© 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Over-Segmentation and Neural Binary Validation for Cursive Handwriting Recognition

Hong Lee and Brijesh Verma

Abstract - A novel Over-Segmentation and Neural Binary Validation (OSNBV) is presented in this paper. OSNBV is a character segmentation strategy for off-line cursive handwriting recognition. Unlike the approaches in the literature, OSNBV is a prioritized segmentation approach. Initially, OSNBV oversegments a handwritten word into primitives. Neural binary validation is iteratively applied to the primitives. The outcome of each iteration is to join two neighboring primitives when the joined one improves the global neural competency. OSNBV introduces Transition Count (TC) and TC for English (EngTC) to prevent under-segmentation error during neural binary validation. OSNBV also incorporates Transition Count Matrix (TCM) into neural global competency. The proposed approach has been evaluated on CEDAR benchmark database. The results showed a significant improvement in segmentation errors. The analysis of results showed that the inclusion of TCM into the validation function has played a major role in improving oversegmentation and bad-segmentation errors.

I. INTRODUCTION

Off-line Cursive Handwriting Recognition (OffCHR) is an automatic process to convert an input handwritten document image into computer-recognizable character representations. OffCHR has been active research domain for decades, and industrial beneficiaries have been trying to automate repetitive manpower oriented tasks such as processing postal address, bank checks, form data, historical manuscripts, etc [1]. Despite sleepless research in OffCHR for decades, the performance of the state-of-the-art OffCHR is below the industrial standard to accommodate the real world problems. The researchers in this field agree that the main contributor of the low OffCHR performance is the segmentation [2-9].

Segmentation is a process to discriminate each letter from others, prior to recognition into electronic character representations. A subimage bound by two neighboring boundaries is called a segment. Segmentation precedes the recognition. In other words, the recognition process is based on the outcomes of the segmentation process. It implies that better recognition performance can be achieved on better segmentation outcomes. However, the segmentation is very troublesome process because of the nature of handwriting, which projects the informality of handwritten characters. As matter of fact, the researchers have not found any rule of thumb segmentation methods on handwritten images. That's why the segmentation process has become a notorious and chronicle contributor to OffCHR field [10-15].

Many researchers have been tackling the segmentation of handwritten image in various approaches. Tripathy et al. [16] incorporated water reservoir approach to segment the connected characters in Oriya text recognition. Zhao et al. [17] proposed

Hong Lee is with CQUniversity, Australia (phone: +61 7 4150 7052; email: h.lee1@cqu.edu.au).

Brijesh Verma is with CQUniversity, Australia (phone: +61 7 4930 9058; email: b.verma@cqu.edu.au).

background thinning segmentation algorithm to segment connected Chinese characters. Alhajj et al. [18] proposed multi-agents to segment handwritten connected digits. Liang et al. [19] proposed a meta synthetic approach to segment handwritten Chinese character strings. Dawoud [20] proposed the iterative cross section sequence graph (ICSSG) for handwritten character segmentation. Renaudin et al. [21] proposed over-segmentation and graph construction technique to segment touching digits. Suwa [22] proposed graph representation technique to segment multiply connected digits.

The remainder of this paper is organized to describe the proposed research methodology in section II. The experiments and results are presented in section III, and the analysis and discussion of the results are explained in section IV. Finally, the section V concludes this paper.

II. PROPOSED RESEARCH METHODOLOGY

The proposed OSNBV over-segments an input word into primitives, and applies neural binary validation to the primitives iteratively. The details of OSNBV are described in following subsections.

A. Overview

Overall system architecture of the proposed approach OSNBV is described in Figure 1.



Figure 1. Overview of Over-Segmentation and Neural Binary Validation (OSNBV)

B. Generating primitives

The core idea of introducing over-segmentation into OffCHR is not to miss any letter boundaries. So, successful over-segmentation generates a segmentation set containing all letter boundaries regardless existence of excessive segmentation points, which are

Manuscript received April 27, 2010.

called over-segmentation points. The best way to increase the chances to have successful over-segmentation is to locate as many segmentation points as possible. Often many rules and heuristics are applied to achieve successful over-segmentation. Every segmentation point from over-segmentation has chance to be a correct segmentation point, so it's called Suspicious Segmentation Point (SSP).

In the proposed approach, the SSPs are generated by using vertical foreground pixel density and stroke thickness variable. The stroke thickness is the most occurring continuous foreground pixel count. It is measured by scanning the segmenting word vertically and horizontally. While scanning, the occurrences are recorded and the most occurring continuous foreground pixel count becomes the stroke thickness of the segmenting word. The details of the stroke thickness is estimated, the SSPs are located where the vertical foreground pixel density is less than the stroke thickness. However, to increase chance of locating the correct boundaries, the SSPs are located where the vertical foreground pixel density is less than three times of the stroke thickness. The continuous SSPs are consolidated as a single SSP by finding the one in the middle.

The suspicious segmentation points are screened by hole detection module to remove the ones crossing hole regions. The reason to incorporate this screening process is two-fold. Firstly, reducing the number of SSP cuts down the computational costs significantly. The computational cost of validation for a SSP is much cheaper than validation by classifier in the later stage. The second reason is to reduce the number of primitives. A primitive is a sub-image defined by two neighboring SSPs. The lesser primitives, the lesser spatial primitive combinations to validate.

C. Neural Binary Validation (NBV)

Neural binary validation is iteratively applied to a set of primitives created from the over-segmenting module. The primitives are more likely to be partial characters at this stage. Therefore, those partial character-like primitives need to be combined with the neighboring primitives and become a whole character. In each iteration, NBV creates evaluation sets. Neural confidence function is applied to each of them to pick one evaluation set with the highest value. When the selected evaluation set is valued higher than the current set, it is passed as the current set for the next iteration. Otherwise, it terminates iteration and outputs the current set as the final outputs. Each final primitive should represent a whole character.

1) Creating evaluation sets

Let P be a current set of primitives, $P = \{P_1, P_2, P_3 \dots P_n\}$, where n is the number of primitives in P. Considering that the '+' sign indicates joint between two neighboring primitives, the maximum of (n - 1)evaluation (E) sets can be expressed as follows:

$$E_{n-1} = \{P_1, P_2, P_3 \dots P_{n-1} + P_n\}$$

However, a joint between two neighboring primitives are allowed only if a joint condition is satisfied. The joint condition concerns Transition Count (TC) and TC for English (EngTC). The details are described in the following subsections.

a) Transition Count (TC)

TC indicates the maximum number of horizontal black-to-white pixel transition for a row of pixels from a primitive. To obtain TC from a primitive, rastering technique is used for scanning the primitive. A primitive is an image, which can be represented into two dimensions, rows and columns of pixels. Therefore, there are as many TC as the number of rows for a primitive. The highest TC among rows represents the TC for the primitive.

b) Transition Count for English (EngTC)

As mentioned earlier, a joint condition of two neighboring primitives into one is to compare TC of the combined primitive to a preset variable EngTC. The use of EngTC is to prevent from producing under-segmentation error during validation. As shown in Table 1, TC for 52 classes of English alphabets has been generated from presegmented handwritten character images. The pre-segmented character images are the same data used to train two-class neural networks classifier. From the table, EngTC is assigned with 5; because the highest value of the table is 5 from 'M' or 'm'. Therefore, any joint between two neighboring primitives are prohibited if TC of the combined primitive is higher than 5.

TABLE 1. TRANSITION COUNT (TC) FOR EACH ALPHABET CLASS GENERATEDFROM PRE-SEGMENTED CHARACTER IMAGES USED TO TRAIN NEURALCLASSIFIER.

Class	тс	Class	ТС	Class	ТС	Class	ТС
Α	4	g	3	N	4	t	3
а	4	Н	4	n	4	U	2
В	4	h	3	0	3	u	4
b	3	Ι	3	0	4	V	3
С	3	i	1	Р	3	v	3
с	3	J	4	р	3	W	4
D	3	j	3	Q	3	w	4
d	4	K	4	q	3	Х	3
Е	3	k	3	R	4	х	3
e	4	L	3	r	3	Y	3
F	3	1	2	S	3	у	3
f	3	Μ	5	s	3	Z	3
G	4	m	5	Т	4	Z	2

2) Calculating confidence for evaluation sets

To calculate confidence value for each evaluation set, the proposed approach utilizes the Neural Networks (NN) outputs. The confidence function, Conf(P), indicates a Class1 output value from two-Class neural networks when a primitive, P, is input. Class1 of two implies the likelihood of being a correct character, and Class2 indicates the likelihood of being rubbish (non-character). Let GlobalConf(W) be a confidence function where W is an evaluation set of n primitives. GlobalConf(W) is sought by calculating the average:

$$W = \{P_1, P_2, P_3 \dots P_n\}$$

GlobalConf(W) = $\frac{\sum_{k=1}^{n} \text{Conf}(P_k)}{2}$ (1)

Therefore, an evaluation set with the highest confidence value, E_{max} , can be obtained by:

$$E_{max} = \frac{ArgMax}{e \in E} GlobalConf(e),$$

where e is an element of evaluation sets. E

a) Transition Count Matrix (TCM)

In the previous section, TC and EngTC were introduced to prevent under-segmentation errors during validation. In this section, the use of TCM is explained to boost the competency of primitives by being multiplied to neural network outputs of primitives inside evaluation function. The ideal outcome of introducing this feature is to reduce overall over-segmentation and bad-segmentation errors. The method to measure TC of a primitive is the same as explained previously. However, the distinguishing difference of TCM is that it is measured in both directions; vertical as well as horizontal; by the way, TC was measured in horizontal direction only. TC was expressed as a single digit, which representing horizontal count. However, TCM is expressed in double digit format. For example, 34 is a value of TCM. The 3 is for the horizontal count and the 4 for the vertical count of a primitive.

To model the pattern of TCM for English language, it requires the pre-segmented character images. Therefore, the same database was used to model TCM for English. Similar to obtaining EngTC from the database, TCM for each sample is acquired and recorded. The pattern of TCM and the number of samples per pattern for each class have been extracted. After obtaining the patterns of TCM for each class, the global TCM can be repopulated in the order of TCM patterns. The global population and normalized value for each TCM pattern has been show in Table 2. For example, the pattern of TCM 44 has 4 samples, and its normalized value is 0.0032. The pattern and its normalized value of a primitive can be incorporated into evaluation function in equation (1). Let M(P) be a TCM function when P is a primitive:

$$GlobalConf(W) = \frac{\sum_{k=1}^{n} \{Conf(P_k) \times M(P_k)\}}{n}$$
(2)

TABLE 2. PATT = PATTERN, POP = POPULATION, NORM = NORMALIZED. IT SHOWS GLOBAL TCM ACROSS THE 52 CLASSES. THE PATTERNS OF TCM HAVE BEEN EXTRACTED POPULATED AND NORMALIZED

BEER		CENTED TITE TO			
Patt	Рор	Norm	Patt	Рор	Norm
44	5	0.0032	12	36	0.0231
22	432	0.2769	21	29	0.0186
35	3	0.0019	43	41	0.0263
23	358	0.2295	42	59	0.0378
36	3	0.0019	41	3	0.0019
33	232	0.1487	32	171	0.1096
24	42	0.0269	31	18	0.0115
34	53	0.034	51	1	0.0006
25	12	0.0077	52	3	0.0019
13	12	0.0077	53	5	0.0032
14	1	0.0006	63	1	0.0006
11	38	0.0244	54	2	0.0013

3) Termination of OSNBV

The improvement is made when the confidence value of an evaluation set with the highest confidence value (GlobalConf(E_{max})) is greater than the confidence value (GlobalConf(S)) of the current set, S. Then, the evaluation set is promoted as a current set for next iteration.

Let E_{max} and P be an evaluation set with the highest confidence value and a current searching set for the iteration respectively. An improvement is made:

IF GlobalConf(
$$E_{max}$$
) > GlobalConf(P)
THEN P = E_{max}

Neural Binary Validation (NBV) will be repeated until certain conditions are erupted. There are two terminating conditions enforced. One of them occurs when there is no evaluation set to be created from the current set. No evaluation set will be produced when the current set has only single primitive or when joints are denied due to the joint condition. The other enforces NBV to terminate when there is no improvement. It ensures that the current set would be the best combinations of primitives.

4) OSNBV in Steps

- Line 1: Create primitives from input word
- Line 2: Create evaluation sets from the current set
- Line 3: Find the most competent evaluation set
- Line 4: Decide whether the competency of the selected evaluation set is improved over the current set
- Line 5: If improved, assign the selected evaluation set to the current set. Go to Line 2.
- Line 6: Output the current set as final primitives

An example of the overall SNBV is shown in Figure 2. In the figure, a word 'st' is over-segmented. Then, the primitives are fed into neural binary validation module for verification.



Figure 2. An example of the proposed methodology, Over-Segmentation and Neural Binary Validation (OSNBV) on a handwritten image word 'st'

III. EXPERIMENTS AND RESULTS

A. Implementation

In the proposed system, all algorithms have been implemented in Java programming language using object oriented principles.

B. Database Preparation

CEDAR is well known benchmark database for cursive handwriting recognition. The proposed approach was evaluated on the CEDAR database. The 311 words from CEDAR/TEST/CITY/BD were extracted and the proposed algorithm was applied to each word. The

results were produced by manual inspection of the final segmentation outputs.

C. Neural Networks Training

A MLP neural network with a single hidden layer was trained on presegmented characters with back-propagation learning algorithm. It takes 100 inputs, and produces 2 outputs. The two outputs represent Class1 and Class2. Class1 is likelihood of being correct character and Class2 is of being rubbish character. Training data for Class1 was pre-segmented characters from CEDAR/TRAIN/BINANUM directory. The number of training samples for Class1 includes 1560 pre-segmented character images. In English, there are 52 alphabet letters. Therefore, 1560 images consist of 30 images per each alphabet letter. Training data for Class2 was generated by oversegmenting words from CEDAR/TRAIN/CITY directory. From the over-segmentation, 1560 rubbish look-like primitives were manually selected for Class2 training data to balance with Class1 training data. The number of hidden units and the number of iteration were incremented by an interval for training. With optimal parameter configuration, the neural classifier was trained to achieve around 79% classification accuracy at most and used in the experiments.

D. Use of TC and EngTC

The primary use of TC is to prevent re-introducing undersegmentation errors during validation. TC and EngTC are the working pair to reduce under-segmentation errors. For examples, two neighboring primitives are combined into a single primitive. TC of the combined primitive is extracted. If the TC is greater than EngTC, the combination is not allowed; because it is regarded as containing more than a single character. To measure the effectiveness of the TC strategy, two criteria were used. The first one is to count the number of occurrences of prevention. The second one is to inspect if the prevention is correct or not. Prevention is defined as correct if the two neighboring primitives belong to different character segments. On the other hand, prevention is regarded as incorrect if the two neighboring primitives belong to a same character segment.

The experiment was conducted by applying the proposed approach to the 311 words described earlier in database section. The variable EngTC is assigned with the maximum TC from Table 1. While segmenting 311 words individually, the number of occurrences of prevention was recorded. The prevented primitives were manually inspected to check if the preventions were correct or not.

In Table 3, total 36 times of joints were prevented by the use of TC. 35 out 36 were correct prevention and only one joint between two neighboring primitives from a same character segment was prevented. In other words, 35 under-segmentation errors were prevented during validation by EngTC strategy.

 TABLE 3. THE EFFECTIVENESS OF USING ENGTC STRATEGY TO PREVENT FROM

 RE-INTRODUCING UNDER-SEGMENTATION ERRORS DURING VALIDATION.

EngTC	No. of Correct Prevention	No. of Incorrect Prevention
5	35	1

E. Segmentation performance

One of the segmentation accuracy measurements is to check the segmentation errors. Achieving higher segmentation accuracy means lowering segmentation errors. The segmentation errors are categorized into three types such as under-segmentation, over-segmentation and bad-segmentation errors [15]. The under-segmentation error is defined if segmentation boundary is missed between two neighboring characters. The over-segmentation error

happens if a character segment is divided into more than two (exclusive) primitives. The bad-segmentation error is defined as incorrect segmentation boundary, which does not belong to undersegmentation or over-segmentation errors.

Two experiments were conducted. The first experiment was to applying the OSNBV with equation (1) individually to the 311 CEDAR words described in database section. The second experiment was to applying the OSNBV with equation (2) to the same database. From the final segmentation output of each word, the segmentation errors were accumulated by manual inspection. For the both experiments, the parameter, EngTC, was statically assigned with the maximum value, 5, from Table 1. The figures for the segmentation errors in Table 4 are calculated by dividing each categorical accumulated error by the total number of characters in 311 words. In the first experiment, the highest segmentation error was caused by over-segmentation, followed by bad-segmentation and undersegmentation in order. The overall average error rate was 6.94% for the first experiment. From the second experiment, the highest error was caused by bad-segmentation error, followed by oversegmentation and under-segmentation respectively. The overall average error from the second experiment was 3.35%.

TABLE 4. FINAL SEGMENTATION RESULTS OF THE PROPOSED APPROACH	
--	--

Exposimonta	Classifian	No. cforenda	Segmentation Error (%)			
Experiments	Classifier	No. of words	Under	Over	Bad	Average
OSNBV without TCM	79%	311	1.96	11.55	7.33	6.94
OSNBV with TCM			1.96	2.76	5.32	3.35

IV. ANALYSIS AND DISCUSSION

As shown in Table 4, it is obvious that the 1.96% of undersegmentation error was re-introduced by the proposed neural binary validation by combining two neighboring primitives, which belongs to the different character segment. However, the under-segmentation error is very minor compared to over or bad in both experiments. Especially, over segmentation error in the first experiment was the highest. The over and bad-segmentation errors during the proposed neural binary validation can be contributed by getting incorrect confidence value of primitives from neural classifier. The proposed neural binary validation entirely depends on the confidence value from neural classifier apart from the interruption by TC strategy. The employed neural classifier was trained up to 79% classification accuracy. However, it was very challenging to model the rubbish class, which virtually has no pattern. Therefore, rubbish look-like patterns could be unknown to the classifier because they weren't included in the training data. Those unknown pattern could give incorrect output values.

The use of TC and EngTC was investigated to find out the effectiveness of under-segmentation error prevention. The performance was measured by counting the number of correct and incorrect prevention. Through the experiment, 36 preventions were discovered. 35 preventions were correct and 1 was incorrect. Therefore, the use of TC and EngTC was very effective to prevent under-segmentation errors.

In the proposed approach, TCM was modeled from pre-segmented characters and the use of TCM was injected into evaluation function during validation. The ideal goal of its use was to reduce the overall segmentation errors to improve the segmentation accuracy. To investigate the objective, two different segmentation experiments were conducted on the same database. The first experiment was to apply OSNBV without TCM. The second one was to use OSNBV with TCM. The overall segmentation error from the first experiment was 6.94%. The second experiment showed 3.35% of overall segmentation error. Through the experiment results and analysis, the use of TCM was very effective to reduce the segmentation errors from 6.94% to 3.35%.

TABLE 5. COMPARING THE FINAL SEGMENTATION PERFORMANCE OF THE PROPOSED APPROACH (OSNBV) TO THE EXISTING SEGMENTATION METHODOLOGIES IN THE LITERATURE

Methodology	Database		Segmentation Error Rate (%)				
in the literature			Over	Under	Bad	Average	
[13]	CEDAR	200	0.00	0.50	1.80	0.76	
[24]	CEDAR	317	10.00	0.20	8.70	6.30	
OSNBV with TCM	CEDAR	311	2.76	1.96	5.32	3.35	

The final segmentation performance of the proposed approach has been compared to the existing ones from the literature and shown in Table 5. The over segmentation error rate of the proposed approach was lower than [24], but higher than [13]. The under segmentation error of the proposed approach was higher than [13,24]. The bad segmentation error of the proposed approach was lower than [24], but higher than [13]. Finally, the average error rate of the proposed approach is lower than the approach presented in [24], but higher than the approach presented in [13]. However, comparing our result to [13] is unfair because authors in [13] used only 200 words by excluding the distorted words from the same benchmark database. In addition, approach in [24] uses over-segmentation and validation strategies similar to the proposed OSNBV. Therefore, it could be the fair comparison of the result of [24] and the proposed approach. As shown in the result, the proposed segmentation strategy is promising and better compared to [24] in terms of average segmentation error.

V. CONCLUSIONS

In this paper, a novel Over-Segmentation and Neural Binary Validation (OSBNV) approach has been proposed as a segmentation strategy for off-line cursive handwriting recognition. The experiments using the proposed approach on CEDAR benchmark dataset have been conducted. The results are very promising. The over, under and bad segmentation errors were 2.76%, 1.96% and 5.32% respectively. A comparative analysis showed that the proposed approach performed well in comparison to some existing techniques in the literature. The analysis also showed that the validation function with transition count matrix has played a major role in reducing the over-segmentation and bad-segmentation errors.

REFERENCES

- H. Fujisawa, "Forty years of research in character and document recognition--an industrial perspective," *Pattern Recognition*, vol. 41, Aug. 2008, pp. 2435-2446.
- [2] B. Verma and M. Blumenstein, "Fusion of segmentation strategies for off-line cursive handwriting recognition," Pattern Recognition Technologies and Applications: Recent Advances, *Information Science Reference*, New York, USA, ISBN: 978-1-59904-807-9, 2008.
- [3] R.M. Suresh and S. Arumugam, "Fuzzy technique based recognition of handwritten characters," *Image and Vision Computing*, vol. 25, Feb. 2007, pp. 230-239.
- [4] N. Arica and F.T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 31, May. 2001, pp. 216-233.
- [5] N. Arica and F.T. Yarman-Vural, "Optical Character Recognition for Cursive Handwriting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.

24, Jun. 2002, pp. 801-813.

- [6] R.G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, Jul. 1996, pp. 690-706.
- [7] E. Vellasques, L.S. Oliveira, A.S. Britto Jr, A.L. Koerich, and R. Sabourin, "Filtering segmentation cuts for digit string recognition," *Pattern Recognition*, vol. 41, Oct. 2008, pp. 3044-3053.
- [8] B. Verma, P. Gader, and W. Chen, "Fusion of multiple handwritten word recognition techniques," *Pattern Recognition Letters*, vol. 22, Jul. 2001, pp. 991-998.
- [9] S. Zhao, Z. Chi, P. Shi, and H. Yan, "Two-stage segmentation of unconstrained handwritten Chinese characters," *Pattern Recognition*, vol. 36, Jan. 2003, pp. 145-156.
- [10] A. Elnagar and R. Alhajj, "Segmentation of connected handwritten numeral strings," *Pattern Recognition*, vol. 36, Mar. 2003, pp. 625-634.
- [11] K. Hussein, A. Agarwal, A. Gupta, and P. Wang, "A knowledge-based segmentation algorithm for enhanced recognition of handwritten courtesy amounts," *Pattern Recognition*, vol. 32, Feb. 1999, pp. 305-316.
- [12] J. Sadri, C.Y. Suen, and T.D. Bui, "A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings," *Pattern Recognition*, vol. 40, Mar. 2007, pp. 898-919.
- [13] X. Xiao and G. Leedham, "Knowledge-based English cursive script segmentation," *Pattern Recognition Letters*, vol. 21, Sep. 2000, pp. 945-954.
- [14] Q. Xu, L. Lam, and C. Suen, "Automatic Segmentation and Recognition System for Handwritten Dates on Canadian Bank Cheques," *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2003, pp. 704-708.
- [15] B. Yanikoglu and P.A. Sandon, "Segmentation of off-line cursive handwriting using linear programming," *Pattern Recognition*, vol. 31, Dec. 1998, pp. 1825-1833.
- [16] N. Tripathy and U. Pal, "Handwriting segmentation of unconstrained Oriya text," *Sadhana*, vol. 31, Dec. 2006, pp. 755-769.
- [17] S. Zhao and P. Shi, "Segmentation of Connected Handwritten Chinese Characters Based on Stroke Analysis and Background Thinning," *PRICAI 2000 Topics in Artificial Intelligence*, 2000, pp. 608-616.
- [18] R. Alhajj and A. Elnagar, "Multiagents to Separating Handwritten Connected Digits," *Systems, Man and Cybernetics, Part A: Systems* and Humans, IEEE Transactions on, vol. 35, Sep. 2005, pp. 593-602.
- [19] Z. Liang and P. Shi, "A metasynthetic approach for segmenting handwritten Chinese character strings," *Pattern Recognition Letters*, vol. 26, Jul. 2005, pp. 1498-1511.
- [20] A. Dawoud, "Iterative Cross Section Sequence Graph for Handwritten Character Segmentation," *Image Processing, IEEE Transactions on*, vol. 16, Aug. 2007, pp. 2150-2154.
- [21] C. Renaudin, Y. Ricquebourg, and J. Camillerapp, "A General Method of Segmentation-Recognition Collaboration Applied to Pairs of Touching and Overlapping Symbols," *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, 2007, pp. 659-663.
- [22] M. Suwa, "Segmentation of connected handwritten numerals by graph representation," *Document Analysis and Recognition*, 2005. *Proceedings. Eighth International Conference on*, 2005, pp. 750-754.
- [23] H. Lee and B. Verma, "A novel multiple experts and fusion based segmentation algorithm for cursive handwriting recognition," *IEEE International Joint Conference on Neural Networks*, 2008, pp. 2994-2999.
- [24] B. Verma, "A contour code feature based segmentation for handwriting recognition," *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2003, pp. 1203-1207.