

Recognition of Expiration Dates Written on Food Packages with Open Source OCR

Kento Hosozawa, Ricky Hendra Wijaya, Tran Duy Linh, Hiroaki Seya, Masayuki Arai, Tsukasa Maekawa, and Kozo Mizutani

Abstract – As part of efforts to build a management system that can keep track of not only the food inside a refrigerator but also expiration dates, this paper introduces a smartphone application that can be used to recognize printed expiration dates. We began by investigating three open source optical character readers (OCRs) to determine the one most suitable to our needs, and then evaluated the selected OCR to determine how well it recognized expiration dates in numerous different environments. After encountering problems related to specific conditions, we designed and implemented an Android software application equipped with preprocessing functions to address them.

Index Terms—OCR, preprocessing, expiration date recognition, refrigerator food management system.

I. INTRODUCTION

Perishable foods are generally stored in refrigerators, but because people sometimes forget what foods have been placed inside their refrigerators, as well as their expiration dates, spoilage is common. With this point in mind, and as part of efforts to build a refrigerator food management system to solve such problems, this paper introduces a smartphone application that can be used to recognize printed expiration dates. The application functions include an image classifier, a barcode recognizer, and a reader capable of resolving the expiration dates written on a variety of food packages. To implement the application, we utilized an open source optical character reader (OCR).

The remainder of this paper is divided as follows. In the next section we outline our proposed system. In Section III, we evaluate three open source OCRs. Section IV depicts the performance evaluation of our selected OCR and the problems we encountered during adapting it to our use. In Section V, we explain how our newly developed Android application resolves those problems and recognizes expiration dates. Related works are discussed in Section VI, while conclusions and future work are discussed in Section VII.

Manuscript received July 15, 2018; revised October 12, 2018.

Kento Hosozawa was with Faculty of Science and Engineering, Teikyo University, 1-1 Toyosatodai, Utsunomiya, Tochigi, Japan. He is now with Nissin Kogyo Co., Ltd.

Ricky Hendra Wijaya is with Faculty of Engineering, Universitas Katolik Indonesia Atma Jaya, Indonesia.

Hiroaki Seya is with Faculty of Science and Engineering, Teikyo University, 1-1 Toyosatodai, Utsunomiya, Tochigi, Japan.

Tran Duy Linh, Masayuki Arai, Tsukasa Maekawa and Kozo Mizutani are with Graduate School of Science and Engineering, Teikyo University, 1-1 Toyosatodai, Utsunomiya, Tochigi, Japan (email: arai@ics.teikyo-u.ac.jp).

II. OUTLINE OF REFRIGERATOR FOOD MANAGEMENT SYSTEM

Fig. 1 depicts the outline of our proposed refrigerator food management system. If the label of an item to be stored in the user’s refrigerator has a barcode or a printed expiration date, the user takes a photograph of each with his or her smartphone. If the packaging does not have a barcode, the user takes a photograph of the item itself. The image is then uploaded to a server, which registers or decides the expiration dates depending on photograph provided.

The system consists of the following three core functions: (1) image classification, (2) barcode recognition, and (3) expiration date recognition (OCR).

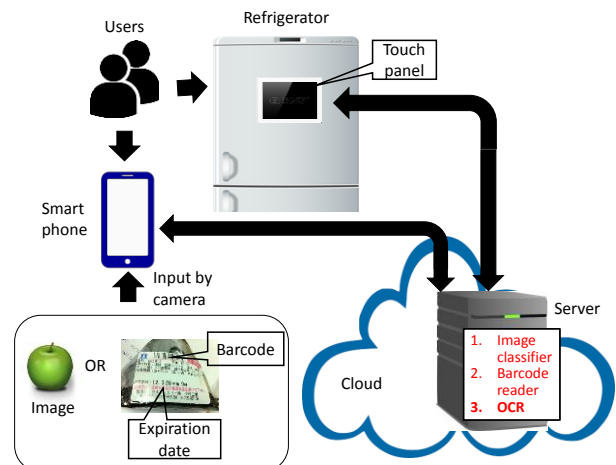


Fig. 1. Outline of our proposed refrigerator foods management system.

III. OPEN SOURCE OCR COMPARISON

Although we began this project with the assumption that utilizing an open source OCR would be helpful, we quickly realized there were numerous such OCRs, and that it would be necessary to evaluate each from the perspectives of accuracy, how many kinds of character types could be recognized, and operating system (OS) compatibility. Table I depicts comparison results for three open source OCRs [1]-[3].

TABLE I: COMPARISON RESULTS FOR THREE OPEN SOURCE OCRS

OCR	Applicable Character type	OS	Remarks
Tesseract-OCR [1]	Numeric, A-Z, a-z, Hiragana, Katakana, Kanji	Android, iOS, Windows**	Many documents, Frequent updates
Nhocr [2]	Numeric, A-Z, a-z, Hiragana, Katakana, Kanji	Linux, Solaris, Android**	Low recognition rates of Alphabet and so forth.
OCROPUS [3]	Numeric, A-Z, a-z	FreeBSD, Linux, Mac OS	

In the table above, we can see that Tesseract-OCR [1] and NHocr [2] can be used to recognize numerous character types including numerals, Roman alphabet, hiragana, katakana and kanji, and that all three OCRs can be implemented in multiple OSs. However, of the three, the Tesseract-OCR had the most supporting documents, so we were able to more thoroughly investigate its performance. Furthermore, since it is also updated frequently, its reliability is somewhat higher than that of other OCRs. For these reasons, we decided to employ the Tesseract-OCR for our system.

IV. OCR PERFORMANCE EVALUATION AND DISCUSSION

This section shows the experimental conditions and results of our Tesseract-OCR performance evaluation, after which we discuss some solutions to problems we encountered when implementing our expiration date recognition function.

Before the evaluation, we collected some food packages and categorized the indication forms of expiration date. Based on the result, we evaluated the performance and considered solution as follows:

A. Input Image Size

Table II shows the images used for this experiment. The left image has a small amount of blank space, while in the center and the right images, half and two-thirds of the image are blank space, respectively.

TABLE II: EXPERIMENTAL CONDITIONS AND RESULTS OF INPUT IMAGE SIZE (“消費期限” AND “製造所固有記号” MEAN EXPIRATION DATE AND PLANT CODE, RESPECTIVELY)

Blank space	Little	Half of image	Two thirds of image
Input image			
Results	Correctly recognized	Misrecognized	Misrecognized

In order to recognize characters correctly, we must first extract the area containing the written characters or exclude the unnecessary blank space.

B. Contrast

Table III depicts the images used for this experiment. We set the contrast levels at -40, 0, and +40%. All three images were recognized correctly. Based on these results, and after considering that most of the images input into the system would be taken indoors, we decided that it would not be necessary to fine-tune the contrast levels.

TABLE III: EXPERIMENTAL CONDITIONS AND CONTRAST RESULTS

Contrast	-40%	0%	+40%
Input image			
Results	Correctly recognized	Correctly recognized	Correctly recognized

C. Brightness

Table IV depicts the images used for brightness experiment. These were set at -40, 0, and +40%. All of three images are

recognized correctly. Based on these results, and after again considering that most of the images input into the system would be taken indoors, we decided that it would not be necessary to fine-tune the brightness levels.

TABLE IV: EXPERIMENTAL CONDITIONS AND BRIGHTNESS RESULTS

Brightness	-40%	0%	+40%
Input image			
Results	Correctly recognized	Correctly recognized	Correctly recognized

D. Fonts

In this experiment, we selected similar fonts to expiration date written on actual food packages. For Roman style characters, the Arial, Georgia, Verdana, and Times New Roman fonts were used, while for Japanese style characters, the MSP Gothic, MSP Mincho, and Meiryo fonts were evaluated. As shown in Fig. 2, the same characters were evaluated for each font. The results show that all of the Roman style fonts, as well as MSP Gothic, were recognized correctly. However, the figure also shows that MSP Mincho and Meiryo were misrecognized. Here, we noted that the stroke widths of those two fonts are thinner than those of the fonts that were recognized correctly.



Fig. 2. Misrecognized fonts. (“消費・賞味期限” means expiration and freshness date).

E. Color



Fig. 3. Input image example for color experiment. (In cases where the background was black, white characters were used.)

TABLE V: MISRECOGNIZED CHARACTERS IN COLOR EXPERIMENT

Background color	Black characters		White characters	
	Kanji	Numeric	Kanji	Numeric
Red	費賞	-	味期	-
Orange red	All	All	All	-
Orange yellow	All	-	賞味期	-
Yellow	-	-	賞味期限	-
Yellow green	All	-	味	-
Green	味期	-	味期	-
Blue green	賞味	1	味期	-
Green blue	All	6	味期	-
Blue	味	0	味期	-
Bluish purple	-	-	味期	-
Purple	賞味期限	15	味期	-
Reddish purple	消	8	味期	All
White	-	-	All	All
Black	All	All	-	-

Twelve colors selected from the Practical Color Coordinate System [4], as well as white and black, were used for

background color in our evaluations. However, for character colors, only black and white were evaluated. According to our investigation these color conditions seem to cover most cases of expiration date indication. All color experiments used same characters, which are shown in Fig. 3. Table V shows the experimental results.

Different experimental results were obtained for each color. Even if similar system colors were used, very different results were observed.

F. Dot Matrix Character

All of the dot matrix fonts we tested were misrecognized. Fig. 4 shows an example of an unrecognized font.



Fig. 4. Example of misrecognized dot matrix characters.

G. Distortion

Expiration dates written on polyethylene terephthalate (PET) and glass bottles were used for this experiment. Glass bottles with diameters less than 5 cm were misrecognized. Fig. 5 shows an example of characters written on a PET bottle. These were recognized correctly.

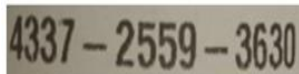


Fig. 5. Example of characters written on PET bottle (recognized correctly).

H. Reflection

Expiration dates directly written on metal or glass surfaces like cans and bottles were used. However, it was found that when even a small amount of light was reflected from the surface, the characters were misrecognized. Fig. 6 shows an example of characters written on a can with a reflective surface.



Fig. 6. Example of characters written on a can with a reflective surface (misrecognized).

I. Slanted Characters

All characters which are not taken horizontally by a camera were misrecognized.

J. Discussion

TABLE VI: EXPIRATION DATE RECOGNITION PROBLEMS AND SOLUTIONS

Problem	Proposed solution
(1) Images including wide non-character space are misrecognized	(a) Crop the image to include only the space containing characters
(2) All characters which are not taken horizontally by a camera are misrecognized	(b) Rotate characters horizontally
(3) Some background and character colors were misrecognized	(c) Convert background and character colors to white and black, respectively
(4) Dot matrix characters are misrecognized	(d) Connect dots
(5) Thin line fonts are misrecognized	(e) Thicken character lines
(6) Very distorted characters are misrecognized	
(7) Images including reflection are misrecognized	

Based on the knowledge obtained from the above-mentioned experiments, Table VI summarizes the problems and our proposed solutions.

To solve Problem (1) shown in Table VI, in which images that include excessive non-character space are misrecognized, we decided to crop the image to just the space in which written characters are present. In regards to Problem (2), in which characters not taken horizontally by a camera are misrecognized, it was determined that they should be rotated to a proper horizontal orientation. As for Problem (3), in which background/character color differences caused misrepresentation, we decided to convert the background and character colors into white and black, respectively. To solve Problem (4), in which dot matrix characters are misrecognized, it was necessary to connect the dots. For Problem (5), which resulted because thin font lines were misrecognized, it was determined that we should thicken character lines to solve the problem.

However, Problems (6) and (7) are currently unresolved, and we are now considering methods to address them.

V. DEVELOPED ANDROID APPLICATION

The selected OCR was designed to be installed in a server, as shown in Fig. 1. However, since we have not yet prepared a dedicated server, we developed an Android smartphone application to recognize expiration dates using Tesseract-OCR [1], Android Studio [5], and OpenCV [6].

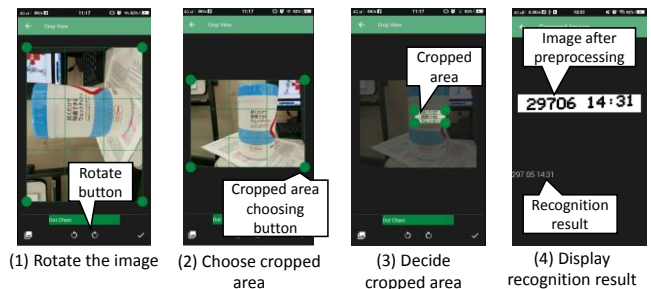


Fig. 7. Overview of our developed Android application.

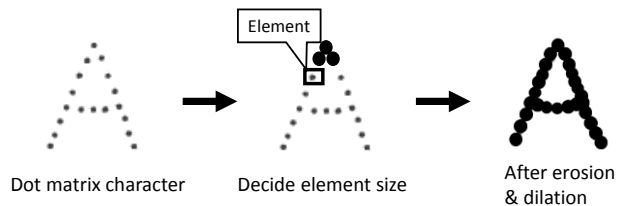


Fig. 8. Erosion and dilation procedure for dot matrix characters. (The element is a matrix that identifies the pixels of the image being processed and defines the neighborhood used in the processing of each pixel.)

The application, for which an overview is shown in Fig. 7, includes a number of functions used to implement the solutions given in Table VI (a-e). For example, it allows the user to crop and manually rotate images, as shown in Fig. 7(1-3). Additionally, in order to implement the solutions shown in Table VI (d,e), we employ erosion and dilation, as shown in Fig. 8. Here, element size is critical because it correlates to the character height. Accordingly, we first binarize the image and then measure the height from its horizontal density histogram, as shown in Fig. 9. Our

experimental results show that the most suitable value of α was 0.1. Finally, the application can output the result, as shown in Fig. 7(4).

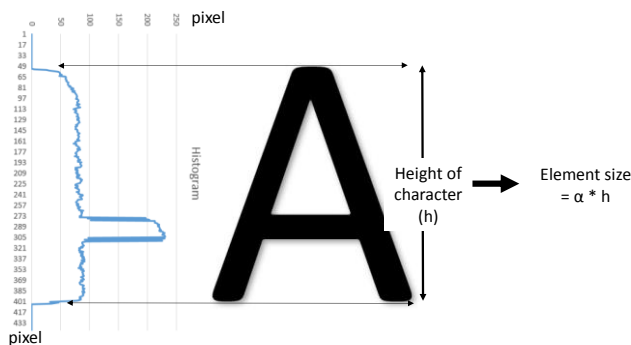


Fig. 9. Element size calculation method.

VI. RELATED RESEARCH

At its current stage of development, our application cannot automatically crop the image to exclude the blank space surrounding characters, nor can it rotate the characters horizontally. Accordingly, in the first part of this section we will review research related to these two techniques, and then discuss studies related to correctly recognizing dot matrix characters.

Studies related to extracting only areas with written characters are classified into two categories. The first are batch-based methods, while the second are connected component- or region-based methods. Batch-based methods scan the image interior and use machine learning to detect high probability areas [7], [8]. However, since this method cannot separate character-containing areas from the background, the images must first be divided into those two areas, thereby increasing computing costs.

In contrast, based on the assumption that pixels belonging to the same character will have similar characteristics, connected component-based methods classify pixels into regions by utilizing the characters' connected components. In general, an advantage of such methods is that the color graduation of the connected components do not depend on scale, direction, and/or font. It has been noted that the maximally stable extremal regions (MSER) method is particularly effective for this task [9], [10].

Several approaches have been proposed for rotating characters by a certain number of degrees [11]. These studies can also be divided into two categories. A typical procedure of the first category is to rotate the characters in the preprocessing stage and then perform character recognition [12]. In methods of the second category, character recognition is attempted first without rotation. Methods of the second category are further classified into two types. The first extracts robust features for rotation and recognition [13], while the second constructs invariant recognizers for rotation [14].

As for a dot matrix text, several methods have been proposed as ways to recognize such characters accurately. These methods are also classified into two groups: preprocessing- and training-based methods. The first group utilizes several conventional preprocessing methods such as

blob-connection and slant/rotation correction [15]. However, dot matrix characters in actual use vary significantly in terms of printing quality due to factors such as font pattern and dot size, which makes it very difficult to use the same process for every dot matrix font. The second group employs models such as neural networks as classifiers [16].

VII. CONCLUSION

The present study described a smartphone application designed to recognize printed expiration dates as part of a refrigerator management system for perishables. We began by evaluating the ability of the selected open source OCRs to recognize expiration dates in multiple environments, and then implemented an Android smartphone software application equipped with a number of preprocessing functions.

In the future, we plan to implement other functions into our application in order to solve problems related to reflection and distortion. We are also planning to implement object classification with deep learning in the near future.

REFERENCES

- [1] Tesseract-OCR. [Online]. Available: <https://www.slideshare.net/takmin/tesseract-oc>
- [2] NHocr. [Online]. Available: <https://ja.osdn.net/projects/nhocr/docs/about-140830/ja/19/about-140830.html>
- [3] OCRopus. [Online]. Available: <http://d.hatena.ne.jp/moriyoshi/20070412/1176377845>
- [4] PCCS (Practical Color Co-ordinate System). [Online]. Available: www.geocities.jp/k_lab_japan/nara/PCCS.pdf
- [5] Android Studio. [Online]. Available: <https://developer.android.com/studio/index.html>
- [6] OpenCV (Open Source Computer Vision Library). [Online]. Available: <https://opencv.org/>
- [7] X. Chen and A. Yuille, "Detecting and reading text in natural scenes," in *Proc. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 366-373, vol. 2, 2004.
- [8] K. Kim, K. Jung, and J. Kim, "Texture-base approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1631-1639, 2003.
- [9] H. Chen, S. Tsai, G. Schroth, D. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *Proc. IEEE International Conference on Image Processing*, pp. 2609-2612, 2011.
- [10] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3538-3545, 2012.
- [11] X. C. Yin, X. Yin, K. Huang, and H. W. Hao, "Robust text detection in natural scene images," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 970-983, 2014.
- [12] K. Suzuki, S. Araya, and R. Nakamura, "A copy-learning model for recognizing randomly rotated patterns and their angles," *Transactions of Information Processing Society of Japan*, vol. 35, no. 7, pp. 1311-1317, 1994.
- [13] S. Sankoda, T. Imura, H. Masuyama, Y. Sato, and S. Tamura, "Rotated-pattern normalization by neural network," *The Journal of the Institute of Image Information and Television Engineers*, vol. 52, no. 11, pp. 1713-1723, 1998.
- [14] U.V. Kulkarni, T. R. Sontakke, and G. D. Randale, "Fuzzy hyperline segment neural network for rotation invariant handwritten character recognition," in *Proc. the International Joint Conference on Neural Networks (IJCNN'01)*, pp. 2918-2923, vol. 4, 2001.
- [15] K. Mohammad and S. Agaian, "Practical recognition system for text printed on clear reflected material," *ISRN Machine Vision 2012*, 2012.
- [16] A. Namane, E. H. Soubari, and P. Meyrueis, "Degraded dot matrix character recognition using csm-based feature extraction," in *Proc. the 10th ACM symposium on Document engineering*, pp. 207-210, 2010.

Hiroaki Seya is a student that belongs to the professor Arai's office in Teikyo University. He is majoring in recognition technology there.



Kento Hosozawa was with Faculty of Science and Engineering, Teikyo University. He is now with Nissin Kogyo Co., Ltd.



Masayuki Arai is a professor in the Graduate School of Sciences and Engineering at Teikyo University. He received his B.E. degree from Tokyo University of Science in 1981 and Dr. Eng. degree from Utsunomiya University in 1995. His research interests include pattern recognition, natural language processing and information visualization. He is a member of the Information Processing Society of Japan and IEEE.

Tsukasa Maekawa was with the Graduate School of Sciences and Engineering at Teikyo University.



Tran Duy Linh is a PhD student at Graduate School of Science and Engineering, Teikyo University, Japan. He received the B.E. degree in information system from the Hanoi University of Science and Technology, Hanoi, Viet Nam in 2010 and received the M.S. degree in computer science from Ho Chi Minh City University of Technology, Ho Chi Minh City, Viet Nam, in 2015. His research interests include image and video processing. He is presently engaged in

research on object detection using deep learning. He is a member of the Information Processing Society of Japan.



Kozo Mizutani received his D.E. degrees from the Graduate School of Science and Engineering at Teikyo University in 2008. He worked as an assistant professor at Medical Information and Systems Research Center, Teikyo University from 2008. Since 2014, he has been working as a lecturer at the Department of Information and Electronic Engineering, Faculty of Science and Engineering, Teikyo University. He is a member of IEICE, IPSJ,

JSiSE and JAMI.



Ricky Hendra Wijaya was with Faculty of Science and Engineering, Teikyo University.

He is now with Faculty of Engineering, Department of Electrical Engineering, Universitas Katolik Indonesia Atma Jaya, Indonesia.