2011 International Conference on Document Analysis and Recognition

A Handwritten Character Extraction Algorithm for Multi-language Document Image

Yonghong Song, Guilin Xiao, Yuanlin Zhang, Lei Yang, Liuliu Zhao
Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University
Xi'an, China
songyh@mail.xjtu.edu.cn

***Abstract*—In this paper, we propose a novel method for extracting handwritten characters from multi-language document images, which may contain various types of characters, e.g. Chinese, English, Japanese or their mixture. Firstly, text patches in document image are segmented based on connected component analysis. Rules for merging connected components are chosen according to the results of language identification. Then features are extracted for each basic analysis unit-text patch. Genetic algorithm is applied for feature fusion and patch type classification. Finally, a Markov Random Field model is utilized as a post-processing step to further correct the misclassification of text patch type by considering the document context. Experimental results show that the proposed algorithm can apparently improve the performance of handwritten character extraction.**

Keywords-handwritten character extraction; multi-language; document segmentation; feature fusion; Markov random field

I. INTRODUCTION

Handwritten character extraction is a necessary and essential technique for many applications, such as handwritten OCR[1,2,3], document image retrieval[4], automatic postal processing[5], historic document processing[6], medical prescription understanding[7] and so on. Most of the previous research on this issue focused on document images with a single language. Thus when dealing with documents of different languages, the existing algorithms need manual adjustment. How to effectively extract handwritten characters from document images, which may contain multiple types of contents and languages, is the main problem remained for the handwritten character extraction problem currently. In this paper, we propose a novel approach to solve this problem.

Handwritten character extraction can be performed at three levels: the text line[8], word[9,10,11,12] and character[13,14,15]. Methods designed at text line level are easy to implement and can be applied for different languages, but they are usually utilized for analyzing documents with some specific formats. At character level, the performance would be poor when characters are curved and overlapping. Considering the limits of methods at text line and character level, this paper extracts handwritten character at the word level.

Here, first of all, a text patch segmentation method is proposed. Text patches in document image are segmented based on an improved connected component analysis algorithm. Rules for merging connected components are chosen according to the results of language identification. Then features are extracted for each basic analysis unit-text patch. Genetic algorithm is applied for feature fusion and patch type classification. Finally, a Markov Random Field model is utilized as a post-processing step to further correct the misclassification of text patch type by considering the document context.

The rest of the paper is organized as follows: Section II is system overview. Section III presents the detail of proposed methods. Section IV shows the experimental results. Section V is the conclusion.

II. SYSTEM OVERVIEW

Fig. 1 illustrates the overall system framework. Firstly, non-text contents are excluded using connected component features. Then text patches are segmented from the document image and further analyzed. Several types of features are extracted for each text patch, and are then weighted respectively by a generic algorithm. After that, with an Adaboost classifier, weighted features are classified to either handwritten class or machine printed class. Finally, in order to further improve the classification precision, a MRF (Markov Random Field) model is implemented by introducing the context of isolated text patches.

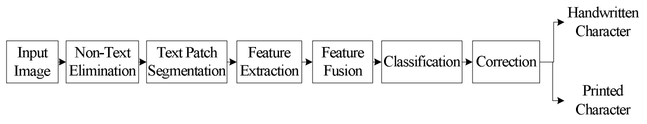

Figure 1. Flowchart of handwritten character extraction

III. METHOD DESCRIPTION

A. *Text Patch Segmentation for Multi-Language document*

Previous text patch segmentation methods utilized simple operations, such as mathematical morphological close operation [12] and simple neighborhood rules [16]. However, when facing noises, overlapping between handwritten and printed contents, and different spatial proximity of different languages, these methods are often ineffective.

To solve the above problems, a novel text patch segmentation method is proposed in Fig. 2. First, the color of input image is inversed, followed by a mathematical morphological close operation with a 3*3 operator to connect closed strokes. Then by applying connected component labeling, small connected components are eliminated, and average character size is estimated. After that, connected components are grouped with each other by to choosing appropriate merging rules based on language identification results, so a number of connected components pairs art to be merged. Finally, the equivalence class algorithm is applied to merge connected component pairs quickly, and text patches are segmented.

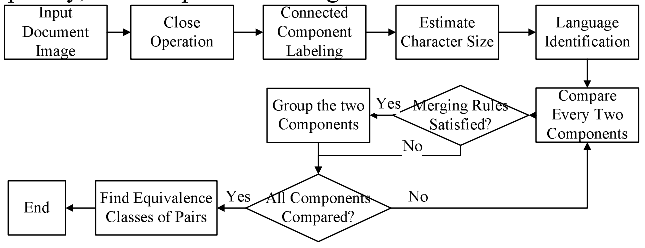

Figure 2. Text patch segmentation for mutiple languages

1) Language Identification of Connected Components

Since the spatial proximity of texts of different languages are quite distinct, different merging rules should be applied for connected components of different languages. So before merging connected components, it is necessary to identify the language type of connected components.

This paper focuses on three languages: Chinese, English and Japanese, which can be categorized into two classes: strokes-composed or letter-composed. We wish to get character patches for strokes-composed texts, and word patches for letter-composed texts.

For each connected component, features including aspect ratio, pixel density, longest run-length, and cross-count feature are extracted, classified by FLD (Fisher Linear Discriminant). Longest run-length feature is the longest horizontal or vertical run-length divided by width or height. Cross-count feature is the horizontal counts of changes of pixels from white to black divided by width.

After classification, a voting procedure is used to improve the accuracy of language identification. First, texts are segmented into lines. Each line contains several connected components and each connected component is classified to one type of language. Then by voting the language types of connected components inside a line, language type with the largest vote is considered as the language type of all components inside that text line.

2) Merging Rules of Connected Components

The distances between two connected components *i* and *j* are shown in Fig. 3.

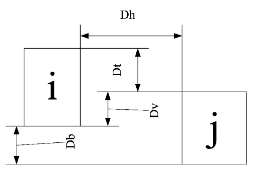

Figure 3. Distances between Two Connected Components

Let *L*, *R*, *T* and *B* represent the left, right, top and bottom coordinates of connected components respectively. The horizontal, vertical, top and bottom distance is defined in Eq.(1).

$$\begin{aligned} D_h(i,j) &= \max(L_i, L_j) - \min(R_i, R_j) \\ D_v(i,j) &= \max(T_i, T_j) - \min(B_i, B_j) \\ D_t(i,j) &= |T_i - T_j| \\ D_b(i,j) &= |B_i - B_j| \end{aligned} , \quad (1)$$

Where L , R , T and B are the left, right, top and bottom coordinates of connected components.

Where $\max(\cdot)$, $\min(\cdot)$ and $|\cdot|$ are maximize, minimize and absolute value function respectively.

The merging rules of two components are defined in Eq.(2).

$$\begin{aligned} & D_k(i,j) < C_k, k \in \{h \quad v \quad t \quad b\} \\ & \max(h_i, h_j) < 2 * \min(h_i, h_j) \end{aligned} . \quad (2)$$

Here h_i and h_j are the heights of i and j . The distance constraints C_k in Eq.(2) depend on the language type of connected components. For stoke-composed languages such as Chinese and Japanese, the distances between strokes and characters are very close, so strict constraints are imposed: $C_h = W_s / 5, C_v = H_s / 8, C_t = C_b = H_s / 2$. *F*or letter-composed languages like English, the distance between letters and words are larger, so less strict constraints are imposed: $C_h = W_l / 3, C_v = 0, C_t = C_b = H_l / 2$, here, W_s, H_s W_l and H_l are the estimated character or letter widths and heights of strokes-composed and letter-composed languages.

The estimated width and height are obtained by searching the peaks of width histogram and height histogram of connected components. In order to avoid the influences of noises, components which are too large or too small are excluded for the calculation of histograms. For each document image, the width and height of stroke-composed and letter-composed character size are estimated separately.

By using language identification, connected components are grouped by using different merge rules according to their language types. A large number of connected component pairs are obtained.

3) Merging Pairs of Connected Components

Connected component pairs are needed to be further merged into text patches. The problem can be modeled as finding the equivalence class.

Given a set $U = \{1, \ 2, \ ..., \ n\}$ and an equivalence relation (satisfies the reflexivity, symmetry and transitivity) $R = \{(i_1, j_1), \ (i_2, j_2), \ ..., \ (i_r, j_r)\}$ on U, the equivalence class of an element a in U is the subset of all elements in U which are equivalent to a: $[a]_{eq} = \{x \in U \mid (a, x) \in R\}$.

In this paper, the set U represents all connected components; the equivalence relation R represents connected components pairs. An equivalence class equals a text patch. Finding all equivalence classes equals obtaining all text patches. By using advanced data structure, the problem can be solved in linear complexity [17].

4) Experimental Results of Text Patch Segmentation

Fig. 4 shows the results of a classical method[12]. In Fig.4 (a), when a long handwritten stroke overlaps with a large area of printed texts, the segmented text patch will contain a lot of printed texts which are not connected with the stroke. Fig.4 (b) shows the result of document image with multiple languages. English texts are segmented as a word; however, several Chinese texts are segmented as a line. Also, some noises, handwritten texts and printed texts are segmented in the same patch.

Fig.5 is the results of proposed segmentation. English texts are segmented as "a word", and Chinese texts are no longer extracted as a line, but as a character. When a long handwritten stroke overlaps with a large area of printed texts, the segmented text patch contains only printed texts that are connected with the stroke. Noises, handwritten texts and printed texts are extracted separately when there are close to each other.

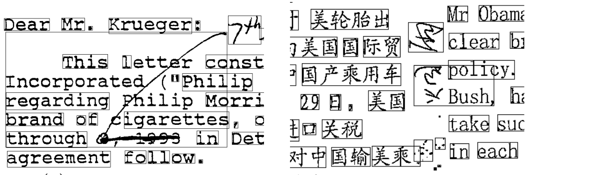
(a) Overlap (b) Multiple languages, noises

Figure 4. Text Patch Segmentation of Previous Method

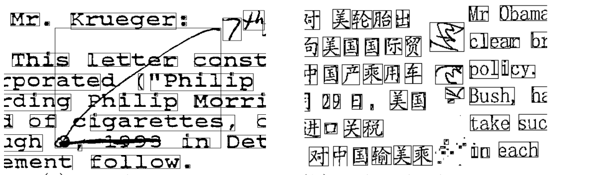
(a) Overlap (b) Multiple languages, noises

Figure 5. Text Patch Segmentation of Proposed Method

B. Feature Extraction and Fusion

For each text patch, four kinds of features are extracted including structure feature, run-length feature, cross-count feature and bi-level co-occurrence feature. These features are explained in [10].

By concatenating these features together, a 50 dimensional feature vector is obtained. However, concatenating features directly could cause some problems such as feature redundancy or incompatibility. So these features need to be fused appropriately. In this paper, feature fusion is solved by assigning a weight to each dimension of the feature vector.

1) Problem Discription for Feature Fusion

Suppose $D=\{(x^1,y_1),\ \ldots,\ (x^n,y_n)\}$ is a set of samples, n is the sample number, $x^j=\{x_1{}^j,x_2{}^j,\ldots,x_d{}^j\}^T$, denotes a feature vector width as its dimension, y_j represents the label of vector j, $w=\{w_1,w_2,\ldots,w_d\}^T$ is the weight vector for each dimension of feature vector. Then $x_i'^j=w_i x_i^j$ is a weighted feature, the samples are weighted as $D'=\{(x'^1,y_1),\ \ldots,\ (x'^n,y_n)\}$. The feature weighting problem is model as optimization problem: finding the optimal weight w^* that minimizes cost function, as in Eq.(3).

$$w^*=\arg\min_w \operatorname{cost} f(D'). \tag{3}$$

This optimization problem could be solved using gradient descent or Gauss-Newton method which tends to local optimum. So we implement the genetic algorithm which is a stochastic method and tends to global optimum.

2) Genetic Algorithm-Based Feature Weighting

The genetic algorithm in [18, 19, 20] is inspired by biology. It simulates the biological evolution process: copy, mutate and crossover. In one iteration, the best solutions are copied, and then mutate and crossover operation are applied to the copied solutions to generate more potential solutions. After several iterations, the solutions tend to be optimum. Here, we use 50 iterations, and 100 potential solutions for each iteration.

The cost function is calculated as follows: 80% of samples are randomly chosen as training samples to train Adaboost classifier, the rest 20% are chosen as testing samples to calculate the classification accuracy.

C. Correction of Mis-Classification by MRF

Due to the limits of features and classifier model, wrong classifications are inevitable. However, they can be corrected using context information of document images.

Markov random field can incorporate contextual information or constraints in a quantitative way, and there are usually several kinds of energy cliques. Some cliques denote positive information and some are negative constraints. Besides the basic features of handwritten and printed characters, there are usually some useful statistic contextual or constrained information around handwritten characters. So MRF is a good way to quantitatively model it. The energy function with cliques of MRF for classification of handwritten characters and printed contents are defined as follows.

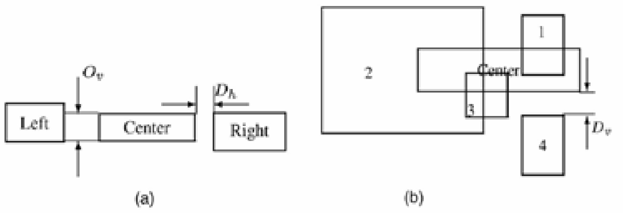

Figure 6. Clique[10]. (a) C_p. (b) C_n.

Let $F=\{F_1,F_2,\ldots,F_m\}$ denote the random field defined on site $S=\{1,2,\ldots,m\}$. Let $X=\{X_1,X_2,\ldots,X_m\}$ denote the observations or features on the site S. The objective is to find a configuration $f^* = \{f_1, f_2, ..., f_m\}$ that maximizes the posteriori probability, as in (4).

$$f^* = \arg\max_f P(f \mid X) \propto \arg\max_f P(X \mid f) P(f) \cdot \quad (4)$$

The equivalence between the MRF and Gibbs distribution can change the maximum a posterior problem to energy minimization problem, as in Eq.(5).

$$f^* = \arg\min_f U(f \mid X) = \arg\min_f (U(X \mid f) + U(f)) \cdot \quad (5)$$

where $U(f \mid X)$ is the posterior energy, $U(X \mid f)$ the likelihood energy, $U(f)$ is the a priori energy, which is defined by summarizing clique potential on all cliques.

In this paper, the likelihood energy is defined by changing the classification result from discrete class labels to continuous probability, as in Eq.(6).

$$U(X \mid f) = \sum_{i \in S} (-P(f_i \mid X_i)) \cdot \quad (6)$$

where $P(f_i = m \mid X_i) = 1/(1 + \exp(-k * g(X_i)))$ is the probability of classifying text patch X_i as machine printed, $P(f_i = h \mid X_i) = 1 - 1/(1 + \exp(-k * g(X_i)))$ is the probability of classifying X_i as handwritten, $g(x)$ is the weighted output of weak classifiers in Adaboost classifier, k is a parameter that controls the gradient of the curve. The a priori energy is the same as in paper [10]. Two kinds of cliques are defined: C_p and C_n (shown in Fig.6).

Clique potentials $V_p(c)$ and $V_n(c)$ are defined on each clique, as in Eq.(7) and Eq.(8). f_l and f_r are the labels of left and right patches of the center patch, and f_1, f_2, f_3 and f_4 are labels of the surrounding patches of the center patch, as shown in Fig.6.

$$V_p(c) = V_p(f_l, f_c, f_r) = -\frac{P(f_l, f_c, f_r)}{(P(f_l)P(f_c)P(f_r))^{\omega}} \quad (7)$$

$$V_n(c) = V_n(f_c, f_1, f_2, f_3, f_4) = -\frac{P(f_c, f_1, f_2, f_3, f_4)}{(P(f_c)P(f_1)P(f_2)P(f_3)P(f_4))^{\omega}} \quad (8)$$

Then the priori energy obtained as in Eq.(9).

$$\cdot\ U(f) = \omega_p \sum_{c \in C_p} V_p(c) + \omega_n \sum_{c \in C_n} V_n(c) \cdot \quad (9)$$

Finally the posterior energy is obtained, as in Eq.(8).

$$U(f \mid X) = \omega_s \sum_{i \in S} (-P(f_i \mid X_i)) + \omega_p \sum_{c \in C_p} V_p(c) + \omega_n \sum_{c \in C_n} V_n(c) \cdot \quad (10)$$

HCF [21] is applied to get the optimal configuration f^*. Four parameters ω_s, ω_p, ω_n and k are determined by using the genetic algorithm mentioned in Part III.

IV. Experiment Result

100 images of multiple languages are constructed, including Chinese, English, Japanese and mixed languages, with 25 images of each language. The handwritten characters are written by ten different writers according to their wishes and using different writing tools. The images are scanned in A4 documents in 200 DPI. Each image contains more than 10 handwritten phrases.

TABLE I. Method of Each Test

Test	MLS	FW	MRF
T1	N	N	N
T2	Y	N	N
T3	Y	Y	N
T4	Y	Y	Y

Precision and recall are used to evaluate the algorithms, as shown in Eq.(11) and Eq.(12). In this way only handwritten characters which are totally extracted are computed into precision and recall, those ones partially extracted will not be included in the correct extraction results.

$$precision(i) = \frac{\text{\# of patches correctly classified as class i}}{\text{\# of patches classified as class i}} \quad (11)$$

$$recall(i) = \frac{\text{\# of patches correctly classified as class i}}{\text{\# of patches belonging to class i}} \quad (12)$$

Four tests are designed, as is shown in table I. "Y" means a method is used, "N" means it is not used. MLS means Multi-Language Segmentation, FW means Feature Weighting, MRF means Markov Random Field.

The results are shown in Fig. 7. Precision and recall are increased from below 85% and 88% in T1 to above 98% in T4. Compared with T1, T2 applied text patch segmentation method proposed in this paper, so the precision is greatly increased. Compared with T2, T3 used feature weighting, so the precision and recall are both increased. Compared with T3, T4 implemented MRF, so the precision increased greatly. By applying new text patch segmentation method, feature weighting and MRF, the performance is greatly improved.

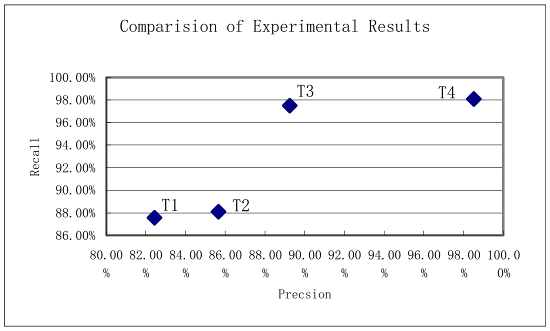

Figure 7. Comparision of Performance

Table II is the results of each language under T4, P and R represent precision and recall. The precision and recall for English document is the best, followed by Chinese document. The two are above the average. The performance of mixed language is worse than the average. The performance of Japanese language is the worst, because many Japanese characters are originated from cursive

Chinese characters, which is similar to handwritten character. So a further study of Japanese character is needed.

Table III is the time performance. The system is implemented in VC++ under VS2005. The total processing time is less than 700 milliseconds. The text segmentation step consumes most time, because a complicated text patch segmentation method is applied.

TABLE II. PERFORMANCE OF EACH LANGUAGE

	# of Samples	# of Classified Samples	# of Correctly Classified Samples	P	R
Chinese	524	517	514	99.42%	98.09%
English	534	533	530	99.44%	99.25%
Japanese	454	459	443	96.51%	97.58%
Multiple	379	374	368	98.40%	97.10%

TABLE III. TIME PERFORMANCE (IN MS)

Text Segmentation	Feature Extraction	Classification	MRF	Total
307	110	149	124	689

Environment: Windows XP, Intel Core 2 Duo @2.20GHz 2.19GHz, 1.00 GB MEM. Test Image: A4 document scanned in 200 DPI, average size: 1728*2156 pixels

Several examples are shown in Fig. 8. In Fig.8 (b1) and (b2), some long printed text lines are extracted as handwritten character. After applying Multi-Language Segmentation, no text line is extracted, as is shown in Fig.8 (c1) and (c2). Fig.8 (b3) and (c3) shows the improvement of feature weighting. In Fig.8 (b4) and (c4), after using MRF, most wrong classifications are corrected.

However, when the contextual information has more influence than the classification probability, it would be possible that some handwritten patches be relabeled as printed character patch by MRF, as is shown in Fig.9.

As the handwritten characters are randomly written by different writers, the test set we made is usually stochastic. We collected 100 images from Legacy Tobacco Document Library[22] and IAM-database[23], in these images the characters are mainly English. We use 20 of these 100 images to train the classifier and left images to test. The precision and recall are both more than 99%, so our system can easily can easily adapt to changing document collection.

V. Conclusion

This paper studied methods of handwritten character extraction in document images with multiple languages. First, an adaptive text segmentation method is proposed by using language identification to choose merging rules, and the equivalence class algorithm is applied to further merge pairs of connected components. Then Feature fusion problem is modeled as optimization problem, and a genetic algorithm is implemented. Finally, Markov Random Field model is improved to correct wrong classification.

Further study will involve more languages and more test images. Also more research on document features, feature fusion, and context modeling of document image are necessary.

REFERENCES

[1] Fujisawa H. Forty years of research in character and document recognition-an industrial perspective[J]. Pattern Recognition, 2008, 41 (8): 2435-2446.

[2] Nagy G. Twenty Years of Document Image Analysis in PAMI[J]. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2000, 22 (1): 38 - 62.

[3] Sellen AJ, Harper RH. The Myth of the Paperless Office[M]: The MIT Press, Cambridge, MA, 2001.

[4] Zhu G, Zheng Y, Doermann D, et al. Signature Detection and Matching for Document Image Retrieval[J]. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2009.

[5] Fujisawa H. The Past and the Future of Character and Document Recognition - An Industrial View[C]. ICDAR 2007, 2007.

[6] Cherieta M, Bunkea H, Hua J, et al. New Frontiers in Handwriting Recognition[J]. Pattern Recognition, 2009, 42 (12): 3129-3130.

[7] Milewski R, Govindaraju V. Extraction of Handwritten Text from Carbon Copy Medical Form Images[J]. Lecture Notes in Computer Science, 2006, 3872: 106-116.

[8] Srihari SN, Shim YC, Ramanprasad V. A System to Read Names and Address on Tax Forms. CEDAR, SUNY, Buffalo, N.Y., 1994.

[9] Guo JK, Ma MY. Separating Handwritten Material from Machine Printed Text Using Hidden Markov Models[C]. Proc. Int'l Conf. Document Analysis and Recognition, 2001: 439-443.

[10] Zheng Y, Li H, Doermann D. Machine Printed Text and Handwriting Identification in Noisy Document Images[J]. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2004, 26.

[11] Shetty S, Srinivasan H, Beal M, et al. Segmentation and labeling of documents using Conditional Random Fields[C]. Proceedings of SPIE, 2007.

[12] Peng X, Setlur S, Govindaraju V, et al. Markov Random Field Based Text Identification from Annotated Machine Printed Documents[C]. 10th International Conference on Document Analysis and Recognition, 2009: 431-435.

[13] Kuhnke K, Simoncini L, Kovacs-V ZM. A System for Machine-Written and Hand-Written Character Distinction[C]. Proc. Int’l Conf. Document Analysis and Recognition, 1995: 811-814.

[14] Zheng Y, Liu C, Ding X. Single Character Type Identification[C]. Proc. SPIE Conf. Document Recognition and Retrieval, 2002: 49-56.

[15] Koyama J, Kato M, Hirose A. Handwritten Character Distinction Method Inspired by Human Vision Mechanism[C]. Lecture Notes In Computer Science, 2008.

[16] Zheng Y, Li H, Doermann D. The Segmentation and Identification of Handwriting in Noisy Document Images[C]. Proc. Int'l Workshop Document Analysis Systems, 2002: 95-105.

[17] Sahni S. Data Structures, Algorithms, And Applications In C++ [M]: Silicon Press, 2004.

[18] Duda RO. Pattern Classification,2nd Edition[M]: Wiley-Interscience, 2000.

[19] Holland JH. Adaptation in Natural and Artificial Systems[M]: The MIT Press, 1992.

[20] Segaran T. Programming Collective Intelligence[M]: O'Reilly Media, 2007.

[21] Li SZ. Markov Random Field Modeling in Image Analysis, second ed.[M]. New York: Springer-Verlag, 2001.

[22] The Legacy Tobacco Document Library (LTDL). University of California, San Francisco, 2007.

[23] Marti U, Bunke H. The IAM-database: an English Sentence Database for Off-line Handwriting Recognition[J]. Int Journal on Document Analysis and Recognition, 2002, 5: 39 - 46.

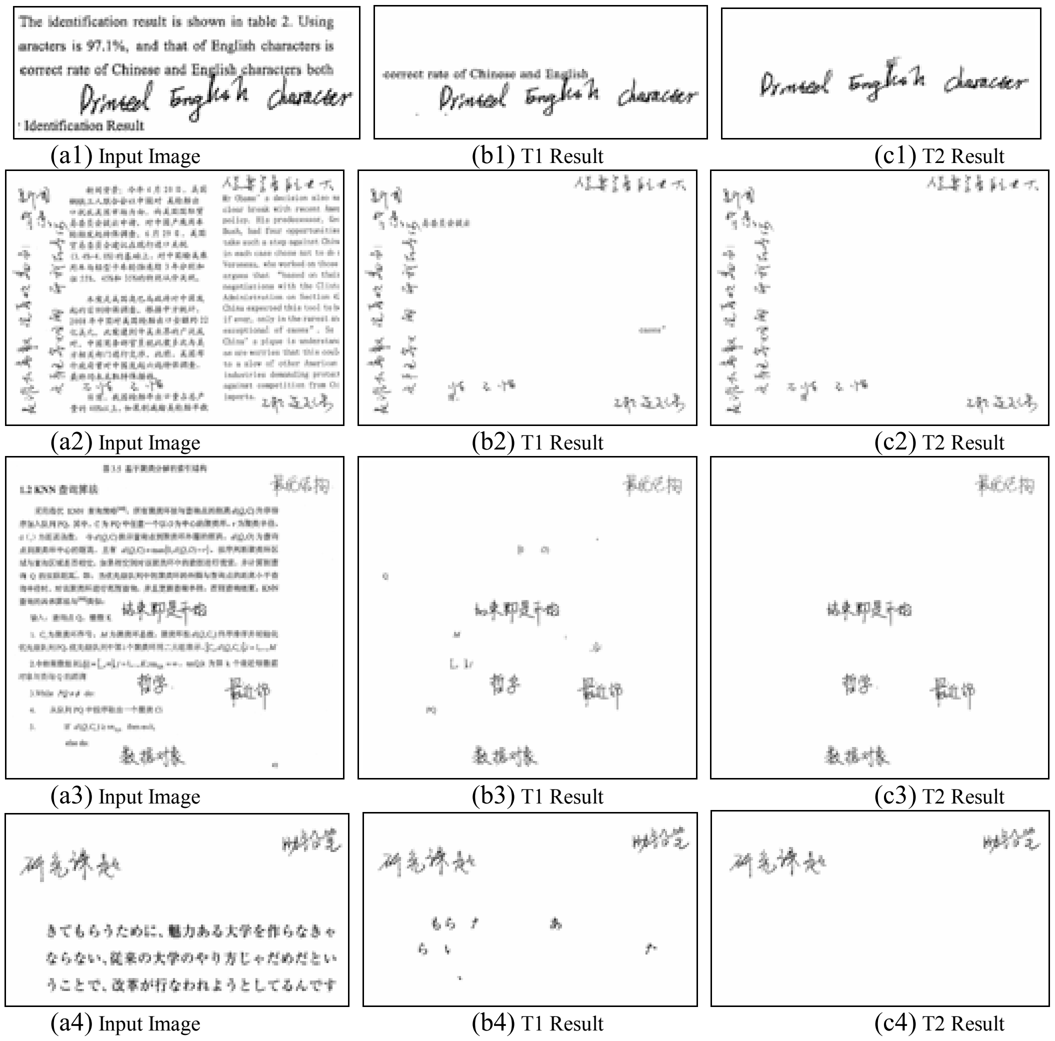

Figure 8. Examples of experimental results

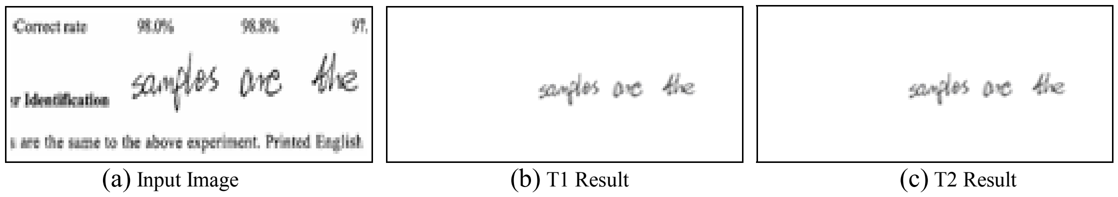

Figure 9. An example of mis-labeled results