwritten word recognition. A number of segmentation techniques have been proposed, some of which are described below.

Yanikoglu and Sandon [4] proposed a segmentation algorithm by evaluating a cost function to locate successive segmentation points along the baseline. The decision to segment at a particular point is made if the first minimum cost is located. The cost is calculated by summing the weights of four global characteristics or “style parameters” in the cursive script. The algorithm used a linear programming technique to obtain the weights of the features. The global characteristics include pen thickness, dominant slant, average character width and distance from the previous segmentation point. Finally, characters are extracted by finding the best angular line.

Nicchiotti and Scagliola [5] presented a simple but effective segmentation algorithm. The algorithm is divided into three main steps. The first step is to detect possible segmentation points by analyzing the minima in the lower contour and holes. The second step is to determine the cut direction of the segmentation point. The chosen direction is the direction that contains the least number of black pixels. Finally, over-segmented strokes are merged back to the main character by some heuristic rules.

Xiao and Leedham [6] presented a knowledge-based technique for cursive word segmentation. Based on connected component analysis, those components that contain more than one character are over-segmented based on a face-up or face-down region. Then over-segmented components are merged into a single character based on the knowledge of the character structure.

In this research, enhancements to a neural-based segmentation technique are proposed. The current technique analyses the surroundings of every suspicious segmentation point found by a heuristic segmenter. It incorporates a rule-based fusion component to combine three neural network-based confidence values for verification of correct and incorrect segmentation points [7].

To enhance the current segmentation technique, a recently proposed feature extraction technique [8] is used for performing Segmentation Point Validation (SPV) [9] and recognizing left and centre characters (LC and CC respectively) associated with each segmentation point. Finally, a novel method for generating segmentation paths between cursive characters is described and evaluated for the purpose of enhancing character extraction.

The remainder of the paper is broken down into 4 sections. Section 2 describes previous work along with enhancements to the neural confidence-based segmentation technique; Section 3 provides experimental results, followed by a discussion in Section 4. Finally, conclusions are drawn in Section 5.

2. SEGMENTATION TECHNIQUE

2.1. Previous work

In previous work by the authors [9], a feature-based heuristic segmenter used in conjunction with an SPV technique was proposed for segmenting cursive handwritten script. Further work [7] included LC and CC validation in the segmentation process.

Initially, the heuristic algorithm mentioned above was used to over-segment each handwritten word whereby a neural-based validation technique was employed to verify whether each segmentation point was "valid" or "invalid" by using the fusion of confidence values of the Segmentation Area (SA), LC and CC.

For SPV, a Density Feature (DF) extraction technique was used to divide the SA into small windows of equal size and analyse the number of foreground (black pixels) in each window [9]. For the purpose of obtaining confidence values from LC and CC recognition, the Transition Feature (TF) extraction technique [10] was used for processing the characters.

Recent work has shown that the Modified Direction Feature (MDF) enhances the character recognition process and outperforms TF [8]. This work has demonstrated the superiority of MDF for describing complex patterns based on their contour or boundary. This prompted the current investigation to determine the feasibility of employing MDF for SPV, LC and CC recognition to enhance the overall segmentation process. Figure 1 provides an overview of the entire neural confidence-based segmentation technique. The investigation in this paper is only concerned with SPV, LC and CC recognition in addition to the character extraction process.

In the next sub-sections, the MDF technique is described, along with the process of SPV and character recognition. Finally, a novel process of segmentation path generation is described.

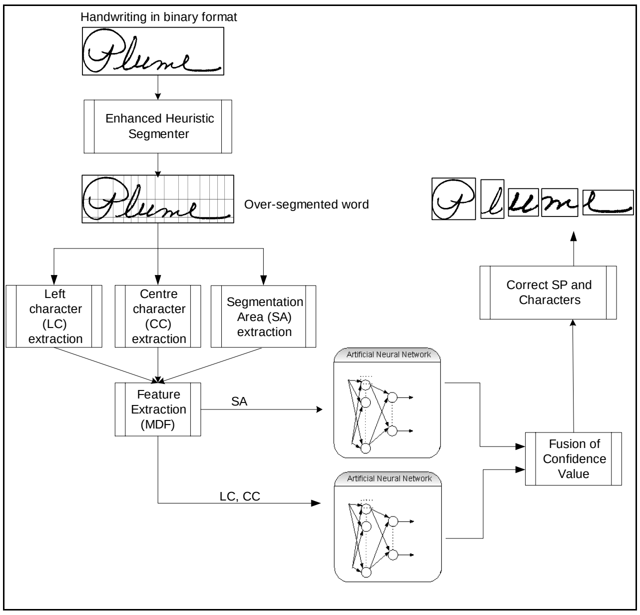

Figure 1: An Overview of the Neural Confidence-based Segmentation Technique

2.2. Modified Direction Feature (MDF)

For MDF [8], feature vector creation is based on the location of transitions from background to foreground pixels in the vertical and horizontal directions of a binary image. When a transition is located, two values are stored: the Location of the Transition (LT) and the Direction Transition (DT). An LT is calculated by taking the ratio between the position where a transition occurs and the distance across the entire image in a particular direction [10]. The DT value at a particular location is also stored. The DT is calculated by examining the stroke direction of an object's boundary at the position where a transition occurs (as defined in Blumenstein et al. [11]). Finally, a vector comprising the [LT, DT] values in each of four possible traversal directions is created.

2.3. Neural confidence calculation

2.3.1 Segmentation Point Validation (SPV)

As mentioned above, following heuristic segmentation it is necessary to discard "incorrect" segmentation points while preserving the "correct" points in a cursive word. This is achieved by calculating a number of confidences for each Prospective Segmentation Point (PSP) generated by the heuristic segmenter. For SPV, a neural network is trained with features extracted from SAs originally located by the heuristic algorithm. The neural network verifies whether each particular area is or is not characteristic of a segmentation point [9]. If an area is positively identified as a segmentation point, the network outputs a high confidence (>0.5). Otherwise the network will output a confidence close to 0.1. In this research, the MDF extraction technique was used to describe the segmentation area.

2.3.2 Left and centre character classification

For this step, additional neural networks trained with handwritten characters (upper case and lower case) are required to confirm the first neural network's output. The network(s) is/are presented with areas immediately centred on/adjacent to each segmentation point. Area width is calculated based upon average character width. If for example, the area immediately to the left of the PSP proves to be a valid character, the network will output a high confidence (LC) for that character class. At the same time, if the area immediately centred on the segmentation point provides a high confidence for the reject neuron (CC), then it is likely that the PSP is a valid segmentation point. The "reject" output of the neural network is specifically trained to recognise non-character patterns (i.e. joined characters, half characters or unintelligible primitives). If this neuron gives a high confidence, this will usually indicate that the particular area being tested is a good candidate for a segmentation point. Otherwise, if any valid characters are given a high confidence (in the centre character area), it is unlikely that that particular area should be segmented. The procedure of SPV, LC and CC validation is illustrated in Figure 2. Fusion of character and segmentation point confidences is detailed in [7].

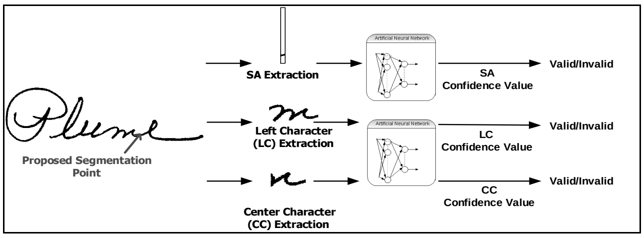

Figure 2: Overview of SA, LC and CC extraction and validation

In this research, two novelties are introduced to enhance the LC and CC classification rate. Firstly, instead of using the Transition Feature (TF) [10] for incorporation of character confidences into the segmentation technique, the neural network was trained on feature vectors produced by MDF. Secondly, in previous work, two separate neural networks were trained for both upper case and lower case characters. This introduced the problem of deciding upon when to use the lower case or upper case networks. Hence, in order to bypass this issue, lower case and upper case characters were combined into a single network containing 37 outputs. The configuration was similar to that undertaken in previous work [11], where upper and lower case characters that were similar in appearance were grouped in the same class i.e. ‘c’ and ‘C’ would share one output class. The only exception was that in this case a reject neuron was also added. The reject neuron was trained to fire when a non-character component was presented to the network (as described above).

2.4. Character extraction by segmentation paths

Previously in the neural confidence-based segmentation technique, LC and CC were extracted using vertical dissections based on the x-coordinates of PSPs provided by the heuristic segmenter (mentioned earlier). It was found that this simplistic scheme was inadequate for the purpose of extracting overlapping and tightly coupled characters in cursive script. The reason being that in some cases, characters would be imprecisely split. This section details a novel character extraction procedure based on the segmentation points output by the heuristic segmentation algorithm.

2.4.1 Segmentation Path Detection (SPD)

The first step of extracting characters using SPD is to measure the ascender and descender of a word image. Ascenders and descenders are strokes that extend above or below the middle zone or main body of a handwriting sample. Next, the main body of the image is equally divided up into 4 sections, namely sections 1, 2, 3, and 4 (See Figure 3). Based on the x-coordinate of a segmentation point, SPD performs backward traversal. Once a foreground (black) pixel is encountered, the system checks whether the location of the black pixel is below section 1. The line at the bottom of section 1, in Figure 3, is called the “best-fit” line.

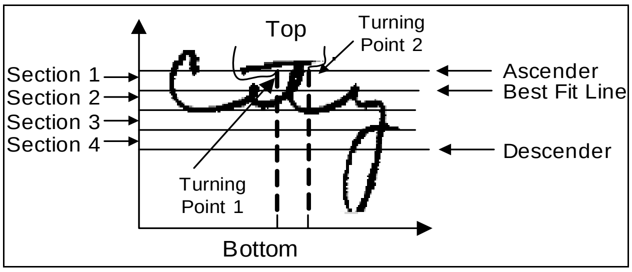

Figure 3: Word sample sections and segmentation path generation

The “best-fit” line is used as a threshold position, which informs the algorithm whether or not an alternate extraction path should be detected. If the encountered black pixel is below the “best-fit” line, then this pixel, along with all connected foreground pixels are ignored. However, if this black pixel exists on or above the “best-fit” line, this is considered to be the starting point of an overlapping stroke. This pixel is called the “turning point”. Commencing from this turning point, a path directed around the overlapping stroke is explored. The algorithm attempts to investigate the right hand side of the turning point. If it is possible to reach the top row of the image, then the extraction path is found. Otherwise, if the traversal to the right hand side is blocked, then it will go back to the turning point, and traverse towards the left hand side. As shown in Figure 3, both left-hand and right-hand segmentation paths of the character ‘t’ are detected. Once an extraction path is located, all pixel coordinates are stored for the purpose of character extraction.

3. EXPERIMENTAL RESULTS

3.1. Handwriting database

For the experiments detailed in this section, patterns for SPV and character recognition were obtained from handwritten words contained on the CEDAR benchmark database [12]. Specifically, segmentation points and characters for training and testing were obtained from the “/train/cities/BD” and “/test/cities/BD” directories respectively.

3.2. Classifier configuration

The classifier chosen for the task of SPV and character classification was a feed-forward Multi-layered Perceptron (MLP) trained with the resilient backpropagation (BP) algorithm. For experimentation purposes, the architectures were modified varying the number of inputs, outputs and hidden units.

The number of inputs to each network was associated with the size of the feature vector for each image. Various vector dimensions were investigated. The most successful vector configurations were of size 80 for SPV and 120 for character classification (using MDF). For SPV and character classification, the number of transitions recorded in each direction was 2 and 3 respectively. The size of the input vector for SPV using the DF technique was 42.

The number of outputs in the SPV experiments varied between one and two (for MDF only).

3.3. SPV Results

Results for SPV are presented below in tabular form. Table 1 presents top results comparing MDF and DF using a total of 32028 segmentation patterns for training and 3162/4854 patterns for testing.

Table 1. SPV rates with a BP-MLP

	Test Set Recognition Rate [%]			
	3162 Patterns		4854 Patterns	
	DF	MDF	DF	MDF
1-Output	81.21	**82.19**	80.61	**81.15**
2-Outputs	N/A	81.97	N/A	**81.15**

3.4. Character classification results

This sub-section lists character classification results using a single neural network trained with both upper and lower case characters in addition to a reject neuron (for non-character patterns). In total, 25830 characters were used for training and 3179 for testing. As the number of reject patterns in the above training set represented a large proportion of the data, it was decided that the number of reject patterns be halved in subsequent experiments to demonstrate the effect on the recognition rate. As a result of this procedure, the training set subsequently contained 20464 characters, with the test set remaining constant. Table 2 lists results using both configurations.

Table 2. Character recognition rates with a BP-MLP

	Test Set Recognition Rate [%]	
	All reject patterns	Half of reject patterns
Total Test Set	**67.54**	64.39
Reject Patterns only	**78.49**	70.10
Characters Only	50.29	**54.83**

3.5. Segmentation path results

Experimental results are displayed below for correct character extraction employing the SPD technique proposed above. Table 3 displays the percentage of words where characters were all successfully extracted whilst including errors introduced by the heuristic segmenter. Table 3 also shows the percentage of words where characters were all correctly extracted without the interference of incorrect segmentation points (ISPs). This is an ideal situation and supposes that all segmentation points are correct.

Table 3. Character extraction rates using SPD

	Character Extraction Rate [%]	
	Including ISPs	Excluding ISPs
317 Words	78.9	95.27

4. DISCUSSION

4.1. SPV discussion

As may be seen from Table 1, in comparing the recognition rates when using DF and MDF, the MLP trained with MDF patterns produces a slightly higher recognition rate. The small increase in recognition rates demonstrates that the MDF is comparable with DF for small, uncomplicated patterns. When a two-output neural network was used (the first neuron indicated a “correct” segmentation and the second indicated an “incorrect” one), the recognition rates on both MDF data sets either remained constant or decreased nominally in comparison with the single-output MLP. A comparison was not directly possible in this case with the DF dataset.

In future experiments, methods for decreasing the MDF input vector size will be explored. It is hypothesized that upon providing less information for MLP training, an increase in the recognition rate will be possible.

4.2. Character classification

The use of a 37-output neural architecture was considered an important step for the overall segmentation process. With the current configuration, although the recognition rate was not excessively high, it is possible to classify both lower and upper case characters with a single network.

As may be seen from Table 2, the results for recognizing reject patterns is nearly 80% when using all available patterns for training. This is a favourable outcome, as the LC and CC depend on this confidence for correct segmentation. Conversely, the character recognition rate is substantially lower, however it may be seen that when half of the reject patterns are removed for training, a higher character recognition rate is achieved. This indicates that the slight disproportion between characters and reject patterns may be leading to a bias during training. In future it may be possible to incorporate a larger set of characters for training and subsequently increase the recognition rate even further.

4.3. SPD discussion

As may be seen in Table 3, the results for correct character extraction are most favourable. The result of 78.9%, using the x-coordinates produced by the heuristic segmenter is encouraging. Upon improving the segmenter in future work, the success of the character extractor may approach the ideal rate of 95.27%.

5. CONCLUSIONS AND FUTURE WORK

This paper described a number of strategies to enhance the process of neural confidence-based segmentation for cursive handwritten words. The enhancements included the use of an MDF extraction technique for SPV, LC and CC classification. Encouraging results were obtained that might contribute to the increase of overall segmentation performance. Lastly, a new SPD technique was proposed for character extraction. This technique proved to be very successful, and has the potential for enhancing the process of LC and CC extraction.

In future work, the above-mentioned enhancements will be integrated into the neural confidence-based segmenter. It is expected that the overall segmentation error will decrease and will in turn facilitate an increase in cursive word recognition accuracy.

6. REFERENCES

[1] R. G. Casey, "Moment Normalization of Handprinted Characters", IBM Journal of Research Development, pp. 548-557, 1970.

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

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

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

[5] G. Nicchiotti and C. Scagliola, "A Simple and Effective Cursive Word Segmentation Method", Proceedings of the 7th International Workshop on the Frontiers of Handwriting Recognition (IWFHR-7), pp. 499-504, 2000.

[6] X. Xiao and G. Leedham, "Knowledge-based Cursive Script Segmentation", Pattern Recognition Letters, Vol. 21, pp. 945-954, 2000.

[7] M. Blumenstein, and B. K. Verma, "Analysis of Segmentation Performance on the CEDAR Benchmark Database", Proceedings of the International Conference on Document Analysis and Recognition (ICDAR '01), pp. 1142-46, 2001.

[8] M. Blumenstein, X. Y. Liu and B. Verma, "A Modified Direction Feature for Cursive Character Recognition", Proceedings of the International Joint Conference on Neural Networks (IJCNN '04), Budapest, Hungary, (CD-ROM Proceedings), 2004.

[9] M. Blumenstein and B. Verma, "A New Segmentation Algorithm for Handwritten Word Recognition", Proceedings of the International Joint Conference on Neural Networks (IJCNN '99), Washington D.C., pp. 878-882, 1999.

[10] P. D. Gader, M. Mohamed and J-H. Chiang, "Handwritten Word Recognition with Character and Inter-Character Neural Networks", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 27, pp. 158-164, 1997.

[11] M. Blumenstein, B. Verma and H. Basli, "A novel feature extraction technique for the recognition of segmented handwritten characters", Proceedings of the Seventh International Conference on Document Analysis and Recognition, (ICDAR '03), Edinburgh, Scotland, pp. 137-141, 2003.

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