Characters Segmentation of Cursive Handwritten Words based on Contour Analysis and Neural Network Validation

Fajri Kurniawan, Mohd. Shafry Mohd. Rahim, Ni'matus Sholihah, Akmal Rakhmadi & Dzulkifli Mohamad

Department of Computer Graphics and Multimedia
Universiti Teknologi Malaysia
Email: fajri_9636@yahoo.com

Abstract. This paper presents a robust algorithm to identify the letter boundaries in images of unconstrained handwritten word. The proposed algorithm is based on vertical contour analysis. Proposed algorithm is performed to generate pre-segmentation by analyzing the vertical contours from right to left. The unwanted segmentation points are reduced using neural network validation to improve accuracy of segmentation. The neural network is utilized to validate segmentation points. The experiments are performed on the IAM benchmark database. The results are showing that the proposed algorithm capable to accurately locating the letter boundaries for unconstrained handwritten words.

Keywords: *character segmentation; contour analysis; neural network validation; unconstrained handwritten word.*

1 Introduction

Despite of success achieved in numbers of hand printed segmentation and recognition over the last few decades, the recognition of off-line cursive handwriting still remains a challenging problem [1]. It is mainly due to the difficulties involved in the segmentation process of cursive handwritten words into individual characters [2].

It is known that contours are important structure features explored by researchers through different views for segmentation, feature extraction and shape matching. Ligatures (unions of characters) may be detected by locating minima in the contours (lower/upper) of word images as presented in [3]. Strathy, et al., [4] introduced contour features to segment handwritten digits. In addition, Strathy, et al. used a set of rules-based that applied on contour direction, location of holes, upper and lower baselines. The upper contours were also used to determine segmentation points in handwritten postal words as part of an entire hybrid recognition system [5]. Additionally, Bozinovic and Srihari [6] attempt to locate possible segmentation points based on proximity of minima at lower contour then some rules performed to force segmentations between two distant segmentation points. Yamada and Nakano [7] performed

Received November 16th, 2010, Revised January 28th, 2011, Accepted for publication January 28th, 2011.

character segmentation of cursive words based on contour features analysis. Recognition rates up to 91.7% were obtained when the segmentation algorithm was used as part of a complete word recognition system using 50 words lexicon size.

Recently segmentation method that based on contour was introduced in the literature. Wang, et al. [8] interestingly enhanced old technique that named drop-fall algorithm then proposed as inertial drop fall and big drop fall algorithm. Wang, et al. dealt with right contour and it must detecting starting point properly. Their method is specifically for digits only where it is true that all digits has right contour. In alphabetic characters right contour may do not appear in several characters such as ‘c’, ‘f’, ‘i’ and others characters with left contour only, therefore their methods will failed. In addition, promising correct segmentation rates up to 99.64% are reported by Omidyeganeh, et al. [9]. Nevertheless, the approach was not suitable for Latin characters since they work for Farsi/Arabic characters and dealt with upper and lower contour only. Interestingly, Cheng, et al., [10] deal with distance to upper contour from baseline to enhancing the heuristic segmenter of [11]. Additionally, upper contour of string [12] with respect to ratio of width and height features are taken to solve segmentation of touching characters problem in printed character of mathematical expressions [13].

To conclude, it is always desirous by the research community to come out with minimum number of inaccurate segmentation points and enhance accuracy with minimum computational complexity such that obtain good speed. Even though character segmentation in printed character has been accepted, segmentation on handwriting character is the most difficult problem in the domain of segmentation. In this regard, few researchers integrated intelligent techniques to achieve accurate segmentation [10,14-21]. Verma [22] proposed robust ANN validation by fusion three confidence values. During preprocessing stage they only utilized baseline detection without applied skew correction, hence baseline detection will fail when the handwriting has skew angle. Consequently, this failure will affect next stage and therefore decrease the overall segmentation accuracy. Likewise, Cheng and Blumenstein [21] implemented neural-based segmentation but with different heuristic segmenter [22]. Even it was claimed that new strategy decreased error rates, still preprocessing stage might produce segmentation problems due to adopt simple baseline detection only.

Literature is evident that contours have potential to locate segmentation points in cursive handwriting. Combining intelligent techniques enhance segmentation accuracy. Contribution of this research is introducing novel character segmentation using contour detection and adopted existing segmentation validation method using ANN to reduce segmentation errors. Initially, vertical contours are analyzed to locate prospective segmentation points in cursive handwritten word images. Finally, an ANN trained with correct and incorrect segmentation points then performed to validate correctness of the segmentation points that found by initial module.

The organization of the paper is as such section 2 presents proposed segmentation approach. Section 3 deals with implementation and experimental results. Analysis and discussion in section 4 and finally conclusion is drawn in section 5.

2 Proposed Segmentation Technique

An overview of the technique is provided in Figure 2. Firstly, input image is preprocessed and normalized. Afterward, contour detection is applied to find prospective segmentation points (PSP). Finally, PSP are fed into ANN for their validation, such that eliminate incorrect segmentation points. Feed forward back-propagation neural networks are used as an intelligent technique for validation of segmentation points. Whereas, density and direction features are extracted for input to ANN.

2.1 Preprocessing

The approach discussed in this paper requires preprocessing of word images before segmentation. Initially all images were in gray-scale format, therefore images are binarized using Otsu’s thresholding algorithm [23].

Otsu's method fully searches for the threshold that minimizes the within-class variance. It is defined as a weighted sum of variances of the two classes as follows:

$$\sigma_{\omega}^{2}(t)=\omega_{1}(t)\sigma_{1}^{2}(t)+\omega_{2}(t)\sigma_{2}^{2}(t) \tag{1}$$

Weights ω_i are the probabilities of the two classes separated by a threshold (t) and variances (σ_i^2) of these classes. Otsu defined that minimizing the within-class variance is the similar as maximizing between-class variance:

$$\sigma_{b}^{2}(t)=\sigma^{2}-\sigma_{\omega}^{2}(t)=\omega_{1}(t)\omega_{2}(t)\left[\mu_{1}(t)-\mu_{2}(t)\right]^{2} \tag{2}$$

Equation (2) is expressed in terms of class probabilities (ω_i) and class means (μ_i) which in turn can be updated iteratively

The vertical contours analysis in this proposed method is a key to locate the character boundary. Thus, slant angle estimation and correction are required [18].

(i) The slant angle (θ) of cursive word can be estimated based on chain code (n_1, n_2, n_3). This chain segment is depicted in Figure 1. The slant angle is calculated using equation 3.

$$\theta = \tan^{-1}\left(\frac{n_1 + n_2 + n_3}{n_1 - n_3}\right) \tag{3}$$

(ii) Then, a shear transformation is performed to correct the slant angle. In equation 4, *(x, y)* and *(x', y')* are the coordinates before and after the transformation respectively.

$$\begin{cases} x' = x + y\tan\theta \\ \quad y' = y \end{cases} \tag{4}$$

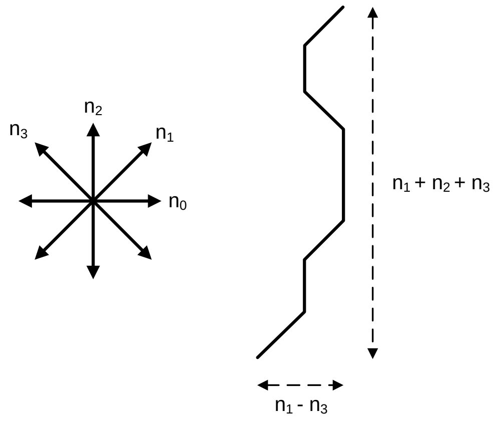


Figure 1 Chain segment.

2.2 Overview of the Contour-based Segmentation Approach

This section describes the proposed algorithm for the extraction of contours by tracing from right to left. The reason that left contour was selected because each Latin character at least has one left contour. It is different with Arabic letters that may not contains left contour so some researchers preferably used upper/lower contour to determine PSP [9]. Contour is defined as foreground pixels that do not have any other foreground pixels at right, top and bottom position up to some vertical distance. The vertical distance is calculated empirically. Figure 3 shows example of contour coordinate that found by proposed contour detection. The Contour detection algorithm is described as follows:

```
INPUT:
black_white_image <- image after thresholing
stroke_width <- average stroke width in black_white_image
traced_column <- pointer to columns of black_white_image, start
                 from most right column.
n <- 0

OUTPUT:
prospective_contour

PSEUDOCODE:
while traced_column > 0
  If sum_of_pixels on traced_column <= 0
    traced_column <- traced_column - 1
  else
    find vertically connected pixels (VCP) on traced_column
  if VCP has no pixel on right side
    candidate_contour <- middle pixel of VCP
    if candidate_contour is not inside hole/concave
      n <- n + 1
      prospective_contour(n) <- candidate_contour
    else
      force_counter <- force_counter + 1
    end if
  else
    if force_counter >= stroke_width
      n <- n + 1
      prospective_contour(n) <- bottom pixel of traced_column
    end if
  end if
  traced_column <- traced_column – stroke_width
end while
```

Naturally, the ligatures should have a minimum number of pixels when cut vertically compare to a character. In addition, heuristic segmentation also keeps number of prospective segmentation points as less as possible for further validated by ANN validation module. Since contour analysis produce numerous prospective contours, thus number of prospective segmentation points are also having great amount. Several prospective segmentation points might unnecessary due to the position of them are closed to each other. This situation is contributed when locating prospective contour based on average stroke width. Even average stroke width is the minimum width of a character. Nevertheless, there are still some gaps between characters. It is still acceptable to eliminate some segmentation points due to their distance are too close. In this regards, proposed heuristic segmentation rule is defined as follows:

(i) Let *PC* is a set of column position *(c)* of prospective contours denoted as $PC = \{pc \in c \mid c = 1,...,w\}$ with *n=length(PC)*, where *w* and *n* are width of the image and number of prospective contour, respectively. Taking pc_k with $k = 1,...,n$, afterward tracing vertical cutting from contour column position pc_k to $pc_{k+1} - t$ by taking into account number of cutting stroke (*cs*) and vertical projection (*vp*). Whereas, *t* is the average stroke width. Finally, the column between $[pc_k, pc_{k+1} - t]$ with minimum *cs* and *vp* are considered as prospective segmentation points. The set of prospective segmentation points is denoted as:

$$PSP = \{psp \in c \mid c = pc_1,..., pc_n, n = length(PC)\}$$

(ii) Reduce *PSP* with *n* segmentation points to become equal or less than estimated number of characters. In this regard, two-stage iterations are introduced to reduce number of segmentation. First iteration will eliminate one of two segmentation points with distance less than average stroke width. Similarly, second iteration will reduce number of segmentation points with respect to dynamic distance value that taken from minimum distance of current adjacent *PSP*.

Prospective segmentation points can be determined using this proposed contour-based segmentation approach. As describe above, PSP should has minimum stroke cutting, which detected by tracing from upper to lower image boundary of contour coordinate to the right side. The experiment is conducted using a test set that taken from IAM database. Each word image is fed to contour detection, and then the results are analyzed. Several results of contour detection that applied on an unconstrained handwritten word are presented below.

2.3 ANN Training for Segmentation Points Validation

To make the technique robust, neural assistance was acquired. In this regard, multilayer feedforward neural network with the back propagation was employed. Blumenstein and Verma [21] reported that ANN validation reduced segmentation errors. Features extracted for ANN input were quite simple and were based on density of the segmentation areas only. Consequently, such simple features couldn’t distinguish between actual ligatures and stroke of alphabet likes ‘m’ or ‘w’[15]. To make ANN decision precise, authors have enhanced the ANN validation by adding direction features.

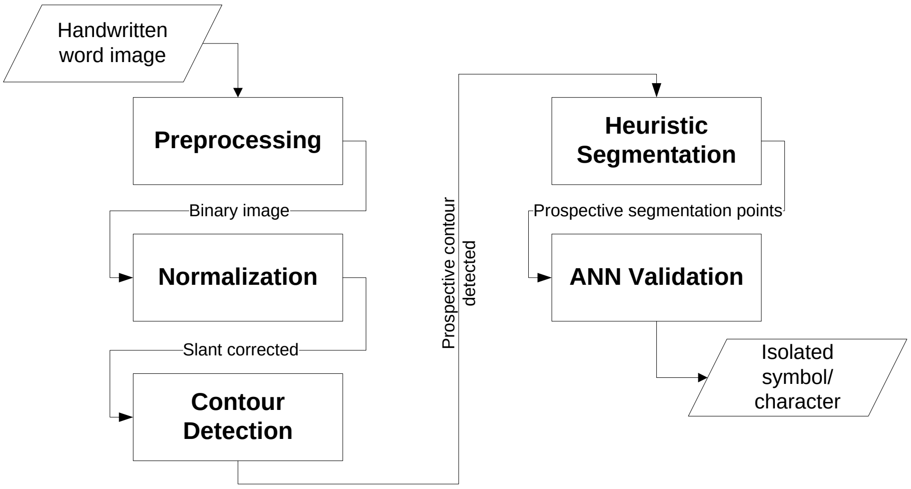


Figure 2 Overview of segmentation approach.

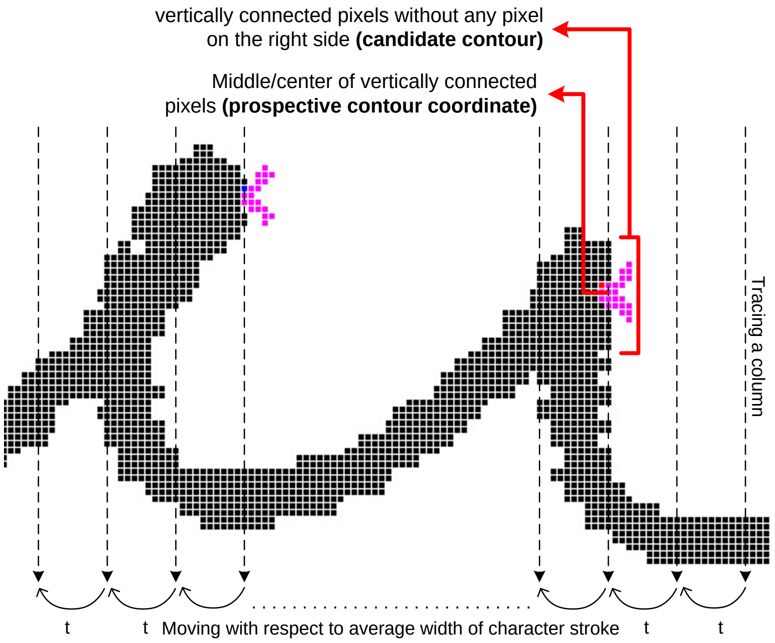

vertically connected pixels without any pixel on the right side **(candidate contour)**
Middle/center of vertically connected pixels **(prospective contour coordinate)**
Tracing a column
t
t
Moving with respect to average width of character stroke
t
t

Figure 3 Contour coordinate extracted using proposed contour detection.

Enhanced features are proposed to improved ANN validation that based on direction. Using these features the shape of object can be represented in an effective and reliable way. This feature has been used by Blumenstein, et al.

[17] for character recognition. It was reported that using direction features recognition rates increased. In this paper, authors have simplified their technique by eliminating line type normalization and feature vector formation. Since authors neither required smoothing nor required to reduce the feature vector. In the main, four labels are used to encode the matrix of pixels as follows: Vertical line segment - 2, Right diagonal line - 3, Horizontal line segment - 4 and Left diagonal line - 5. It has been describe in more detail in [17]. Figure 4 describes features extraction of density and direction features.

To achieve optimal results, ANN input feature vector size was set using zoning [24] to 160-300 features experimentally. Meanwhile, output of ANN is divided into two separate classes of correct and incorrect segmentation points.

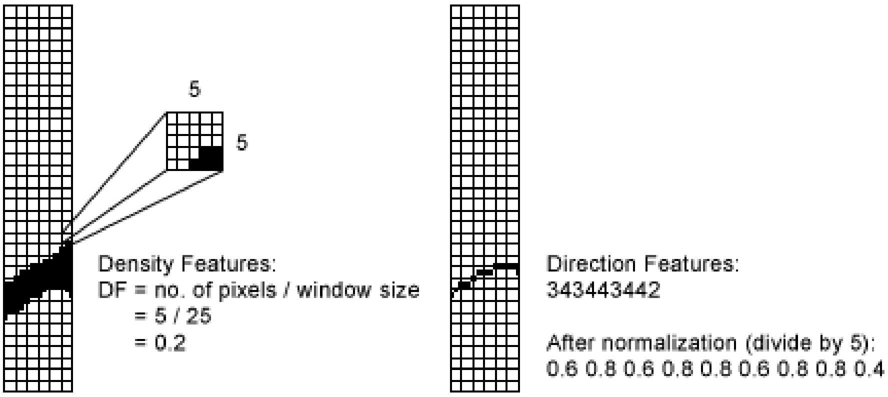


Figure 4 Hybrid features extraction.

3 Experimental Results

Following ANN training, for testing 317 words (consist of 1,683 characters) were selected from IAM database that were not used in training phase. The word images are segmented using contour based approach and the segmentation points were fed to ANN for their classification into valid and invalid without employing human factor. The invalid segmentation points were rejected and the correct were retained only.

Some result of preprocessing stage is presented in Figure 5. The result shows either left-slant or right-slant can normalized properly.

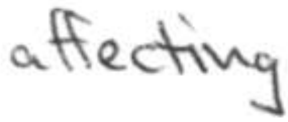
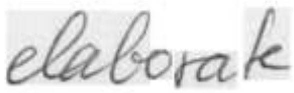
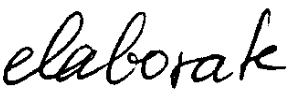
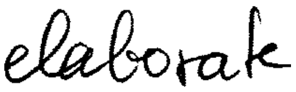
Original	Thresholding	Normalization
		
		

Figure 5 Several word images after preprocessing stage

It should be noted that normalization scheme is not adopted in this experiment. Figure 6 depicted the detected contours. After performed heuristic segmentation, the PSP may contain incorrect segmentation points. It is required to validate it using robust technique such as ANN. Some segmentation result is presented in Figure 7.

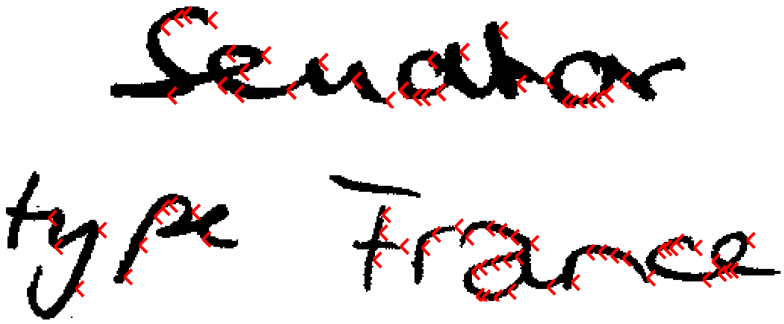


Figure 6 Contour detection on cursive word.

Proposed contour based segmentation approach performed well and put forward all possible segment points for cursive handwritten word. Some characters were over-segmented as shown in Figure 7.

The ANN is trained with 6,417 training patterns (valid and invalid segmentation points) taken from 1,000 words of IAM benchmark database [25]. Table 1 presented some ANN accuracy with respect to number of features. During experiment, it found that optimal ANN structure contains 214 neurons in the input layer, 1 hidden layer with 90 hidden neurons and one neuron as output layer. Those parameters are selected by taking into account the error rates of ANN. Error rates are calculated by tested the ANN with validation data that taken randomly from trained data. ANN structure with minimum error rates value is considered as optimal configuration.

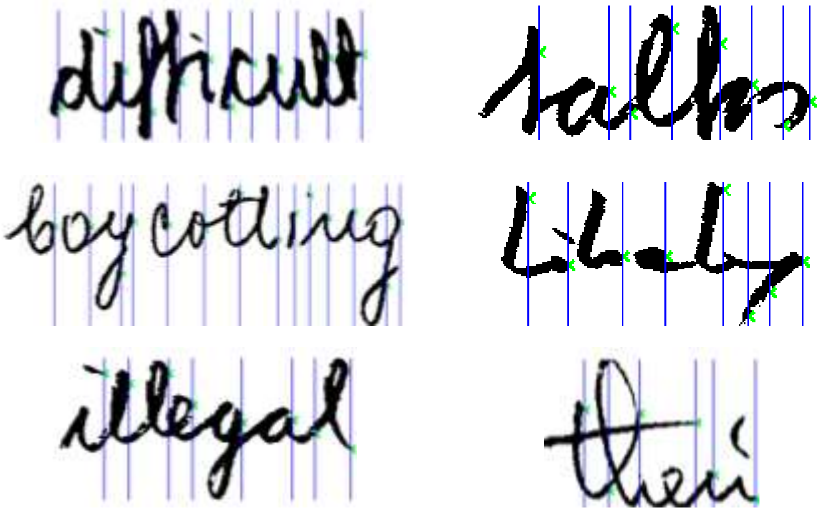


Figure 7 Prospective segmentation points based on contours analysis.

Table 1 Accuracy of neural validation.

Type	Matrix Size	Density + Direction Features	Accuracy (%)
I	90 x 45	162 + 4	81.79
II	100 x 50	200 + 4	86.57
III	105 x 50	210 + 4	86.61
IV	110 x 55	242 + 4	85.27
V	115 x 60	276 + 4	85.90

As mentioned before, the feature vector size is 210 density features plus 4 direction features. Finally, normalized feature vectors are feed into the neural validation, and then the neural validation responds by assign confidence value on each segmentation points. Since the neural validation is trained with the target values 0.1 and 0.9 and using sigmoid function, so it produces an output between 0.0 through 1.0. Thus, a value must be defined as decision's threshold value. Segmentation points with confidence value that greater than 0.5 is considered as correct segmentation. Meanwhile, segmentation points with confidence value that equal or less than 0.5 is rejected.

Prospective segmentation points are generated following contour based segmentation along with reduction scheme. Afterward, entire prospective segmentation points are validated using neural validation. Rejected segmentation points are denoted by arrowhead. Numerous incorrect segmentation points can be rejected accurately and correct segmentation points left as validated segmentation points. Neural validation shows promising accuracy in normal writing style. Neural validation result is shown in Figure 8.

Commonly, there are three criterions [15] that can be used to test the accuracy of the segmentation approach. Those are 1) number of over segmentations, 2) missed segmentations and 3) bad segmentations. Over segmentation is indicated when the character is segmented by more than two segmentation points. Missed segmentation is denoted when a correct segmentation point is not pointed by the segmentation approach. Bad segmentation means the segmentation point does not separate two characters properly. The error rates are shown in Table 2. Figure 10 showed final segmentation result of proposed approach.

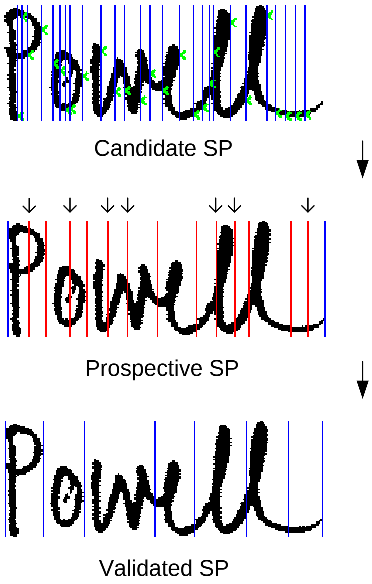


Figure 8 A word image was segmented using proposed method followed by neural validation.

4 Analysis and Discussion of Results

Segmentation algorithm based on contour analysis in conjunction with neural network achieved recognition rate 82.63% for valid identification of 1,902 pattern of segmentation point. It is showed that using enhanced features extractor the segmentation rates comes better. This can be explaining that direction features can distinguish stroke of ligatures/union and stroke of characters using information of its direction. Proposed contour detection can provide enough prospective segmentation points for further feed to ANN validation.

Two problems are found during the analysis of the results. Firstly, as seen in Figure 7, heuristic segmenter based on contour always over-segment character 'm', 'n' or 'u'. This problem is common since heuristic segmenter cannot distinguish its context. It can be solved using intelligent technique to validate segmentation points as drawn above. Secondly, touched/overlapped characters caused for miss-segmentation errors. This type of problem is very hard to deal with. When two characters are tight together, ligatures cannot be found and therefore they cannot be segmented. Nevertheless, touching characters is rare case in cursive handwriting. Hence overall correct segmentation results are not significant affected.

As seen in Figure 6, each character has numerous detected contours that denoted by arrowhead. Most of the characters with loop come out approximately with more than five detected contours. This situation is typical because the force mechanism on contour detection. Even though, the main goal of contour detection can be reached, since prospective contours that closest to the character boundary are properly detected. It is still difficult to choose the best prospective contour from numerous detected contours. It is important to reduce number of detected contours, because too many contours will raise the error rate and add computational cost. In this regard, heuristic rule is investigated.

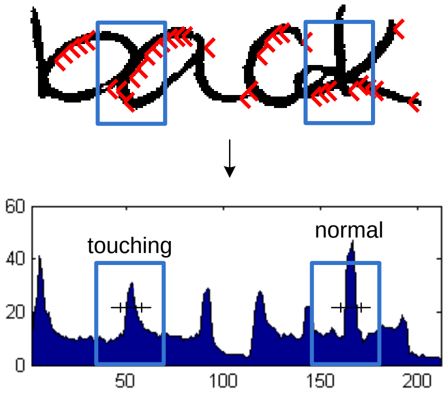


Figure 9 Contour detection on word level with touching character.

Some result of contour detection on typical handwriting has been discussed in section II. There is also uncommon handwriting style that exists in real situation. Where, some characters are touched or joined such that similar to a single character. This problem is known as touching character problem. Touching character problem is an extraordinary case in handwriting domain. Conversely, if it exists on a handwritten word, then it might disturb a character segmentation method and leads to incorrect segmentation.

Figure 9 depicted a result of contour detections on word samples with touching characters. Touching region is denoted by the square box. Vertical projection analysis also carried out in Figure 9. It confirms that the contour detection not possible to locate segmentation point on touching region because projection of touching region has similar projection of character stroke. Both projections have comparable thick and height.

Based on experiment, contour detection is unable to locate a contour at touching portion. On the other hand, it is only able to detect prospective contours that closest to the touching portion. Hence, it is impossible solving the problem using contour detection.

Table 2 Segmentation errors.

Over-segmentation	**Missed-segmentation**	**Bad-segmentation**
8.83	0.17	8.37

5 Conclusion and Future Works

A novel intelligent vertical contour based character segmentation approach has been presented in this paper that produces encouraging results. It is used to segment difficult cursive words taken from IAM benchmark database. The performance of word segmentation on small database is very promising. This study also described where touching character can disturb segmentation accuracy.

For future works, it can be extended to include additional improvement of contour based segmentation such as accurate character width estimation method, and noise removal such as for unwanted line stroke.

Some special cases of character segmentation problems, such as touching with two, three or more characters, touching character that overlapping is also challenging.

The artificial neural network classifier in neural validation can be substituted with other classifier variants such as support vector machine (SVM), hidden markov model (HMM), etc and another robust feature can be explored.

Different dataset from real script is suggested. Finally, it is hoped that this research will bring a significant advancement to the character segmentation domain.

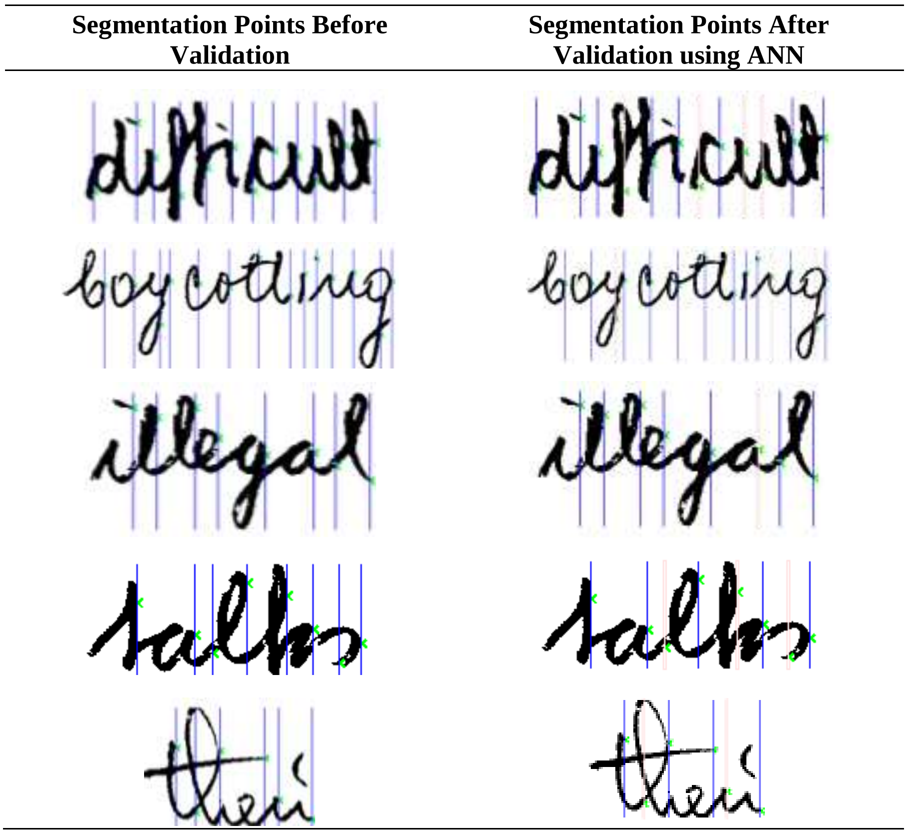

Segmentation Points Before Validation
Segmentation Points After Validation using ANN

Figure 10 Final segmentation result. On right column, some white slice shows invalid/rejected PSP.

Acknowledgement

The researchers would like to acknowledge the Research Management Center (RMC), Universiti Teknologi Malaysia (UTM) and FRGS Grant for the financial funding through Vote 78568.

References

[1] Oliveira, L.S. & Sabourin, R., *Support Vector Machines for Handwritten Numerical String Recognition*, 9th International Workshop on Frontiers in Handwriting Recognition, Kokubunji, Tokyo, Japan, October 26-29, pp. 39-44, 2004.

[2] Bortolozzi, F., Souza, A., Britto Jr., Oliveira, L.S. & Morita, M., *Recent Advances in Handwriting Recognition*, Document Analysis, Editors: Umapada Pal, Swapan K. Parui, Bidyut B. Chaudhuri, pp. 1-30, 2005.

[3] Holt, M., Beglou, M., & Datta, S., *Slant-Independent Letter Segmentation for Off-line Cursive Script Recognition*, From Pixels to Features III, S. Impedovo and J.C. Simon (eds.), Elsevier, **41**, 1992.

[4] Strathy, N.W., Suen, C.Y. & Krzyyzak, A., *Segmentation of Handwritten Digits using Contour Features*, ICDAR '93, pp. 577-580, 1993.

[5] Kimura, F., Shridhar, M. & Narasimhamurthi, N., *Lexicon Directed Segmentation-Recognition Procedure for Unconstrained Handwritten Words*, Pre-Proceedings IWFHR III, Buffalo, pp. 122, 1993.

[6] Bozinovic, R.M. & Srihari, S.N., *Off-line Cursive Script Word Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **11**(1), 68-83, 1989.

[7] Yamada, H. & Nakano, Y., *Cursive Handwritten Word Recognition Using Multiple Segmentation Determined by Contour Analysis*, IEICE Transactions on Information and Systems, E79-D, 464-470, 1996.

[8] Wang, X., Zheng, K. & Guo, J., *Inertial and Big Drop Fall Algorithm*, International Journal of Information Technology, **12**(4), 39-48, 2006.

[9] Omidyeganeh, M., Azmi, R., Nayebi, K. & Javadtalab, A., *A New Method to Improve Multi Font Farsi/Arabic Character Segmentation Results: Using Extra Classes of Some Character Combinations*, Springer-Verlag Berlin Heidelberg, 670-679, 2007.

[10] Cheng, C.K., Liu, X.Y., Blumenstein, M., & Muthukkumarasamy, V., *Enhancing Neural Confidence-Based Segmentation for Cursive Handwriting Recognition*, 5th International Conference on Simulated Evolution and Learning, Busan, Korea, SWA-8, 2004.

[11] Verma, B., *A Contour Code Feature Based Segmentation for Handwriting Recognition*, Proceedings of 7th International Conference on Document Analysis and Recognition (ICDAR'03), pp. 1203-1207, 2003.

[12] Heutte, L., Paquet, T., Moreau, J.V., Lecourtier, Y. & Olivier, C., *A Structural/Statistical Feature based Vector for Handwritten Character Recognition*, Pattern Recognition Letters, **19**(7), 629- 641, 1998.

[13] Tian, X. & Zhang, Y., *Segmentation of Touching Characters in Mathematical Expressions Using Contour Feature Technique*, Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 206-209, 2007.

[14] Blumenstein, M. & Verma, B., *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.

[15] Blumenstein, M. & Verma, B., *Analysis of Segmentation Performance on the CEDAR Benchmark Database*, Proceedings of 6th International

Conference on Document Analysis and Recognition, pp. 1142-1146, 2001.

[16] Eastwood, B., Jennings, A. & Harvey, A., *A Feature Based Neural Network Segmenter for Handwritten Words*, Int'l Conf. Computational Intelligence and Multimedia Applications, Gold Coast, Australia, pp. 286-290, 1997.

[17] Blumenstein, M., Liu, X.Y. & Verma, B., *A modified direction feature for cursive character recognition*, Proceedings on the IEEE International Joint Conference on Neural Networks, **4**, pp. 2983-2987, 25-29 July, 2004.

[18] Cheng, C.K. & Blumenstein, M., *The Neural Based Segmentation of Cursive Words Using Enhanced Heuristics*, Proceedings of the 8th International conference on Document analysis and recognition, **2**, pp. 650-654, 2005.

[19] Cheng, C.K. & Blumenstein, M., *Improving the Segmentation of Cursive Handwritten Words using Ligature Detection and Neural Validation*, Proceedings of the 4th Asia Pacific International Symposium on Information Technology (APIS 2005), Gold Coast, Australia, pp. 56-59, 2005.

[20] Ghosh, M., Ghosh, R. & Verma, B., *A Fully Automated Offline Handwriting Recognition System Incorporating Rule Based Neural Network Validated Segmentation and Hybrid Neural Network Classifier*, International Journal of Pattern Recognition and Artificial Intelligence, **18**(7), 1267-1283, 2004.

[21] Blumenstein, M. & Verma, B., *Neural-based solutions for the segmentation and recognition of difficult handwritten words from a benchmark database*, Proceedings of the Fifth International Conference on 5th International Conference on Document Analysis and Recognition, Bangalore, India, 20-22 Sep, pp. 281-284, 1999.

[22] Verma, B., *A Contour Character Extraction Approach in Conjunction with A Neural Confidence Fusion Technique for the Segmentation of Handwriting Recognition*, Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'O2), **5**, pp. 2459-2463, 2002.

[23] Otsu, N., *A threshold selection method from gray-scale histogram*, IEEE Transactions on Systems, Man, and Cybernetics **9**(1), 62–66, 1978.

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

[25] Marti, U, & Bunke, H., *The IAM database: An English sentence database for off-line handwriting recognition*, International Journal of Document Analysis and Recognition, **15**, 65-90, 2002.