Camera-based Kanji OCR for Mobile-phones: Practical Issues

Masashi Koga[1], Ryuji Mine[1], Tatsuya Kameyama[1], Toshikazu Takahashi[1],
Masahiro Yamazaki[2], and Teruyuki Yamaguchi[3]
[1]*Hitachi Central Research Laboratory, Tokyo, Japan*
[2]*Hitachi Ubiquitous System Platform R&D Laboratory, Yokohama, Japan*
[3]*Hitachi-Omron Terminal Solutions, Corp., Owariasahi, Japan*
{koga, mine, kameyama, takahash}@crl.hitachi.co.jp
m-yamaza@msrd.hitachi.co.jp, teruyuki_yamaguchi@hitachi-omron-ts.com

Abstract

A camera-based optical character reader (OCR) for Japanese Kanji characters was implemented on a mobile-phone. This OCR has three key features. The first is discriminative feature extraction (DFE) which enables a character classifier needing only small memory size. The second is a word-segmentation method specially designed for looking up Japanese words in a dictionary. The third feature is a GUI suitable for a mobile-phone. A prototype mobile-phone Kanji OCR was constructed and experimentally tested. Recognition accuracy of over 95% was obtained under the best conditions, which shows the potential of our prototype as a new type of electronic dictionary.

1. Introduction

The mobile-phone is one of the most commonly used appliances today. Many of them have cameras. Some have microprocessor units powerful enough to run optical character reader (OCR) software, though the processing power of a mobile-phone is about one-tenth that of a typical PC. Prototypes of mobile-phone OCRs for digits and the Roman alphabet appeared in 2001, and commercial products came to market in 2002 [1][2]. These recognize English words for looking-up in a dictionary, e-mail addresses, URLs, phone numbers, etc. Some even recognize Kanji [3]. We recently developed a prototype Kanji OCR for a mobile-phone which differs from previous ones in that it has an integrated word-matching mechanism. However, implementing a Kanji OCR in a mobile-phone is not straightforward for several practical reasons.

Firstly, a lot of storage is needed to recognize Kanji characters. Typical mobile-phones have many applications such as schedulers, address books, games, and e-mail programs. Each of them requires a certain amount of memory. Consequently, the total required memory becomes so large that memory is the most precious resource in a mobile-phone. More than 4,000 characters are used in the Japanese writing system and the memory size needed for a conventional Kanji OCR is prohibitively large.

Secondly, word segmentation is problematic in the case of Kanji characters. There are no spaces between words in Japanese sentences. Morphological analysis is one solution to this problem, but it requires memory and additional processing time. Another difficulty is the small field of vision. A mobile-phone can process only a small part of a page because of its low resolution camera and low processing power. Thus words must be extracted even from incomplete sentences, making it difficult to apply conventional linguistic processing.

Thirdly, there are problems regarding user interaction. The current application of our mobile-phone OCRs is recognition of a word or short phrases. In such an application, the layout analysis should be interactive rather than automatic as in the conventional OCR, so that users can easily indicate what they want the mobile-phone to recognize.

In this paper, we outline the prototype Kanji mobile-phone OCR we have developed. This OCR recognizes machine-printed Japanese words and translates them into English. We also describe the three practical approaches that we use in the development: reduction of required memory size by a discriminative-feature extraction (DFE), a new word-segmentation method in which linguistic information is used in combination with positional information, and a newly designed graphical user interface (GUI). Finally, we show some experimental results that show the ease with which the prototype can be used.

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:18:41 UTC from IEEE Xplore. Restrictions apply.

2. Prototype mobile-phone Kanji OCR

2.1. Overview

A block diagram of the prototype mobile-phone Kanji OCR is shown in Fig. 1. As the hardware platform of the prototype, we used a typical mobile-phone available in Japan that had a camera and a color LCD. We coded the OCR software and GUI in C++. The compiled program is highly portable, running on the BREW(R) application execution environment. The OCR software recognizes Japanese words from a captured image. The Japanese-English dictionary-software application looks up the recognized word.

Figure 2 shows the display of our prototype mobile-phone Kanji OCR recognizing a Japanese word "画像," which means "image."

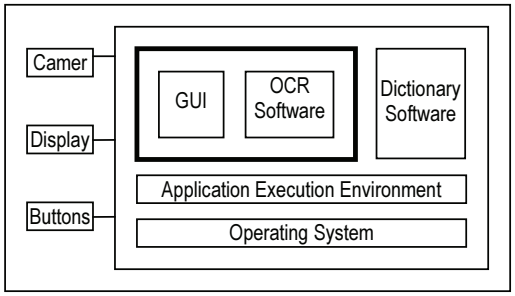


Figure 1. Block diagram of the prototype mobile-phone Kanji OCR

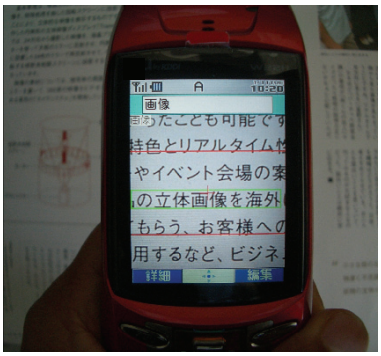


Figure 2. Prototype mobile-phone Kanji OCR recognizing the Japanese word "画像," which means "image."

The data flow of the OCR software is shown in Fig. 3. The OCR software consists of two parts: a text-line-extraction module and a word-recognition module. The text-line-extraction module detects text lines from the captured image, and the result is displayed by the GUI program. This process is repeated about ten times per second until a user presses the enter button on the mobile-phone. In this way, the user can adjust the position of the camera so that the prototype can detect the desired text line. When the enter button is pressed, the word-recognition module recognizes the word at the center of the detected text line. The recognized word is displayed, and sent to other applications such as a Japanese-English dictionary.

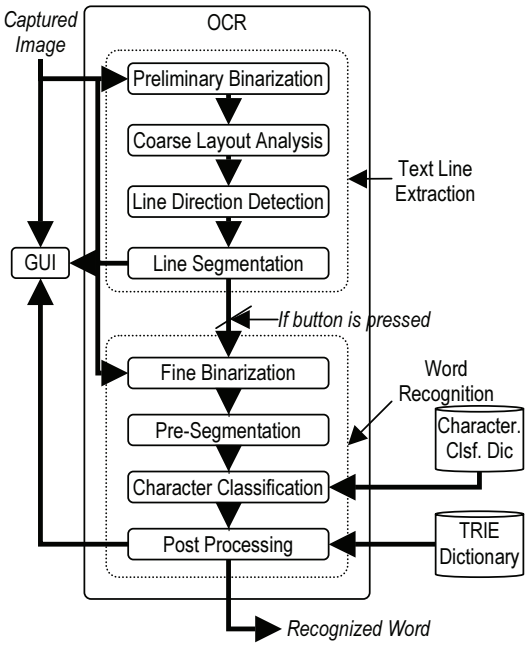


Figure 3. Data flow chart of the OCR software

2.2. Text-line extraction

2.2.1. Preliminary binarization. There are four steps in the text-line extraction. The first step is a preliminary binarization to generate a black-and-white image from the captured image. This is based on Bernsen's method [3] with some modification to accelerate the process. Only the intensity of the input color image is used for simplicity and to reduce the processing time. The image is encoded in terms of run length so that it is not necessary to access pixels in the following steps, which makes the process quicker.

2.2.2. Coarse-layout analysis. The second step is a coarse-layout analysis. As shown in Fig. 4, typical Japanese magazines are narrowly columned, sometimes only as wide as 12 characters. Sometimes separator lines are used. Illustrations and photographs

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:18:41 UTC from IEEE Xplore. Restrictions apply.

are printed very close to the characters. The coarse-layout analysis detects the boundaries of the column at the center of the image by a connected-component analysis.

2.2.3. Line-direction detection. The third step is line-direction detection. As shown in Fig. 4, Japanese text lines may be printed either horizontally or vertically. In this step, the direction of the text line is detected according to the distribution of the connected components. If the peak of the horizontal projection is larger than a predefined threshold, the text lines are regarded as horizontal, and vice versa.

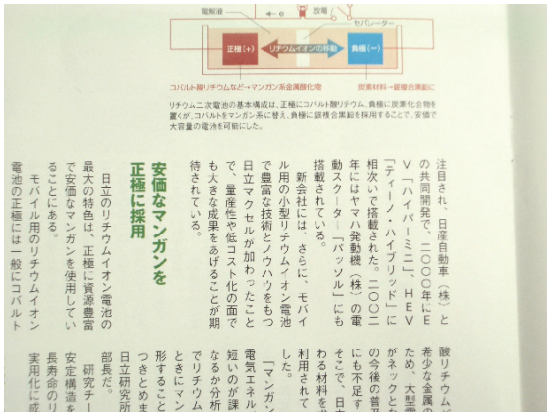


Figure 4. Examples of Japanese documents

2.2.4. Line segmentation. The fourth step is line segmentation. This detects text lines through connected-component analysis. Because the prototype is designed to recognize a word at the center of the image and translate the word, it outputs only the coordinates of the text line closest to the center of the image.

2.3. Word recognition

2.3.1. Fine binarization. In the first step of the word recognition, the region of the detected text line is binarized again. The text line is separated into small square blocks whose width equals the height of the text line. A threshold is calculated for each block through an iterative thresholding method [6]. This step is done quickly because the area is limited, even though it requires more processing time per unit of area than the preliminary binarization.

2.3.2. Pre-segmentation. After the fine binarization, the sub-images, each of which possibly corresponds to a character, are segmented. This step is called pre-segmentation. We applied a method presented in [5] for the presegmentation. The result of the pre-segmentation can be represented as a network as schematically shown in Fig. 5. We henceforth call it a segmentation hypothesis network.

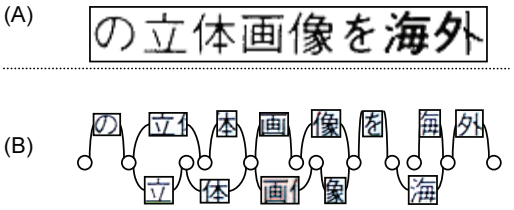


Figure 5. Example of a binarized image (A) and a schematic of a segmentation hypothesis network (B).

2.3.3. Character classification. The next step is character classification that recognizes 4,345 characters from segmented sub-images. For feature extraction, we adopt a gradient feature extraction method [6]. It is a cluster-based hierarchical minimum-distance classifier. The distance metrics is Euclidean distance. By separating the classification into two stages, we reduced the processing time by 75%. As described below, we use an LVQ and a DFE technique [7] in the training.

2.3.4. Post-processing. Finally, in the post-processing stage, the results of the character classification are analyzed and the most plausible word from the word dictionary is chosen. Unlike conventional methods, candidate words are ranked according to not only the likelihood of the characters but also the position of the candidate word. A candidate word close to the center of the image gets higher priority.

3. Discriminative Feature Extraction

The parameters for character classification generally require the largest portion of the memory used by Kanji OCRs. A typical OCR requires 0.5 Kbytes to 10 Kbytes per character, and 2MB to 40MB in total to recognize the 4,000 characters commonly used in Japanese documents. It is a prohibitively large amount of memory for current mobile-phones. Dimensionality reduction is an effective way to reduce the amount of memory needed. We use DFE for dimensionality reduction. The DEF is a technique that optimizes both subspace axes and classifier parameters. DFE has been shown to yield high classification

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:18:41 UTC from IEEE Xplore. Restrictions apply.

accuracy even on a low-dimensional subspace. The details of this method are given in [7]. In our Kanji OCR prototype, we reduced the dimensionality of the feature vector from 512 to 80 and thus drastically compressed the necessary memory size (to less than 100 bytes/character). The classification error for our validation samples was about 1% even after the dimensionality reduction. This error rate is low enough to be recovered by the post-processing.

4. Word segmentation

In our Japanese-English-dictionary application, the OCR has to recognize a word which a user wants to translate. However, the lack of spaces between words in Japanese sentences makes it impossible to extract a word from a text line by observing only the inter-character gaps.

In our prototype OCR, the word segmentation is not explicit. We solved the word segmentation problem as follows: in the post-processing, candidate words are extracted from a text line, and then one is selected that is plausibly the word the user wants the machine to recognize.

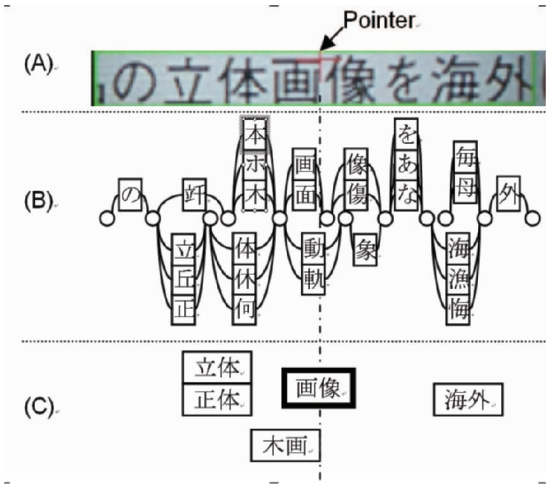


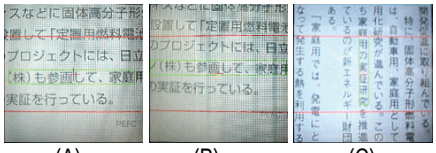
Figure 6. Example of a captured image with pointer (A), results of the character classification represented in terms of the candidate character network (B), and candidate words (C).

Candidate word extraction is a kind of word matching. Like the result of the pre-segmentation, the result of the character classification can also be represented in terms of a network (Fig. 6). We henceforth call it the candidate character network. In the candidate word extraction, partial paths that match words in a lexicon are extracted from the candidate character network. The lexicon is stored in a TRIE memory [8] to accelerate the matching process. We applied a heuristic method of merging the nodes of the TRIE to reduce the required memory size and accelerate the matching process.

Candidate selection is done according to the distance values given by the character classification, the consistency of the sizes and positions of characters, and the position of the word relative to the center of the image. In the example, the word “画像” is selected because it is closest to the center of the image that is shown by a cross pointer in Fig. 6.

5. Graphical user interface

In our prototype, the OCR is used for dictionary look-up in an interactive environment. A good man-machine interface is needed to enhance the usability of the prototype. In contrast to a sophisticated method to recognize a long string presented in [3], we designed a GUI that provide easy way to specify a word to recognize. The GUI of our prototype Kanji OCR shows the results of text-line detection frame-by-frame so that a user can easily adjust the camera and specify what he wants to translate. As shown in Fig. 7, a cross pointer is displayed at the center and the detected text-line is marked by a narrow rectangle. It tracks the text-line when the camera moves ((A) and (B)). It also automatically adapts to vertical text-lines (C).



(A) (B) (C)

Figure 7. GUI that displays detected horizontal (A) (B) and vertical (C) text lines

6. Experimental Results

6.1. Recognition accuracy

We captured 331 images and recognized them with Kanji OCR software running on a PC equivalent to that of the prototype. We tested the OCR under four sets of conditions: with either automatic or manual line direction detection, and with or without a word dictionary. Without the word dictionary, the OCR software outputs the detected path in the candidate-character-network having the smallest sum of distance values given by the character classification. For the N

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:18:41 UTC from IEEE Xplore. Restrictions apply.

(=331) samples, we measured the recognition accuracy according to two metrics:

n_1: number of samples for which the recognition results are exactly to the same as the word intended to be read.

n_2: number of samples for which recognition results include the word intended to be read.

Test results are shown in Table 1. Naturally, the recognition accuracy was much lower without the word dictionary, even when we ignored the excessive characters (n_2/N). If the line direction was input manually, the recognition accuracy was as high as 95.8%, however it fell to 90.6% with the automatic line direction detection. In fact, the recognition error included the line segmentation error. In an interactive GUI environment, though, we expect the recognition rate to be higher because the user is will press the key only after checking the line-segmentation results.

Table 1: Evaluation results

Line Direction Detection	Word Dictionary	Recognition Accuracy	
		n_1/N	n_2/N
Manual	Yes	95.8%	95.2%
Auto	Yes	90.9%	90.6%
Manual	No	81.8%	63.1%
Auto	No	74.1%	50.2%

6.2. Processing time

We measured the processing time of the OCR software running on a mobile-phone with a 150-MHz processor. The text line extraction and the word recognition required 0.1 and 0.5 s respectively.

6.3. User impressions

Test users reported a good impression of the prototype. Most people found it is easy to look up words with the prototype. However, we found some weak points during testing. Users had to learn to press the enter key in a way that did not move the camera, otherwise the image will be blurred. It could be frustrating for some users. We have tested another version of the OCR software that repeats the text-line detection and word-recognition automatically so that pressing the enter key is unnecessary. In this case, blurring was not a problem, but it works prohibitively slowly.

7. Summary

A Kanji OCR was implemented on a mobile-phone. Three techniques were used to solve the practical problems in the implementation: memory size reduction by DFE based character classification, a newly devised word-segmentation with an integrated word-matching mechanism, a GUI suitable for a mobile-phone. The prototype mobile-phone Kanji OCR was experimentally tested. The test results showed that recognition accuracy of over 95% can be achieved. To improve usability, though, we will have to shorten the processing time.

8. References

[1] "Multimedia Mobile Appliance with OCR", Axis, Vol. 92, P. 62, Tokyo, 2001 (in Japanese).

[2] S. Senda *et al.* "Camera-typing interface for ubiquitous information service," Proc. 2nd International Conference on Pervasive Computing and Communications, Florida, USA, 2004

[3] Shuji Senda, Kyosuke Nisiyama, and Toshiyuki Asahi, "Japanese Character Recognition with a Mobile Camera," Technical Report of the Institute of Electronics, Information and Communication Engineers, PRMU2004-124, pp. 7-12, 2004 (in Japanese)

[4] J. Bernsen, "Dynamic thresholding of grey-level images", Proc. 8th International Conference on Pattern Recognition, pp 1251-1255, Paris, France, 1986

[5] M. Sonka, V. Hlavac, and R. Boyle, "Image Processing: Analysis and Machine Vision," pp. 124-134, Thomson-Engineering, 2nd ed., London, 1998

[6] M. Koga *et al*. "Recognition of Monetary Amount Printed by Checkwriter," Systems and Computers in Japan, Vol. 35, No. 7, pp. 49-60, 2004

[7] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques, Pattern Recognition," 37(2): 265-279, 2004.

[8] C.-L. Liu, R. Mine, and M. Koga, "Building Compact Classifier for Large Character Set Recognition Using Discriminative Feature Extraction," to be published in proc. ICDAR2005

[9] M. Koga, R. Mine, H. Sako, and H. Fujisawa, "Lexical Search Approach for Character-String Recognition," Document Anal. Systems: Theory and Practice. Third IAPR Workshop, DAS'98. Selected Papers, pp. 115-129, Springer-Verlag

[10] H. Fujisawa, H. Sako, Y. Okada, and S.-W. Lee, "Information Capturing Camera and Developmental Issues," In Proc. Int. Conf. Document Analysis and Recognition, pp. 205-208, Bangalore, India, 1999

[12] D. Doermann, J. Liang, H. Li, "Progress in Camera-Based Document Image Analysis," Proc. ICDAR '03, pp. 606-616, Edinburgh, USA, 2003

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE

Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:18:41 UTC from IEEE Xplore. Restrictions apply.

IEEE COMPUTER SOCIETY