IJDAR
DOI 10.1007/s10032-015-0257-8

ORIGINAL PAPER

A Nom historical document recognition system for digital archiving

Truyen Van Phan[1] · **Kha Cong Nguyen**[1] · **Masaki Nakagawa**[1]

Received: 12 December 2014 / Revised: 31 October 2015 / Accepted: 11 November 2015

Abstract A Nom historical document recognition system is being developed for digital archiving that uses image binarization, character segmentation, and character recognition. It incorporates two versions of off-line character recognition: one for automatic recognition of scanned and segmented character patterns (7660 categories) and the other for user handwritten input (32,695 categories). This separation is used since including less frequently appearing categories in automatic recognition increases the misrecognition rate without reliable statistics on the Nom language. Moreover, a user must be able to check the results and identify the correct categories from an extended set of categories, and a user can input characters by hand. Both versions use the same recognition method, but they are trained using different sets of training patterns. Recursive X–Y cut and Voronoi diagrams are used for segmentation; k–d tree and generalized learning vector quantization are used for coarse classification; and the modified quadratic discriminant function is used for fine classification. The system provides an interface through which a user can check the results, change binarization methods, rectify segmentation, and input correct character categories by hand. Evaluation done using a limited number of Nom historical documents after providing ground truths for them showed that the two stages of recognition along with user checking and correction improved the recognition results significantly.

Keywords Nom script · Historical documents · Text digitization · Off-line character recognition · Binarization · Character segmentation · Recursive X–Y cut · Area Voronoi diagram · Document image analysis

1 Introduction

Three main scripts have been used in Vietnam historically: *Chữ Hán* (漢字, "Han script"), *Chữ Nôm* (𡨸喃, "Nom script"), and the currently used script-*Chữ Quốc Ngữ*(國語, "National language"). *Han script* is the native Vietnamese name for a form of classical Chinese used in pre-modern Vietnam from 111 BC until the early twentieth century. *Nom script* is the previous transcription system for vernacular Vietnamese language text. It represents Vietnamese sounds by using original Chinese as well as new characters created in a way similar to the way that Chinese characters are formed from radicals. It was used from 939 AD until the beginning of the twentieth century in parallel with *Han script*. *National language*, the modern writing system, is based on the Latin-based alphabet with some digraphs and nine accent marks. It was created in the seventeenth century, but it has been widely used only since the 1920s.

From the tenth century into the twentieth century, much of Vietnamese literature, philosophy, history, law, medicine, religion, and government policy were written in *Nom script*. Specially, it was widely used from the fifteenth to nineteenth centuries by Vietnam's cultured elite. This heritage is now nearly lost, however, due to displacement by the modern script—*National language*. According to the Vietnamese Nom Preservation Foundation (VNPF), fewer than 100 schol-

✉ Masaki Nakagawa
nakagawa@cc.tuat.ac.jp

Truyen Van Phan
truyenvanphan@gmail.com

Kha Cong Nguyen
congkhanguyen@gmail.com

[1] Department of Information and Communication Engineering, Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan

Published online: 12 December 2015

(a) (b)

Fig. 1 Example images of Nom documents

ars worldwide can read *Nom script* today. Furthermore, over 90 % of the collected documents have not yet been translated into modern Vietnamese, and very few of them have been text-digitized. As a consequence, much of Vietnam's history is, in effect, inaccessible to the 90 million speakers of the language. For this reason, it is urgent to preserve this cultural heritage.

Due to the enormous value of historical documents for studying history, particularly the social aspects and life styles in previous time periods, many countries are working to preserve their historical documents. Since our problem is to read old Vietnamese of Chinese origin, we focus here on projects related to Chinese or to languages of Chinese origin.

Kim et al. [1] developed a system for digitizing more than 10 million handwritten Hanja historical documents. Hanja was used until the late nineteenth and early twentieth century before Hangul came into widespread use. It is the Korean name for Chinese characters incorporated into the Korean language with Korean pronunciation. The system uses manual typing and handwriting recognition based on the Mahalanobis distance. They set 2568 classes for character recognition out of the 5599 classes that they found in vol. 29 of the Seungjungwon Diary.

In China, a huge project was performed over 18 years by Digital Heritage Publishing Ltd. to digitize more than 36,000 volumes (4.7 million pages) of Siku Quanshu (四庫全書). Siku Quanshu is the largest collection of books on Chinese history and was compiled by 361 scholars during the Qianlong period (1711–1799). They first applied optical character recognition (OCR) to segment and recognize characters and then manually corrected misrecognized characters. Since the project was done by a private company, more detailed information is unavailable.

For Nom text digitization, Vietnamese agencies and worldwide foundations have been collecting hundreds of thousands of *Hán Nôm* documents [2]. The term *Hán Nôm* (漢喃, "Han-Nom") in Vietnamese refers to text that is written in a mixture of *Han script* and *Nom script* or *Han script* with parallel translation to *Nom script*. Among the various preservation projects, the most notable is the Digitization Project of the *Han-Nom* Special Collection, a collaborative project between the National Library of Vietnam and the VNPF. Over 5200 historical documents have been scanned, and approximately 1907 documents containing 133,495 pages are available at http://hannom.nlv.gov.vn. Figure 1 shows examples of historical document pages written in *Han-Nom* (Hereafter in this paper, the term "Nom documents" refers to *Han-Nom* documents). Although the project has scanned a number of documents, there was no recognition, so helpful functions such as search and annotation are not supported.

Nom text digitization has a handicap compared to Korean and Chinese historical text digitization: Nom is an almost lost language. The number of sample patterns tagged with the ground truth is quite limited. In addition, although the regular Nom character set is defined, there are no reliable statistics on the number of categories for generally used Nom characters. This means that more than 30,000 character categories have to be considered. There have been no reports on using OCR for such a large number of categories.

With the construction of digital libraries, valuable Vietnamese documents are being preserved and made available to the public. This is helpful for scholarly research, teaching, and learning Nom in Vietnam and abroad. In order for this important heritage for learning about the past to be utilized, however, the documents must be fully digitized, i.e., indexed, annotated, recognized, and hopefully translated into modern Vietnamese. The traditional method for doing this has been reading and typing by experts. While this approach is accurate, it is labor intensive and too slow given the large number of documents and the decreasing number of experts. Thus, this process cannot be done in a short period of time. Document recognition techniques can be used to speed up the process [1].

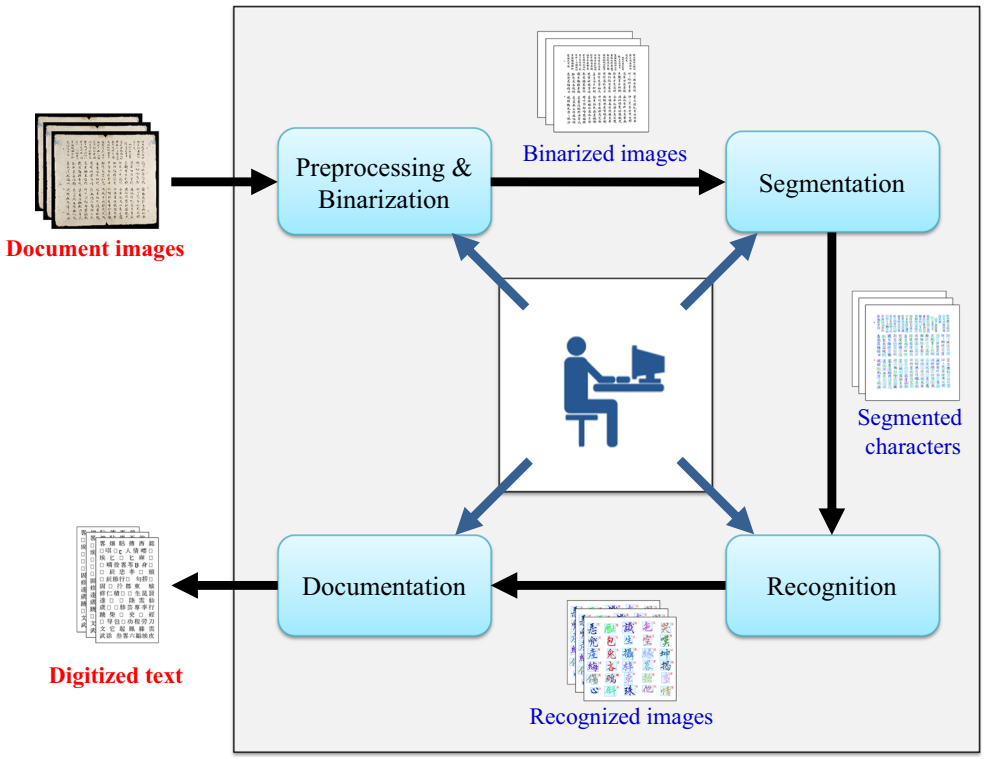

Fig. 2 Digitization system for Nom documents

OCR is often used for document recognition and digitization. The latest techniques show high performance on modern printed materials and handwritten materials using current writing styles. For historical documents, however, they are still insufficient. The difficulties come from problems of layout analysis for degraded and damaged documents, various and complex layouts, blurred text, diverse writing styles, and so on. Especially with Nom, there are other challenges related to character recognition: (1) there is no OCR engine for Nom, (2) there is no set of Nom character patterns with ground truths for training a recognizer, (3) Nom has a large character set with several thousand categories (more than 30,000), and (4) text recognition using a language model is difficult since there are only a few well-known classics that have been transcribed.

In previous work, we started solving these problems. We worked on document preprocessing, binarization, and character segmentation [3], and we tried to cluster and provide Nom character patterns extracted from historical Nom documents with ground truths by using a recognizer trained on Japanese off-line handwritten character patterns since Japanese Kanji of Chinese origin shares with Nom a considerably large set of characters [4].

We have now designed an off-line character recognition method for Nom and have created a system for building a digital text library of Nom historical documents based on the results of our previous work. Our objective is to construct a system that enables people who are not proficient in Nom to contribute to building the library.

The proposed character-recognition-based Nom historical documents digitization system consists of four main steps: (1) preprocessing and binarization, (2) segmentation, (3) recognition, and (4) documentation, as shown in Fig. 2.

The preprocessing and binarization step converts color images into binarized images. A single page image or hundreds of page images of a document can be processed at a time. Next, the character segmentation step splits the binarized images into individual character patterns. Then, the recognition step identifies class labels for character patterns automatically by using a version of our OCR engine. In these steps, the processing results can be checked and corrected through a graphical user interface. The class labels of character patterns can be also fixed in the recognition step with another version of our OCR that can recognize an extended set of character categories. Finally, the documentation step completes the document recognition process by adding the character codes and layout information.

The remainder of this paper is organized as follows: Nom character segmentation and recognition are described in Sect. 2; the graphical user interface for checking and correcting Nom documents is described in Sect. 3; the character pattern sets used to evaluate the system are shown in Sect. 4;

the experiments performed to evaluate the system are presented in Sect. 5; and the key points are summarized and future work is mentioned in Sect. 6.

2 Automatic Nom page digitization by OCR

Automatic Nom page digitization is done using two processes: segmentation of Nom pages into separate characters and recognition of the characters by OCR.

2.1 Page layout analysis

We describe the preprocessing, binarization, and segmentation steps briefly as they were previously reported [3,4]. These processing steps are aimed at obtaining clean and clear binarized images of document pages that can be segmented into individual character images.

The proposed page layout analysis for Nom documents consists of three main steps: noise removal, binarization, and segmentation, as shown in Fig. 3.

In most of the Nom documents digitized in the Digitization Project of the *Han-Nom* Special Collection, there is a black region on the border of each page image from when the documents were scanned. There are often comments in red ink on them as well, as shown in Fig. 1b. Moreover, in some documents, there is a boundary rectangle and vertical ruled lines due to the woodblock layout, as shown in Fig. 1a. Color filtering in the HSL color space is used to detect the red pixels forming the comments. They are turned gray so that their color is similar to that of the document paper background in the grayscale image and are thus removed in the subsequent binarization step. The black image border is removed from the grayscale image by histogram analysis.

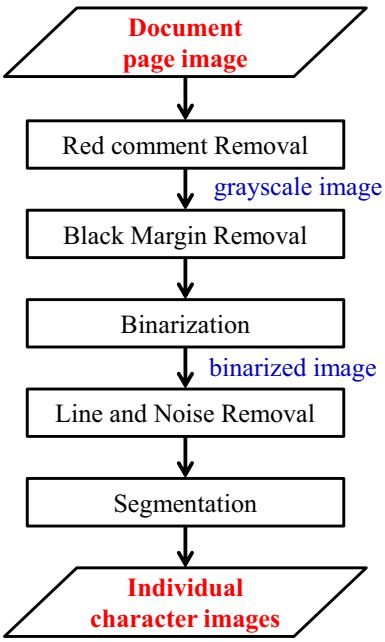

Fig. 3 Page layout analysis for Nom documents

Next, in the binarization step, a global or local thresholding method is used to convert the grayscale image into a binarized image. Suitable methods for this include the local thresholding method proposed by Su et al. [5] and 16 global thresholding methods: Otsu's method [6], a method based on simple image statistics [7], and others that are supported in Fiji—an image processing package [8]. The method used depends on the document type. For images with uniform contrast distribution of background and foreground, global thresholding is more appropriate. For degraded document images, with considerable background noise or variation in contrast and illumination, binarization with local thresholding is suitable. We use Su's method in our system as the default, and the user can select another method if it is not satisfactory. The boundary line and ruled lines are removed by histogram analysis. Finally, segmentation is applied to the noise-removed and binarized image to get segmented character images. Three methods can be used for segmentation: a recursive X–Y cut method, a Voronoi-based method, and a method combining them. The recursive X–Y cut method is a tree-based top-down method. It recursively splits the document into rectangular blocks representing columns, paragraphs, etc. that form intermediate nodes of the tree. The leaf nodes represent segmented characters. The Voronoi-based method is a bottom-up method. A Voronoi diagram is generated using sample points obtained from the contours of the connected components. Unnecessary Voronoi edges are deleted using a criterion for identifying segmented characters. The method is described in detail elsewhere [3]. Example results of the page layout analysis are shown in Fig. 4.

2.2 Two-stage architecture

We must assume that there are more than 30,000 categories for reading Nom documents. This number is 3 or 4 times larger than the numbers that are handled by current OCRs in Japan and China. Although pursuing a single recognizer is challenging, the recognition rate is degraded due to the less frequently appearing categories, even for state-of-the-art methods.

Therefore, we apply two versions of off-line character recognition: one for automatic recognition of scanned and segmented character patterns (a regular set of categories) and the other for user handwritten input (an extended set of categories). This separation is used since including less frequently appearing categories in automatic recognition increases the misrecognition rate without reliable statistics on the Nom language. Moreover, a user must be able to check the results and identify the correct categories from an extended set of categories, and a user can input characters by hand.

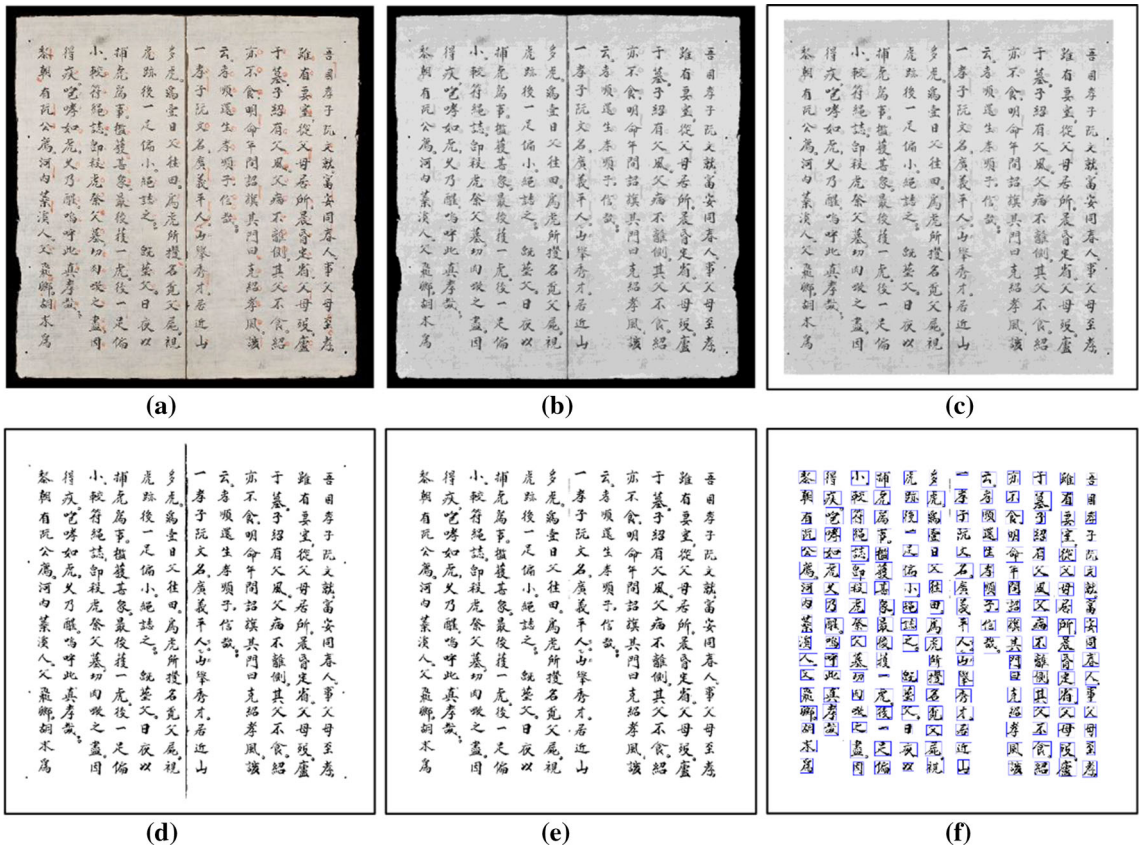

Fig. 4 Example results of page layout analysis: **a** original image, **b** red-comments-blurred grayscale image, **c** black-margin-removed image, **d** binarized image, **e** lines-and-noises-removed image, **f** character-segmented image

While both versions can use the same recognition method, they are trained using different sets of training patterns.

2.3 Character recognition method

We use one of the most accurate methods for recognizing Chinese and Japanese characters. It consists of four steps: nonlinear normalization, feature extraction, coarse classification and fine classification.

An input handwritten character pattern is first normalized using line density projection interpolation [9]. After normalization, normalization-cooperated gradient feature extraction [10] is used to extract features from the character pattern. The gradient feature vector is computed from the original character pattern without generating the normalized character image explicitly. The gradient feature vector is then decomposed into two components in two neighboring chain codes of eight direction planes, as shown in Fig. 5 [11]. The sizes of the normalized plane and the direction plane are set to 64×64 pixels and 8×8 pixels, respectively. After directional decomposition, each direction plane is blurred using a low-pass Gaussian filter. On each of the 8 direction planes, 8×8 feature values are extracted; as a result, a total of 512 features

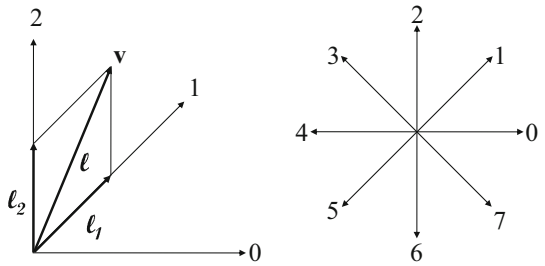

Fig. 5 Directional decomposition of gradient vector (**a**) and 8 chain code directions (**b**)

are obtained. To reduce classifier complexity and improve classification accuracy, the dimension is reduced from 512 to 160 or 100 by using Fisher linear discriminant analysis [12].

For recognition, there are two stages: coarse classification using the Euclidean distance and fine classification using the modified quadratic discriminant function (MQDF2) [13]. We select n candidate classes in accordance with the coarse classification. Then, the fine recognition is applied to only the candidate classes. We use k principal eigenvectors for each class.

2.4 Improvement in coarse classification

Classification using MQDF2 is computationally expensive. To reduce the cost when recognizing a large number of character categories, from thousands to tens of thousands in Chinese, Japanese, or Nom, coarse to fine classification is commonly used. Coarse classification is performed to reduce the number of candidates subjected to fine classification. In coarse classification, the k-nearest-neighbor (k-NN) rule is commonly used to find the nearest neighbors of an unknown pattern. The mean vector of the remaining samples of each class is used as the prototype of the class, and Euclidean distances from an input pattern to prototypes are calculated to obtain k candidate classes with minimum distances.

In another study, prototype learning algorithms, including learning vector quantization (LVQ) [14], GLVQ [15], and minimum classification error (MCE) [16], generally outperformed the k-NN rule with the original training patterns as prototypes [17]. GLVQ yielded the best results for Chinese character recognition [17,18]. Using prototypes trained on GLVQ resulted in higher accuracy than using the traditional prototypes, which use the mean vectors.

An approximate search method based on the k–d tree algorithm produced a promising result among nearest neighbor search methods [19]. It has only one parameter to set, bound error ε. The larger the bound error, the faster the searching. Using a k–d tree increases the speed of candidate searching compared with the traditional sequential search.

We improved the coarse classification by using the k–d tree structure algorithm to increase the recognition speed and by using the GLVQ prototype learning algorithm to improve the recognition accuracy. As a result, coarse classification is speeded up while keeping the same recognition rate.

3 Graphical user interface

Due to noise, degradation, distortion, variation, and so on, perfect results for binarization, segmentation, and recognition are unlikely. Hence, users must be able to review the processed results and correct any errors. We thus developed an interface that supports user review and correction.

The interface and basic functions are based on a system for archiving images of excavated wooden tablets called "mokkan" from ancient ruins in Japan [20]. Using the interface shown in Fig. 6, a user can work on several pages

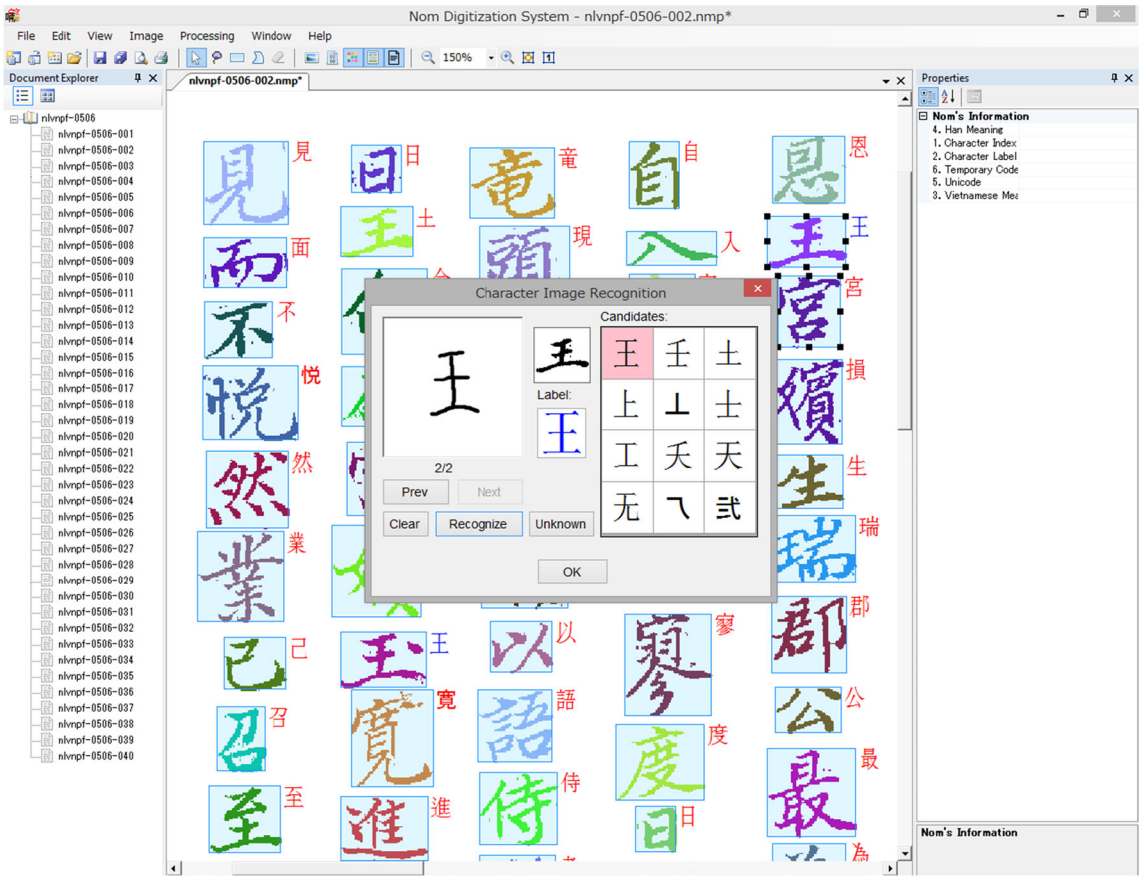

Fig. 6 Screenshot of graphical user interface

Table 1 Character pattern sets used

Char. pattern sets	No. categories	No. patterns	Source
JIS	4060	3,983,823	ETL9B, JEITA-HP (DATASET-A and DATASET-B), NTT-AT, and TUAT HANDS-Nakayosi and Kuchibue
Art_Nom_Small	7660	17,960,880	27 CJKV fonts
Art_Nom_Large	32,695	28,035,360	15 CJKV fonts
Jis1	2965	766,915- training and 675,840-testing	TUAT HANDS-Nakayosi and Kuchibue
Jis1&2	6355	1,965,041-training and 1,361,647-testing	TUAT HANDS-Nakayosi and Kuchibue, Tehon
Real_Nom_Char	2539	13,111	47 real Nom pages

of a document. Both basic functions, including file/project management and image viewing/processing (zoom, rotate, grayscale, adjust brightness/contrast or hue/saturation), and advanced functions, including binarization, segmentation, and recognition, are supported. Moreover, a set of tools is provided for checking the results of the advanced functions. An eraser tool can be used to delete noise in a binarized image. A selection tool can be used to binarize and segment a specified area. A rectangle and lasso tool can be used to segment a specified character area by hand. A property window can be used to edit the label of a character pattern. Users can also use the character image recognition window, as shown in Fig. 6, either to rewrite misrecognized characters and thereby enable the recognizer for the extended categories to subsequently recognize them or to assign them to the *Unknown* category. Some characters are not readable and other characters are not found in dictionaries. They are probably not within the extended set, so we label them *Unknown.*

4 Preparation of character pattern sets for system evaluation

To evaluate the system, we needed a large set of sample patterns. For Nom characters, however, there is no such character pattern set. Therefore, we prepared artificial Nom character patterns to use in addition to a small set of real patterns. We also exploited the fact that Nom, Chinese, and Japanese share a large set of characters by using patterns for Chinese and Japanese handwriting recognition for training and testing the system. Table 1 summarizes the character pattern sets we used. The following sections describe them in more detail.

4.1 Determination of Nom categories

As of now, only a few Nom documents have been digitized, and the Nom character set has not been completely defined. There is no statistical information about how many Nom characters there are in total, how many there are in common use, etc. Hence, we determined the number of regularly used categories and of the extended set of categories on the basis of the list of regular Nom characters at http://www.chunom.org/pages/charset as well as on the basis of the Chinese BIG5 level 1 and GB2312 level 1 character sets, the Japanese JIS level 1 Kanji character set, and the character sets of available Nom fonts. The numbers of categories in these character sets are listed in Table 2.

Table 2 Number of categories in character pattern sets by language

Language	Char. pattern sets	No. categories	No. categories without duplication
Japanese	JIS level 1	2965	7660 32,695
Chinese	BIG5 level1	6572	
	GB2312 level 1	3755	
Nom	Regular*Nom*	770	
	Nom Na Tong	22,573	–
	Chu Nom Khai	20,308	
	Chu Nom Minh	20,516	
	Chu Nom Minh U	21,515	

The number of categories (7660) for scanned and segmented patterns comes from the character sets of Japanese Kanji level 1 (JIS level 1), traditional and simplified Chinese character level 1 (BIG5 level 1 and GB2312 level 1), and Regular Nom without duplications. The number of categories (32,695) for handwritten input comes from the character set of 7660 categories and all categories in Nom fonts (Nom Na Tong, Chu Nom Khai, Chu Nom Minh, Chu Nom Minh U), again without duplications.

4.2 Preparation of training patterns

To construct an OCR engine for Nom, we must prepare a database of sample patterns. In our previous work [3], we used Kanji character patterns of Japanese databases: ETL9B, JEITA-HP (DATASET-A and DATASET-B), NTT-AT, and the TUAT HANDS databases—Kuchibue and Nakayosi [21]. The first two databases store off-line character patterns (bitmap images), while the other three were converted from online patterns (pen-trace patterns) by connecting succes-

Fig. 7 Font character patterns and their artificial patterns

sive pen-points and thickening the stroke. The total numbers of categories and characters in the training set were 4060 and 3,983,823, respectively. We call this character pattern set *JIS*.

Since many Nom character categories are not included in the Japanese databases, we constructed a database of artificial patterns created from font character patterns. A full set of 27 CJKV fonts (computer fonts containing a wide range of Chinese, Japanese, Korean, and Nom characters) such as Arial Unicode MS, MS Mincho, SimSun, MingLiU, and Nom Na Tong was used to create patterns for 7660 categories, while a set of 15 CJKV fonts from the full set was used to create patterns for 32,695 categories.

To create artificial patterns, we first placed a font character with a size of 48 points on an image of 96 × 96 pixels to get a base pattern. Then we applied four linear distortion models (rotate, shear, shrink, perspective) to this pattern [22] by changing the angle from $-11°$ to $+11°$ every 2° step to get 84 patterns, applied the nonlinear distortion model proposed by Leung and Leung [23] to obtain 15 patterns, and applied affine transformation with random parameters to generate 20 patterns. We generated 120 artificial patterns in total, including the base pattern. Example font character patterns and their artificial patterns are shown in Fig. 7.

Consequently, we constructed two sets of character patterns: one with 17,960,880 patterns generated from 149,674 font patterns for 7660 categories and the other with 28,035,360 patterns generated from 233,628 font patterns for 32,695 categories. The former, called Art_Nom_Small, was used for training the recognizer used to automatically recognize the scanned and segmented character patterns. The latter, called Art_Nom_Large, was used for training the recognizer used to recognize handwritten input.

4.3 Preparation of Japanese character patterns

Since *Art_Nom_Small* and *Art_Nom_Large* store only artificial patterns, we could not trust the results of their use in the evaluation of the recognition method, so we used two character pattern sets of Japanese off-line handwritten character patterns. The first, *Jis1*, is an off-line version with 2965 JIS level-1 Kanji categories obtained from the TUAT HANDS databases—Nakayosi and Kuchibue [21]—with 766,915 patterns for training and 675,840 patterns for testing. The second, Jis1&2, with 6355 JIS level-1 and level-2 Kanji categories includes the *Jis1* character pattern set and more patterns mainly for JIS level-2 Kanji categories. It additionally stores the TUAT HANDS collection of 2319 JIS level-2 Kanji categories for 20 participants (one pattern for one category and one participant) with artificial patterns made from the collection and TUAT HANDS-Tehon, a standard character pattern database including 6355 JIS level-1 and level-2 Kanji categories (one pattern for one category) with artificial patterns made from the database. A nonlinear distortion model [23] and affine transformation with random parameters were used to generate artificial patterns. For each category in the JIS level-2 collection, 192 patterns (12 participant patterns and their artificial patterns) and 128 patterns (8 participant patterns and their artificial patterns) were prepared for training and testing, respectively. Similarly, for each category in Tehon, 91 patterns (including the original pattern) and 60 patterns were prepared for each category for training and testing, respectively. In summary, Jis1&2 contained 1,965,041 training patterns and 1,361,647 testing patterns.

4.4 Preparation of real Nom patterns

If we had had a large benchmark database of Nom character patterns, we could have straightforwardly evaluated the document recognition method and system. We could have separated the patterns and cross-validated them by using a portion for testing and the rest for training and then switching their roles to get the average score. Since we did not have such a database, we prepared training patterns as described in 4.2. For testing, we prepared a set of ground truth documents with labeled character patterns.

We selected 10 real documents with different writing layouts and writing styles. For each document, we selected four or five page images that had more characters than other pages.

We selected 47 page images in total to be annotated as ground truth. We used the interface of our document recognition system to annotate the documents. There were five main steps in the annotation task:

1. Create new document project with selected page images.
2. Binarize page images in batch mode.
3. Segment page images in batch mode.
4. For each page image,
   - 4.1. Edit segmentation results.
   - 4.2. Use 7660-category recognizer to recognize segmented characters automatically.
   - 4.3. Check recognition result for each segmented and recognized character image and edit its label if misrecognized by using edit window, as shown in Fig. 8. If label is wrong, segmented character image is recognized using 32,695-category recognizer. If character image is not recognized, user can use own handwritten character to get label from candidate list. If character is still unrecognized, it is labeled *Unknown*.
5. Save the segmentation and recognition results to files.

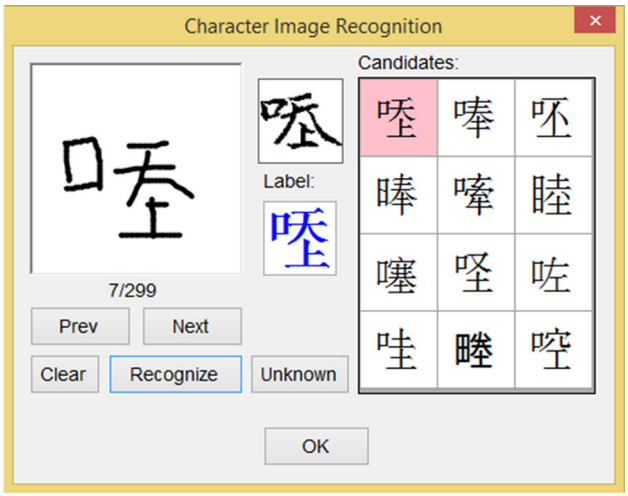

Fig. 8 Edit window for assigning character label

In this way, we annotated 47 page images and collected 13,111 patterns in 2539 categories. Among them, 1442 character images were labeled *Unknown*. We denote the dataset of collected Nom character patterns as Real_Nom_Char. The results of this segmentation and recognition process are listed in Table 3.

Since 47 pages are too few for evaluating the segmentation, we also sampled 525 pages having a variety of layouts and annotated the ground truth for each page layout. We call this set of pages *Real_Nom_Page*.

5 Evaluation

We conducted a preliminary evaluation of the recognition method and document recognition system in their current state since a large set of sample patterns with ground truths was not available. Six experiments were performed, as shown in Table 4.

We performed the experiments on a PC with an x64-based processor, an Intel® Core™ i7-2600 CPU (3.40 GHz), and 8.0 GB RAM.

5.1 Nom character recognition using Japanese character patterns

As mentioned in Sect. 4, many Nom characters are included in Japanese. To evaluate the recognition performance of the proposed method, we tested it using the two character pattern sets of Japanese off-line handwritten character patterns, *Jis1* and Jis1&2.

Table 3 Results of segmentation and recognition process

Document	No. pages	No. categories	No. char.	Average no. char./pages	No. unknown	No. labelled
nlvnpf-0023	5	675	1704	341	335	1369
nlvnpf-0073	5	569	940	188	55	885
nlvnpf-0141	5	488	981	196	142	839
nlvnpf-0221	5	610	1361	272	165	1196
nlvnpf-0439-01	5	542	1753	351	174	1579
nlvnpf-0501	3	287	1176	392	279	897
nlvnpf-0506	5	616	1765	353	133	1632
nlvnpf-0510-01	5	421	958	192	96	862
nlvnpf-0510-02	5	271	632	126	33	599
nlvnpf-0991-01	4	504	1841	460	30	1811
Total	47	2539	13,111	279	1442	11,669

Table 4 Experiments

Experiment	Training char. pattern sets	Testing char. pattern sets	Purpose
1	Jis1 training and Jis1&2 training	Jis1 testing and Jis1&2 testing	Evaluate Nom character recognition using Japanese character patterns
2	Jis1 training and Jis1&2 training	Jis1 testing and Jis1&2 testing	Evaluate improvement in coarse character classification
3	Art_Nom_Small and Art_Nom_Large	Art_Nom_Small and Art_Nom_Large	Evaluate training using artificial patterns generated from CJKV fonts
4	Art_Nom_Small	Real_Nom_Char and Real_Nom_Page	Evaluate real Nom document recognition
5	JIS, Real_JIS_in_Nom, Art_Nom_Small or two ways of combination	Real_Nom_Char	Compare training using only artificial patterns with that using real patterns
6	Art_Nom_Small and Real_JIS_in_Nom or Art_Nom_Large and Real_JIS_in_Nom	Real_Nom_Char	Analyze effect of number of categories on testing performance

Table 5 Memory size, speed, and recognition rate

Char. pattern sets	No. categories	$(d, n, k) = (100, 10, 10)$			$(d, n, k) = (160, 100, 50)$		
		Mem. size (MB)	Speed (ms/pat)	Recog. rate (%)	Mem. size (MB)	Speed (ms/pat)	Recog. rate (%)
Jis1	2965	6.50	0.138	97.20 (99.28)	47.85	1.781	97.94 (99.91)
Jis1&2	6355	13.93	0.233	96.63 (98.95)	102.55	1.957	96.77 (99.82)

Three main parameters affect recognition performance: the dimension of reduced feature vectors d, the number of candidates n used in coarse classification, and the number of principal eigenvectors k used in fine classification (MQDF2). We used two often-used parameter combinations of (d, n, k): (100, 10, 10) and (160, 100, 50). The memory size, speed, and recognition rate are summarized by character pattern set in Table 5.

The recognition rate was quite high for both character pattern sets. With a larger number of dimensions for reduced feature vectors d, a higher number of candidates n, and a higher number of principal eigenvectors k, the accuracy was higher but the memory size of the dictionary was greatly increased, and the speed was seriously degraded. Therefore, we set the (100, 10, 10) parameter set as the default configuration for subsequent experiments.

The results of this experiment showed that the recognition method should work well for Nom character patterns if a training set of character patterns as large as *Jis1* and *Jis1&2* is available.

5.2 Improvement in coarse classification

To evaluate the improvement in coarse classification, we first used *Jis1* and *Jis1&2* with $d = 100$ and $k = 10$ to evaluate the effect of using GLVQ. The number of candidates n was varied from 5 to 100. The recognition rates of the recognizers with and without prototype learning using GLVQ for *Jis1* and *Jis1&2* are plotted in Fig. 9.

The use of GLVQ increased accuracy for both *Jis1* and *Jis1&2*, particularly when n was less than 10. When n was 10, accuracy improved 0.16 points (from 97.20 to 97.36 %) for *Jis1* and 0.23 points (from 96.63 to 96.86 %) for *Jis1&2*. Since the recognition rates when n was 10, i.e., 97.36 % for *Jis1* and 96.86 % for *Jis1&2*, were almost the same as those when n was 100, there is no need to increase the number of candidates to improve accuracy. Ten candidates are sufficient when using GLVQ.

We next used *Jis1&2* without prototype learning to evaluate the effect of using a *k–d* tree on speed in coarse classification. Without a *k–d* tree, the accuracy was 93.11 %, and the times taken to search for the top 10 and top 50 candidates were 0.280 ms and 0.374 ms, respectively.

Figure 10 shows that when bound error ε was between 1.5 and 2.0, the search time was greatly reduced while the recognition rate remained almost unchanged. As a result, in searching for the top-10 candidates with $\varepsilon = 2.0$, the search time was reduced to 0.129 ms, from 0.280 ms (46.07 %) while the recognition rate was 93.05 %, down only 0.06 points from 93.11 %. Similarly, in searching for the top-50 candidates with $\varepsilon = 2.25$, the search time was reduced to 0.171 ms, two

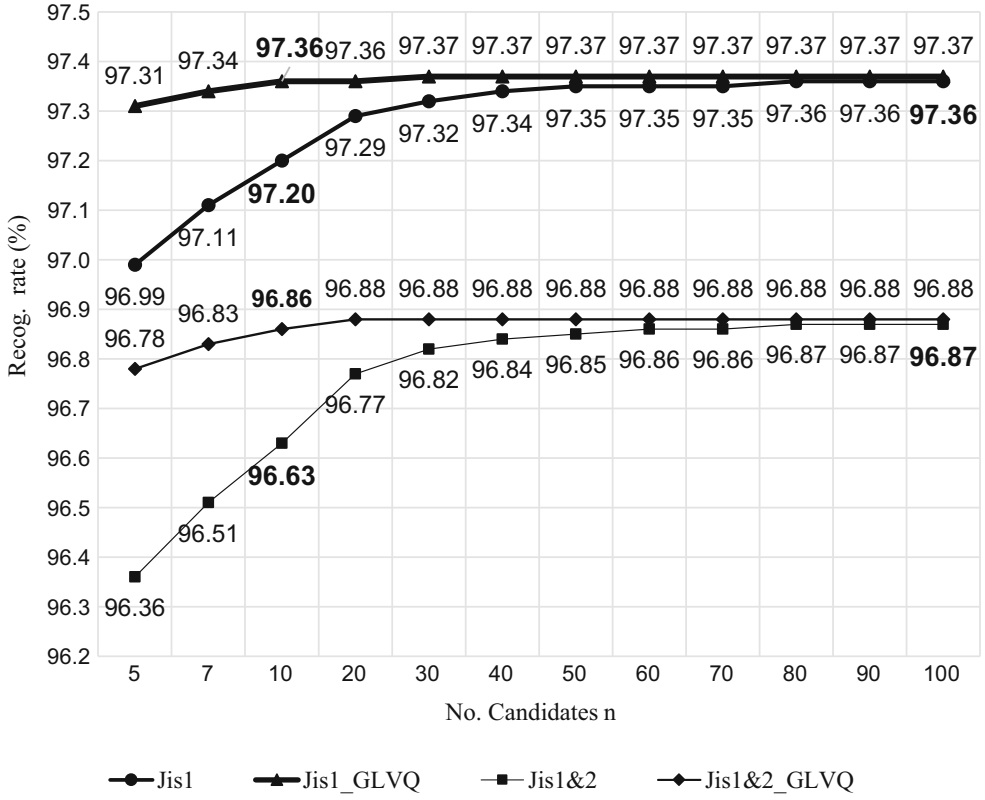

Fig. 9 Comparison of accuracies with and without prototype learning using GLVQ

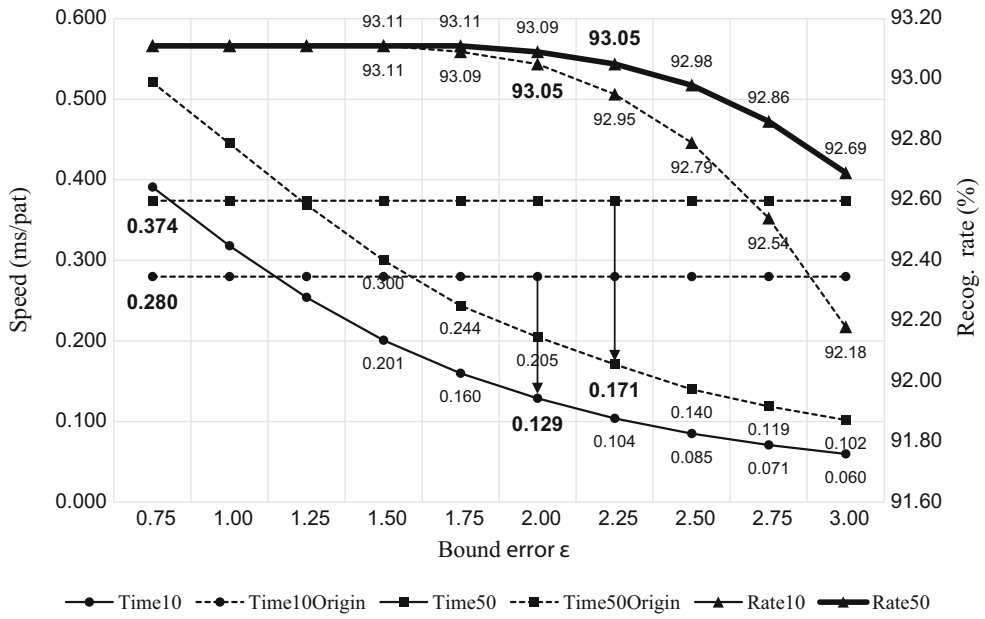

Fig. 10 Accuracy and speed trade-off in coarse classification when using *k–d* tree

times faster, while the recognition rate decreased by the same amount.

Finally, we used *Jis1* and *Jis1&2* to evaluate the effect of the GLVQ and *k–d* tree combination on coarse classification. When using GLVQ, we can get a higher recognition rate, so we can reduce the recognition time by using a larger value for ε. We thus set it to 2.25. Table 6 shows the recognition rate and coarse classification speed for the two character pat-

Table 6 Recognition rate and coarse classification speed with GLVQ and *k–d* tree

Char. pattern sets	Performance	Original	With GLVQ	With *k–d* tree	With GLVQ and *k–d* tree
Jis1	Recog. rate (%)	97.20	97.36	97.08	97.25
	Speed (ms/char)	0.133	0.147	0.076	0.086
Jis1&2	Recog. rate (%)	96.63	96.86	96.52	96.75
	Speed (ms/char)	0.283	0.303	0.131	0.151

tern sets. The average time for fine classification for each pattern was 0.012 ms. When GLVQ was used, the recognition rate was improved by 0.16 points for *Jis1* and 0.23 points for *Jis1&2*. Moreover, with *k–d* tree, the recognition time was reduced ∼35 % for *Jis1* and ∼45 % for *Jis1&2* from the original time while the recognition rate remained almost the same, or even improved. This shows that, the larger the category set, the better the accuracy, and the higher the speed.

5.3 Training using artificial patterns generated from CJKV fonts

Since artificial patterns were used to train the Nom recognizers and we did not have a large set of real patterns, we conducted an experiment to evaluate the memory size and speed of the recognizers. To maintain a good balance among speed, accuracy, and dictionary size, we set d to 100 and n to 10 when building the Nom recognizer on the two character pattern sets of the Nom artificial character patterns, *Art_Nom_Small* and *Art_Nom_Large*. The dictionary sizes were 16.79 and 71.65 MB, respectively. The recognizer trained using *Art_Nom_Small* was used to classify hundreds or thousands of segmented character images at a time, so we simply set the number of candidates n to 10. The recognizer trained using *Art_Nom_Large* had higher accuracy and lower speed requirements, so we set n to 50.

We used *Art_Nom_Small* and *Art_Nom_Large* for testing as well as training to evaluate the recognition speed. The speed for *Art_Nom_Small* was 0.276 ms/pattern and that for *Art_Nom_Large* was 1.058 ms/pattern. From these results, i.e., a speed of about 0.25 ms/pattern for *Art_Nom_Small*, we estimated the recognition time for a document page with 200 characters and for a document with 100 pages to be approximately 0.05 s and 5 s, respectively. Although the accuracy may be questionable, it was 98.54 % for *Art_Nom_Small* and 95.61 % for *Art_Nom_Large*. These values would be lower for real testing patterns, but we can consider them to be the upper limit if we have a sufficient number of training and testing patterns.

5.4 Real Nom document recognition

We first conducted a small experiment to compare labeling using the proposed system with manual labeling. We had a Vietnamese graduate student label characters manually. He spent 5 h labeling two Nom pages, one containing 370 characters and one containing 350. He consulted dictionaries frequently and labeled many characters *Unknown*. He complained that the work was boring and tedious. We thus decided to forgo asking more students to do the same task as it was evident that having ordinary people label Nom characters manually is impractical. This is a serious problem for an almost lost language like Nom. Although manual labeling by an expert would result in better performance, there are less than 100 scholars worldwide who can read Nom. The proposed system, however, would help an expert in the labeling task.

Three elements affect system performance: segmentation, automatic recognition, and the user's experience level. Using the recursive X–Y cut method and a Voronoi-based segmentation method, we evaluated the segmentation performance on *Real_Nom_Page* [3]. The metric for performance is an f-measure: the harmonic mean of the precision and recall. Precision is the number of correct character segments extracted over the number of all extracted character segments. Recall is the number of correct character segments extracted over the number of true character segments. Use of the Voronoi-based method alone and the Voronoi and X–Y cut methods combined resulted in f-measures of 80.19 and 85.77 %, respectively.

Although the combined method produced a higher f-measure, the performance depends on the document layout, so the system provides an option for selecting one of the three methods. In preparing *Real_Nom_Char*, we used different methods for different layouts to segment documents into characters. Then we edited the segmentation results by hand. Manual editing took ∼10 min per page on average (from 5 to 20 min) for the Nom documents listed in Table 3.

After editing the segmentation results, we applied automatic recognition using the recognizer trained on *Art_Nom_Small*. GLVQ was used for coarse classification without a *k–*

Table 7 Recognition rate for *Real_Nom_Char*

Document	No. cat.	No. unrecog. cat.	No. char.	No. correct	Recog. rate (%)	Recog. rate (%) (top 10)
nlvnpf-0023	674	291	1369	746	54.49	70.78
nlvnpf-0073	568	179	885	606	68.47	79.89
nlvnpf-0141	487	182	839	498	59.36	71.99
nlvnpf-0221	609	244	1196	716	59.87	69.23
nlvnpf-0439-01	541	118	1579	1171	74.16	83.09
nlvnpf-0501	286	136	897	394	43.92	60.76
nlvnpf-0506	615	170	1632	1095	67.10	78.31
nlvnpf-0510-01	420	123	862	552	64.04	78.31
nlvnpf-0510-02	270	79	599	432	72.12	82.14
nlvnpf-0991-01	503	70	1811	1599	88.29	95.58
Total	2538	865	11,669	7809	66.92	78.34

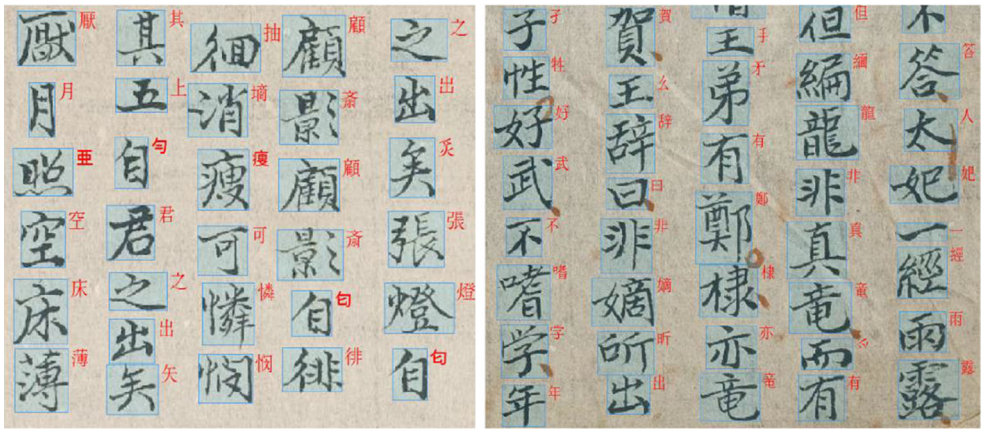

Fig. 11 Example recognition results

d tree since a high recognition rate was required, while high speed was not. The recognition rates for *Real_Nom_Char* are listed in Table 7. Note that the rate was not calculated for *Unknown* character patterns.

In the automatic recognition, there were 3860 misrecognized character images out of 11,669 character images. Together with 1442 *Unknown* character images, there were 5302 character images in total that needed to be edited or annotated by hand. The automatic recognition rate of ~67 % is still too low but nevertheless reduces manual editing significantly. Example recognition results for Nom historical documents are shown in Fig. 11.

We then asked a non-expert to annotate the ground truths for 47 page images from 10 documents. It took him about ~30 min per page on average to edit the character recognition results after spending 10 min editing the segmentation results, as mentioned above. Thus, it took ~40 min in total on average to annotate a page image. The annotator was a Vietnamese who was learning Japanese but knew less than 1000 basic Japanese Kanji characters and did not know *Nom script*. Hence, it took him much time to check the recognition results and input handwritten characters to get the correct labels. An annotator familiar with the system and proficient in Chinese or Japanese would be able to process a page image more quickly. Actually, for the first several page images, it took the Vietnamese non-expert over 1 h for each page image but then only ~20 min for each one thereafter. This is far quicker than manual labeling.

5.5 Comparison of training using only artificial patterns with that using real patterns

We used only artificial patterns for training for all Nom character categories. Since Japanese Kanji and Nom share a considerably large set of characters, we can test the effect of real patterns by using these characters for training. The *JIS* character pattern set stores character patterns for 4060 Japanese Kanji categories, and 3,930,488 patterns (denoted by *Real_JIS_in_Nom*) in 3735 categories are shared with Nom. They were used for training the Nom automatic character recognizers. Although some of them are made from online patterns, they are real patterns except for stroke

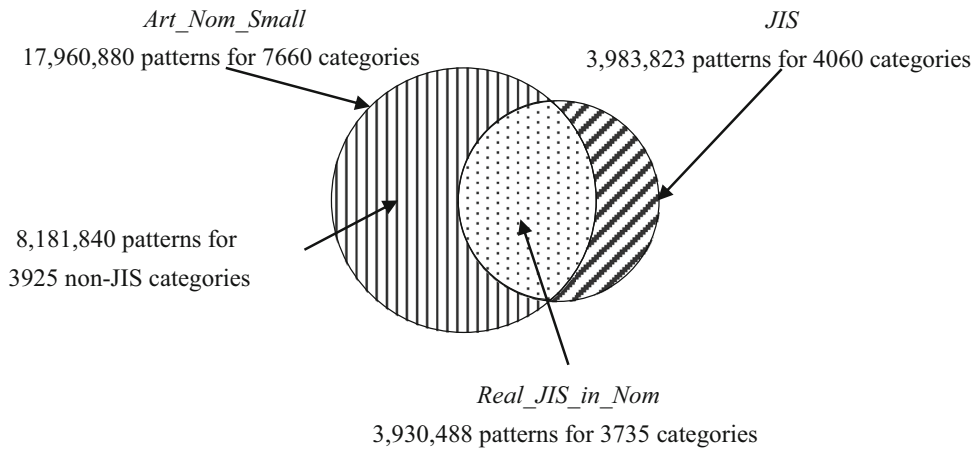

Fig. 12 Number of categories and character patterns in character pattern sets

Table 8 Recognition accuracies using different training patterns

Char. pattern sets for training	No. cat.	No. patterns	Recog. rate (%)	Recog. rate (%) (top 10)
Real_JIS_in_Nom	3735	3,930,488	62.90	74.31
JIS	4060	3,983,823	63.00	74.51
Art_Nom_Small	7660	17,960,880	66.92	78.34
Art_Nom+Real_JIS	7660	12,112,328	64.47	79.75
Art_Nom++Real_JIS	7660	21,891,368	71.48	82.95

width. Hence, we compared the recognizers trained using *Real_JIS_in_Nom* alone, *JIS* alone, and *Art_Nom_Small* alone, as well as the recognizer trained using 8,181,840 artificial patterns in 3925 non-JIS categories in *Art_Nom_Small* and the above-mentioned *Real_JIS_in_Nom* in place of artificial patterns for the shared 3735 categories (denoted by *Art_Nom+Real_JIS*), and that trained using all patterns of *Art_Nom_Small and Real_JIS_in_Nom* in 7660 categories (denoted by *Art_Nom++Real_JIS*). The numbers of categories and character patterns for the three training character pattern sets are illustrated in Fig. 12. The recognition rates and the recognition rates for the top 10 candidates of the five recognizers trained using these character pattern sets are shown in Table 8.

The results show that using only real patterns is not good enough for recognizing Nom characters. They also show that replacing artificial patterns with real patterns does not improve the recognition rate and that using them together with artificial patterns effectively improves the accuracy even though their number is not large.

5.6 Effect of number of categories on testing performance

The larger the category set used for recognition, the worse the recognition rate due to the inclusion of less frequently appearing categories. Although reducing the number of categories would improve the rate, the amount of manual tagging would increase due to the larger number of excluded categories. Thus, our last experiment compared the performance of the recognizer trained using *Art_Nom++Real_JIS* with 7660 categories and that trained using *Art_Nom_Large* & *Real_JIS_in_Nom* (denoted by *Art_Nom_Large++Real_JIS*) with 32,695 categories containing 11,669 character patterns in *Real_Nom_Char.*

Table 9 shows the recognition rates of the two recognizers. The rate difference was 2.4 points, equivalent to 280 misrecognized characters in 11,669 character patterns. This states that time must be taken to correct them if we use the recognizer for the large category set. As the number of recognized categories increases, more and more effort and time are required to correct misrecognized characters.

6 Conclusion

We have described a document recognition system for digitizing Nom historical documents. It uses binarization, recursive X–Y cut, and Voronoi diagrams for segmentation, a k–d tree and GLVQ for coarse classification, and MQDF2 for fine classification. The system has an interface through which a user can check the results, change binarization methods,

Table 9 Recognition rates of two recognizers trained using *Art_Nom ++ Real_JIS* and *Art_Nom_Large ++ Real_JIS*

Character pattern sets	No. cat.	Recog. rate (%)	Recog. Rate (%) (top 10)	No. misrecognized
Art_Nom++Real_JIS	7660	71.48	82.95	3328
Art_Nom_Large++Real_JIS	32,695	69.08	86.03	3608

rectify segmentation, and input correct character categories manually.

Character recognition is done in two stages. First, segmented character patterns are classified using a 7660-category recognizer. Second, user handwritten characters are classified using a 32,695-category recognizer in order to identify the correct categories for patterns unrecognized in the first stage. The same recognition method is used in both stages, but the training character pattern sets differ in terms of the number of categories.

Training patterns were artificially generated from 27 Chinese, Japanese, and Nom character fonts since the three languages share a considerable number of character categories, and ground truth real patterns are not available for most Nom categories. Using real handwritten patterns from off-line Japanese character pattern databases together with the artificial patterns effectively improved the recognition rate.

We optimized the parameters to maintain a good balance among accuracy, speed, and dictionary size. The two-stage architecture for character recognition was effective. Confining the character categories used for recognition in the first stage to 7660 frequently appearing categories increased the recognition rate to 66.92 % from 55.50 % for the extended set, which reduced the time and labor needed to manually tag unrecognized patterns. The recognition time for a document page with approximately 200 characters was about 0.05 s. The memory size was 16.79 MB.

We plan to improve the page layout analysis, use text recognition, and enhance the user interface so that the system can handle many kinds of Nom historical documents. Highlighting the characters recognized by OCR with a lower level of confidence would enable users to focus on the characters needing more checking.

Acknowledgments We thank the National Library of Vietnam and the Vietnamese Nom Preservation Foundation for providing the Nom historical document pages. This research is being supported by Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science (JSPS) (contract numbers (B) 24300095 and (S) 25220401).

References

1. Kim, M.S., Jang, M.D., Choi, H.I., Rhee, T.H., Kim, J.H., Kwag, H.K.: Digitalizing scheme of handwritten Hanja historical documents. In: Proceedings of the 1st International Workshop on Document Image Analysis for Libraries, USA, pp. 321–327, Jan. 2004
2. Shih, V.J., Chu, T.L.: The Han Nom Digital Library. In: The International Nom Conference, The National Library of Vietnam, Hanoi, pp. 12–14, Nov. 2004
3. Phan, T.V., Zhu, B., Nakagawa, M.: Development of Nom character segmentation for collecting patterns from historical document pages. In: Proceedings of 1st International Workshop on Historical Document Imaging and Processing, China, pp. 133–139, Sep. 2011
4. Phan, T.V., Zhu, B., Nakagawa, M.: Collecting handwritten Nom character patterns from historical document pages. In: Proceedings of 10th IAPR International Workshop on Document Analysis Systems, Australia, pp. 344–348, Mar. 2012
5. Su, B., Lu, S., Tan, C.L.: Binarization of historical handwritten document images using local maximum and minimum filter. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, USA, pp. 159–165, Jun. 2010
6. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
7. Kittler, J., Illingworth, J.: Threshold selection based on a simple image statistics. Comput. Vis. Graphics Image Process. **30**, 125–147 (1985)
8. Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Cardona, A.: Fiji: an open-source platform for biological-image analysis. Nat. Methods. **9**(7), 676–682 (2012)
9. Tsukumo, J., Tanaka, H.: Classification of handprinted Chinese characters using non-linear normalization and correlation methods. In: Proceedings of the 9th International Conference on Pattern Recognition, Italy, pp. 168–171 (1988)
10. Liu, C.L.: Normalization-cooperated gradient feature extraction for handwritten character recognition. Pattern Anal. Mach. Intell. IEEE Trans. **29**(8), 1465–1469 (2007)
11. Kawamura, A., Yura, K., Hayama, T., Hidai, Y., Minamikawa, T., Tanaka, A., Masuda, S.: Online recognition of freely handwritten Japanese characters using directional feature densities. In: Proceedings of the 11th International Conference on Pattern Recognition, Netherlands, 2, pp. 183–186 (1992)
12. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, San Diego (1990)
13. Kimura, F., Takashina, K., Tsuruoka, S., Miyake, Y.: Modified quadratic discriminant functions and the application to Chinese character recognition. IEEE Trans. PAMI **9**(1), pp. 149–153 (1987)
14. Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J., Torkkola, K.: LVQ PAK: The learning vector quantization program package. In: Technical Report, Laboratory of Computer and Information Science Rakentajanaukio 2(C), pp. 1991–1992 (1996)
15. Sato, A., Yamada, K.: Generalized learning vector quantization. In: Proceedings of the 1995 Conference on Advances in Neural Information Processing Systems, vol 8, pp 423–429. MIT Press, Cambridge, USA (1996)
16. Juang, B.-H., Katagiri, S.: Discriminative learning for minimum error classification. Signal Process. IEEE Trans. **40**(12), 3043–3054 (1992)
17. Liu, C.L., Nakagawa, M.: Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition. Pattern Recognit. **34**(3), 601–615 (2001)
18. Fukumoto, T., Wakabayashi, T., Kimura, F., Miyake, Y.: Accuracy improvement of handwritten character recognition by GLVQ. In: Proceedings of the 7th International Workshop on Frontiers in handwriting recognition, pp. 687–692. The Netherlands (2000)
19. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM **18**(9), 509–517 (1975)
20. Phan, T.V., Nakagawa, M., Baba, H., Watanabe, A.: MokkAnnotator - A system for archiving Mokkan images. In: Proceedings of the 16th Biennial Conference of the International Graphonomics Society, Japan, pp. 54–57, Jun. 2013
21. Nakagawa, M., Matsumoto, K.: Collection of on-line handwritten Japanese character pattern databases and their analysis. Doc. Anal. Recognit. **7**(1), 69–81 (2004)

22. Chen, B., Zhu, B., Nakagawa, M.: Effects of generating a large amount of artificial patterns for on-line handwritten Japanese character recognition. In: Proceedings of the 11th International Conference on Document Analysis and Recognition, China, pp. 663–667, Sep. 2011

23. Leung, K.C., Leung, C.H.: Recognition of handwritten Chinese characters by combining regularization, Fisher's discriminant and transformation sample generation. In: Proceedings of the 10th International Conference of Document Analysis and Recognition, Spain, pp. 1026–1030 (2009)