

Cursive Character Segmentation Using Neural Network Techniques

Author

Blumenstein, Michael

Published

2008

Book Title

Machine Learning in Document Analysis and Recognition

DOI

https://doi.org/10.1007/978-3-540-76280-5_10

Downloaded from

<http://hdl.handle.net/10072/23529>

Link to published version

https://link.springer.com/chapter/10.1007/978-3-540-76280-5_10

Griffith Research Online

<https://research-repository.griffith.edu.au>

Cursive character segmentation using neural network techniques

Michael Blumenstein¹

School of Information and Communication Technology, Griffith University, Gold Coast campus, PMB 50 Gold Coast Mail Centre, Queensland 9726, Australia
M.Blumenstein@griffith.edu.au

Summary. The segmentation of cursive and mixed scripts persists to be a difficult problem in the area of handwriting recognition. This research details advances for segmenting characters in off-line cursive script. Specifically, a heuristic algorithm and a neural network-based technique, which uses a structural feature vector representation, are proposed and combined for identifying incorrect segmentation points. Following the location of appropriate anchorage points, a character extraction technique, using segmentation paths, is employed to complete the segmentation process. Results are presented for neural-based heuristic segmentation, segmentation point validation, character recognition, segmentation path detection and overall segmentation accuracy.

1 Introduction

The problem of automated handwriting recognition has endured for many decades. Active research still persists in order to pursue 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 refers to the method 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 in the literature, some of which are described below.

In [4], Bozinovic and Srihari attempt to locate possible segmentation points based on proximity to minima in the lower contour and the use of other rules that force segmentations in areas that are between two distant segmentation points. A technique proposed by Cheriet [5] for extracting “key letters” in cursive script analyses face-up and face-down valleys along with open loop regions. Cheriet employs background analysis to achieve segmentation.

Some of the more recent studies employing dissection or presegmentation include that of Han and Sethi [6] who proposed an algorithm for segmenting handwritten words based on a number of features such as crossing points, loops, concave and convex points. They reported that 50 real-world postal address images were segmented with an accuracy of 85.7%. Yamada and Nakano [7] reported a segmentation algorithm that segmented cursive words based on contour features. Reasonable recognition rates were obtained when the segmentation algorithm was used as part of a complete word recognition system.

Yanikoglu and Sandon [8] 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 included pen thickness, dominant slant, average character width and distance from the previous segmentation point. Finally, characters were extracted by finding the best angular line.

Eastwood *et al.* [9] proposed a neural-based technique for segmenting cursive script. In their research they trained a neural network with feature vectors representing possible segmentation points as well as “negative” features that represented the absence of a segmentation point. The feature vectors were manually obtained from training and test words in the CEDAR benchmark database. The accuracy of the network on a test set of possible segmentation points was 75.9%.

Dimauro *et al.* [10] proposed an advanced technique for segmenting cursive words as part of a recognition system to read the amounts on Italian bank cheques. The segmentation technique is based on a hypothesis-then-verification strategy. Initially, the entire word image is searched, and connected components are located within the image. Each “block” detected via this process is passed to a recogniser. If the block is rejected, a hypothesis is generated to split the block by using a “drop falling” algorithm. The algorithm employs a number of rules that analyse the background of the image to determine the first cutting point. They then employ a descending procedure that simulates a “drop-falling” process. The dropping procedure is guided by rules that take into account neighbouring pixels and a regional analysis of

the upper contour to form an appropriate segmentation path. The hypothesis is then verified by classifying the strokes that have originated as a result of segmentation. A nearest neighbour technique is employed for this process. If the stroke is classified with high confidence, the segmentation hypothesis is accepted. Otherwise, a different hypothesis is considered.

Nicchiotti *et al.* [11] 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 analysing 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 one 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 [12] 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 paper, an existing neural-based segmentation technique [13] is enhanced to validate prospective segmentation points. The existing technique first uses a Feature-based Heuristic Segmenter (FHS) [14] to over-segment the handwriting. Following this, a neural confidence-based module is used to evaluate a prospective segmentation point by obtaining a fused value from three neural confidence values: segmentation point validation (SPV), left character validation (LCV) and centre character validation (CCV). The segmentation technique has two advantages. Firstly, it can reduce the number of missed segmentation points and hence increase the overall character/word recognition rate in subsequent processing. Secondly, since the number of segmentation points is optimised directly following over-segmentation, it can reduce the processing time of later stages.

The enhancements to the existing segmentation technique include an Enhanced Heuristic Segmenter (EHS) that employs ligature detection and a neural assistant for obtaining better prospective segmentation points. In addition, the neural confidence-based module is improved by using 1) a recently proposed feature extraction technique [15] for processing relevant features, 2) a single character classifier for the recognition of left characters and centre characters and 3) a segmentation path detection-based character extraction technique [16].

The remainder of the paper is broken down into 4 sections. Section 2 describes the enhanced neural-based segmentation technique. Section 3 provides experimental results, followed by discussion in Section 4. Finally, conclusions are drawn in Section 5.

2 Enhanced Segmentation Technique

This section presents some enhancements to the neural-based segmentation technique. The new heuristic segmenter, EHS, employs two new attributes - ligature detection and a neural assistant. The first component was investigated since the former segmenter, FHS, could not effectively locate prospective segmentation points that were located under over-lapped strokes. The second feature, the neural assistant, uses a hybrid strategy that combines a character classifier and heuristic rules to over-segment the handwriting. Figure 1 shows an overview of the EHS algorithm.

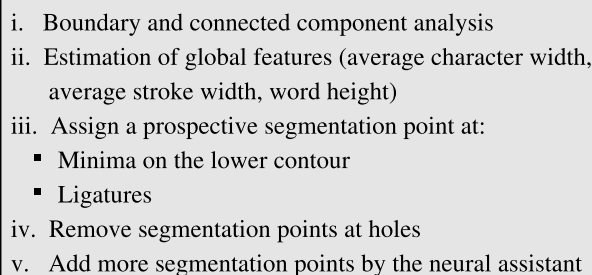
- 
- i. Boundary and connected component analysis
 - ii. Estimation of global features (average character width, average stroke width, word height)
 - iii. Assign a prospective segmentation point at:
 - Minima on the lower contour
 - Ligatures
 - iv. Remove segmentation points at holes
 - v. Add more segmentation points by the neural assistant

Fig. 1. Overview of EHS algorithm

The improved neural confidence-based module uses a newly proposed feature extraction technique, the Modified Direction Feature (MDF) for SPV, LCV and CCV. LCV and CCV use a single classifier for character recognition and a Segmentation Path Detection (SPD) technique is used to extract characters for the recognition process. Figure 2 illustrates an overview of the entire neural-based segmentation technique. In the following sub-sections, further details of ligature detection, the neural assistant, MDF, neural confidence calculation/fusion and SPD are provided.

2.1 Ligature Detection

A ligature is a small stroke that is used to connect joined/cursive characters. One of the major features of a ligature is that it is usually located within the “middle-region” of handwritten words spanning an area down to the word baseline. Hence, a baseline detection technique can be used to identify this middle region. In this work, a modified vertical histogram is generated based on the middle region of the handwriting to locate possible ligatures.

Baseline detection

Small strokes in a word image may extend above or below the main body of handwriting. Such letter components are called ascenders and descenders

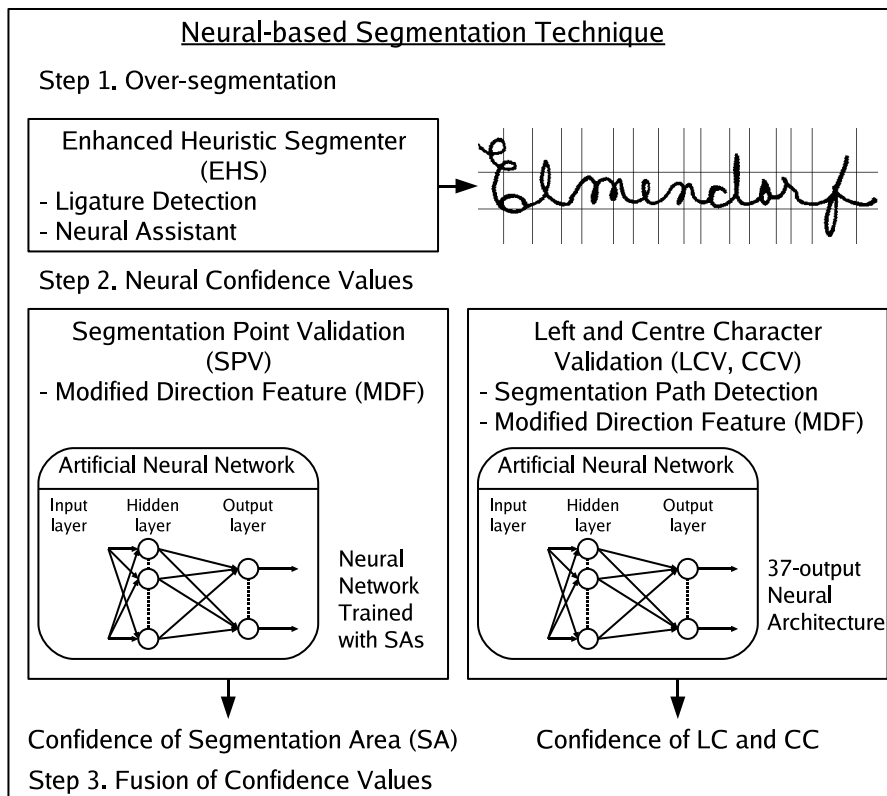


Fig. 2. Overview of the improved, neural-based segmentation technique

respectively. Examples of letters that contain such strokes are: ‘f’, ‘j’, ‘g’, ‘T’ etc. Hence the letters that contain ascenders or descenders may overlap parts of characters in the main body that do not contain such strokes. In order to over-segment the word image more accurately, it is necessary to remove ascenders and descenders before the actual segmentation process. In this research, the technique calculates the average vertical value of the maxima and minima of the upper and lower contours respectively. Outlier maxima and minima values are removed based on this average value. Finally, baselines are estimated by the average of the remaining maxima and minima.

Modified vertical histogram

The second step in the ligature detection algorithm is to analyse the middle region and to locate ligatures. One common approach is the use of vertical (density) histogram analysis. The analysis is based on the vertical distribution of foreground pixels. The histogram is drawn by a projection of the total

number of foreground pixels in each column of the word image. Areas with low pixel density are then identified as possible segmentation points. Figure 3 illustrates an example vertical histogram; the vertical histogram is formed based on the middle region of the word “Top”.

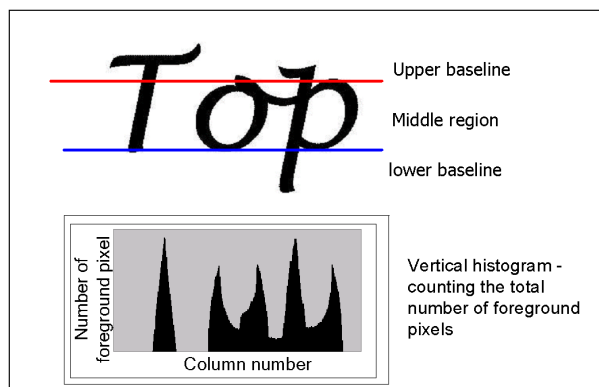


Fig. 3. Vertical histogram analysis

Figure 3 illustrates that there are an excessive number of “low” density regions. This is because the vertical histogram is not adequate to distinguish the difference between “holes” and “ligatures”. In this research, a modified vertical histogram was developed to improve the accuracy of ligature location. Figure 4 shows the modified vertical histogram of the word shown in Figure 3.

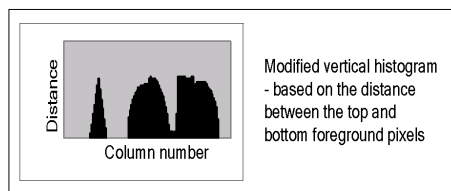


Fig. 4. Modified vertical histogram analysis

The concept of the modified vertical histogram is formed by calculating the distance between the top and bottom foreground pixels for each column in a word image. As may be seen from Figure 4, the ligature region is clear and hence easy for the segmenter to detect. One weakness of the modified vertical histogram is that it is not suitable for characters with overlapped strokes. But in this research, since the overlapped strokes are removed in most cases (i.e. the modified vertical histogram is formed from the middle region), the advantage of the modified vertical histogram can then be maximized.

Ligatures are located using the modified vertical histogram and a heuristic based on the average stroke width. Regions with distance values smaller than the average stroke width are defined as ligatures.

2.2 Neural Assistant

The neural assistant uses a character classifier and some extra heuristics to generate additional segmentation points following regular feature-based segmentation point assignment. Regions between two successive prospective segmentation points are extracted and processed by MDF in order to obtain a confidence value. Additional segmentation points are added based on the confidence value and the distance between the two prospective segmentation points. Experimental results in [16] showed that the classifier could be effectively used to distinguish character and non-character components, and hence could provide appropriate assistance in the current step.

2.3 Modified Direction Feature (MDF)

Recent work has shown that the Modified Direction Feature (MDF) enhances the character recognition process and outperforms some popular feature extraction techniques such as the Transition Feature (TF) [15]. This work demonstrated the superiority of MDF for describing patterns based on their contour or boundary. This prompted an investigation to determine the feasibility of employing MDF for SPV, LC and CC recognition to enhance the overall segmentation process. The details of MDF have been described in [15].

2.4 Neural confidence calculation and fusion

Segmentation Point Validation (SPV)

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 confidence values for each prospective segmentation point (PSP) generated by the heuristic segmenter. For SPV, a neural network is trained with features extracted from segmentation areas (SAs) originally located by the heuristic algorithm. The neural network verifies whether each particular area is or is not characteristic of a segmentation point [14]. 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.

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 5. Fusion of character and segmentation point confidences is detailed in the next sub-section and in [13].

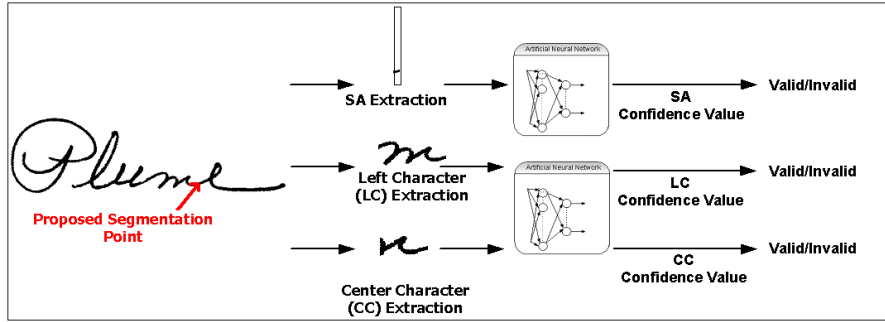


Fig. 5. Overview of SA, LC and CC extraction and validation

Confidence fusion

A Correct Segmentation Point (CSP) is found:

if $f_{SPV_Ver}(ft1) \geq 0.5$ AND
 $f_{LCC_Ver}(ft2)$ is a high character confidence AND
 $f_{CCC_Ver}(ft3)$ is a high non-character confidence;

$$f_{CSP}(ft1, ft2, ft3) = f_{SPV_Ver}(ft1) + f_{LCC_Ver}(ft2) + f_{CCC_Ver}(ft3)$$

where, $f_{SPV_Ver}(features)$ - Confidence value from the Segmentation Point Validation neural network. $f_{LCC_Ver}(features)$ - Left Character Confidence (LCC) value from the character neural network. $f_{CCC_Ver}(features)$ - Centre Character Confidence (CCC) from the character neural network (reject neuron output).

An Incorrect Segmentation Point (ISP) is found:

if $f_{SPV_Ver}(ft1) < 0.5$ AND
 $f_{LCC_Ver}(ft2)$ is a high non-character confidence AND
 $f_{CCC_Ver}(ft3)$ is a high character confidence;

$$f_{ISP}(ft1, ft2, ft3) = (1 - f_{SPV_Ver}(ft1)) + f_{LCC_Ver}(ft2) + f_{CCC_Ver}(ft3)$$

where, $f_{SPV_Ver}(features)$ - Confidence value from Segmentation Point Validation neural network. $f_{LCC_Ver}(features)$ - Left Character Confidence value from character neural network (reject neuron output).
 $f_{CCC_Ver}(features)$ - Centre Character Confidence value from character neural network (highest confidence from 36 character neuron outputs).

Finally, the outcome of fusion is decided by the following equation:
 $f(confidence) = \max(f(CSP), f(ISP))$

Enhancements to classification procedure

Building on previous work, two novelties are introduced to enhance the LC and CC classification rate. Firstly, instead of using the Transition Feature (TF) [17] 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 [18], 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).

2.5 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.

Segmentation Path Detection (SPD)

The first step of extracting characters using SPD is to measure the ascenders and descenders of the word image. As mentioned earlier, 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 6). 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 6, is called the “best-fit” line.

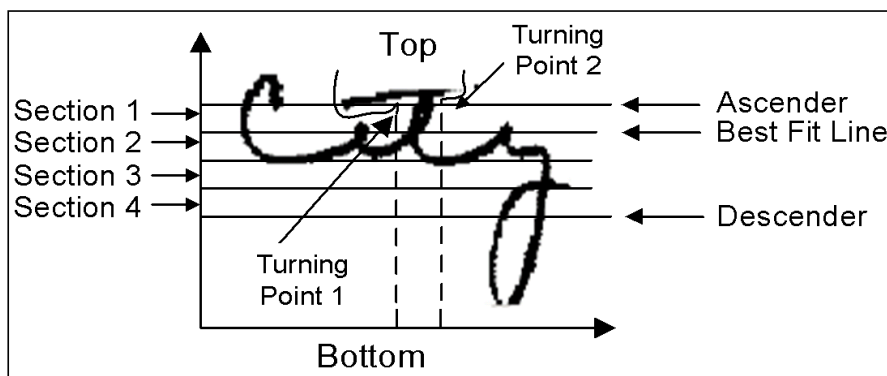


Fig. 6. Word sample sections and segmentation path generation

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 the

algorithm returns to the turning point, and traverses towards the left hand side. As shown in Figure 6, 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

A number of experiments were conducted in this research. Experiments were first conducted to compare the performance of the EHS and FHS algorithms. Further experiments were conducted to compare the performance of MDF and a simple Density Feature (DF) extraction technique for SPV and subsequently the accuracy of character classification for LC and CC. In addition, the performance of character extraction was evaluated using SPD and finally the overall neural confidence-based segmentation technique for validating prospective segmentation points, was tested.

Segmentation performance is measured based on three types of segmentation errors: “over-segmentation”, “missed” and “bad” metrics. “Over-segmentation” refers to a character that has been divided into more than three components. A “missed” error occurs when no segmentation point is found between two successive characters. The “bad” error refers to a segmentation point that could not be used to extract a character perfectly, but might still be used for the purpose of character separation.

3.1 Handwriting Database and Neural Network Configuration

The training and testing patterns for this work were obtained from handwritten words contained in the CEDAR benchmark database [19], specifically the “/train/cities/BD” and “/test/cities/BD” directories respectively.

The classifiers used in this research were feed-forward Multi-layered Perceptrons (MLPs) trained with the resilient backpropagation (BP) algorithm. For experimental purposes, the architectures were modified varying the number of inputs, outputs and hidden units.

3.2 EHS and SPV Segmentation Performance

Table 1 shows the segmentation performance of FHS and EHS. The results are based on the 1031 segmentation points that existed between joined, cursive characters contained in the CEDAR words used for testing.

Results for SPV are presented below in tabular form. Table 2 presents top results comparing MDF and DF using a total of 32028 segmentation patterns for training and 3162/4854 patterns for testing.

Table 1. Segmentation performance of EHS and FHS (1031 segmentation points)

	Segmentation Error Rates		
	Over-segmented [%]	Missed [%]	Bad [%]
FHS	4.07	4.07	6.99
EHS	2.72	2.42	4.56

Table 2. 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	<i>N/A</i>	81.97	<i>N/A</i>	81.15

3.3 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 3 lists results using both configurations.

Table 3. 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	70.1
Characters Only	50.29	54.83

3.4 Segmentation Path Results

Experimental results are displayed below for correct character extraction employing the SPD technique proposed above. Table 4 displays the percentage of words where characters were all successfully extracted whilst including errors introduced by the heuristic segmenter. Table 4 also shows the percentage of words where characters were all correctly extracted without the interference of incorrect segmentation points (ISPs). The latter is an ideal situation and supposes that all segmentation points are correct.

Table 4. Character extraction rates using SPD

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

3.5 Performance of the Neural-based Segmentation Technique

The errors of the enhanced neural-based segmentation technique are calculated based on the number of correct segmentation points obtained in the word samples. The total number of segmentation points in the 317 test word samples is 1718. Only 1031 segmentation points that existed between joined/cursive characters were chosen for testing purposes. The reason for this is to test the segmenter on its ability to separate cursive character components. Table 5 shows the overall results of the enhanced neural-based segmentation technique and the existing neural-based segmentation technique using 317 testing words.

Table 5. Overall results of the neural-based segmentation technique (1031 segmentation points)

	Segmentation Error Rates		
	Over-segmented [%]	Missed [%]	Bad [%]
Existing Technique	7.08	2.33	10.86
Experiment 1 (FHS)	8.73	0.1	8.63
Experiment 2 (EHS)	7.37	0.1	6.79

4 Analysis and Discussion of Results

4.1 Analysis of EHS Over-segmentation

The introduction of ligature detection to locate prospective segmentation points hidden by large horizontal strokes or overlapping characters proved quite successful. As may be seen from Table 1, EHS performed fairly well on the test set with only 2.42% of “missed” errors being generated.

Two problems were found during the inspection process. The first problem arises when segmenting very noisy characters. Since the enhanced heuristic algorithm was heavily dependent on contour analysis, heavy noise that was inherent around the handwriting could cause serious errors. One of the solutions to this problem was additional pre-processing.

The second problem that was observed related to the neural assistant. The main problem was incorrect classification. However, overall the classification

rate was acceptable based on the current character classifier’s recognition accuracy (approx. 89%).

The missed segmentation points were due to the neural assistant misrecognising two joined characters as a single character. This type of error is very hard to deal with, since when two characters are tightly coupled, the ligature cannot be detected. One solution is to employ a better neural classifier or incorporate more heuristic rules. However, in some cases the missed segmentations may be recovered when the neural-based segmentation technique is employed, which uses the centre area associated with each segmentation point. Figure 7 provides some sample handwriting with segmentation points found by EHS.

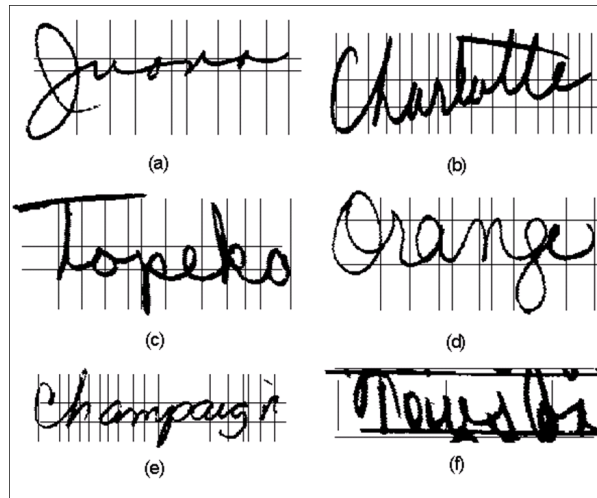


Fig. 7. Sample word images segmented by the enhanced feature-based heuristic segmenter. (a), (b), (c) successful words. (d), (e), (f) unsuccessful words.

Although neural classifiers may contribute problems in some instances, their use in the described segmenter was very beneficial, because it could introduce more segmentation points without using complex heuristics.

4.2 SPV Discussion

As may be seen from Table 2, 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.

4.3 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 3, 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.

4.4 SPD Discussion

As may be seen in Table 4, 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 further, the success of the character extractor may approach the ideal rate of 95.27%.

4.5 Analysis of Neural-based Segmentation

As may be seen from Table 5, the segmentation technique was successful at discarding bad segmentation points as well as recovering “missed” segmentation points by adding them at large gaps between points in words based on the average character width. Both experiments (using FHS and EHS) recorded the same “missed” error of only 0.1%, which is a very promising result. Furthermore, the results also showed that the enhanced heuristic segmenter was able to produce better inputs to increase overall segmentation results.

The reason for the higher “bad” errors by the neural-based segmentation technique as compared to those obtained by the enhanced heuristic segmenter is because some “missed” errors are turned into “bad” ones. This is due to the technique recovering “missed” segmentation points based on the average character width. In some cases, it could not perfectly locate the character boundary (using SPD) and hence contributed to the “bad” error. Although the “over-segmentation” error went up slightly as compared to previous work, it is possible to recover this at a later stage.

Another reason for the increase of the segmentation performance is related to the use of the MDF and the segmentation path-based character extraction

technique (SPD). Since ‘clean’ characters can be extracted and MDF provides better features for the single classifier, the performance of LCV and CCV are improved.

5 Conclusions and Future Work

This paper describes an improved neural-based segmentation technique for cursive words. The technique included an enhanced heuristic segmenter to over-segment handwriting in addition to the use of an MDF extraction technique for SPV, LCV and CCV. The enhanced heuristic segmenter provided better inputs to the subsequent neural validation process. Encouraging results were obtained that can increase the overall performance of a segmentation-based handwriting recognition system.

In the future, EHS will be used to locate prospective segmentation points from the training set facilitating re-training and testing of the SPV classifier for further enhanced performance. The above-mentioned technique will also be tested on a larger dataset to validate the improvements proposed. Finally, a new character extraction technique that uses the direction feature on the character’s boundary will be investigated.

References

1. Casey R G (1970) Moment Normalization of Handprinted Characters. *IBM Journal of Research Development* 14:548–557.
2. Casey R G, Lecolinet E (1996) A Survey of Methods and Strategies in Character Segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18:690–706.
3. Lu Y, Shridhar M (1996) Character Segmentation in Handwritten Words - An Overview. *Pattern Recognition* 29:77–96.
4. Bozinovic R M, Srihari S N (1989) Off-Line Cursive Script Word Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 11:68–83.
5. Cheriet M (1993) Reading Cursive Script by Parts. *Proceedings of the Third International Workshop on Frontiers in Handwriting Recognition (IWFHR-3)*, Buffalo, New York, May 25-27, 403–408.
6. Han K, Sethi I K (1995) Off-line Cursive Handwriting Segmentation. *Proceedings of the 3rd International Conference on Documents Analysis and Recognition (ICDAR '95)*, 894–897.
7. Yamada H, Nakano Y (1996) Cursive Handwritten Word Recognition Using Multiple Segmentation Determined by Contour Analysis. *IEICE Transactions on Information and Systems* E79-D:464–470.
8. Yanikoglu B, Sandon P A (1998) Segmentation of Off-Line Cursive Handwriting using Linear Programming. *Pattern Recognition* 31:1825–1833.
9. Eastwood B, Jennings A, Harvey A (1997) A Feature Based Neural Network Segmenter for Handwritten Words. *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications, Gold Coast, Australia*, 286–290.

10. Dimauro G, Impedovo S, Pirlo G, Salzo A (1998) An Advanced Segmentation Technique for Cursive Word Recognition. In: Lee S W (ed) *Advances in Handwriting Recognition*, World Scientific Publishing, 255–264.
11. Nicchiotti G, Scagliola C, Rimassa S (2000) A Simple and Effective Cursive Word Segmentation Method. *Proceedings of the 7th International Workshop on the Frontiers of Handwriting Recognition (IWFHR-7)*, 499–504.
12. Xiao X, Leedham G (2000) Knowledge-based Cursive Script Segmentation. *Pattern Recognition Letters* 21:945–954.
13. Blumenstein M, Verma B K (2001) Analysis of Segmentation Performance on the CEDAR Benchmark Database. *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR '01)*, Seattle, USA, 1142–1146.
14. Blumenstein M, Verma B (1999) A New Segmentation Algorithm for Handwritten Word Recognition. *Proceedings of the International Joint Conference on Neural Networks (IJCNN '99)*, Washington D.C., 878–882.
15. Blumenstein M, Liu X Y, Verma B (2004) A Modified Direction Feature for Cursive Character Recognition. *Proceedings of the International Joint Conference on Neural Networks (IJCNN '04)*, Budapest, Hungary, 2983–2987.
16. Cheng C K, Liu X Y, Blumenstein M, Muthukkumarasamy V (2004) Enhancing Neural Confidence-Based Segmentation for Cursive Handwriting Recognition. *5th International Conference on Simulated Evolution And Learning (SEAL '04)*, Busan, Korea, SWA-8, CD-ROM Proceedings.
17. Gader P D, Mohamed M, Chiang J-H (1997) Handwritten Word Recognition with Character and Inter-Character Neural Networks. *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics* 27:158–164.
18. Blumenstein M, Verma B, Basli H (2003) 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, 137–141.
19. Hull J J (1994) A Database for Handwritten Text Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 16:550–554.

