Proposal of a Method for Automatic Fill-in Fields Detection and for Labels Assignment to Generate Electronic Forms

Yuya Kimura*, Tetsuro Katayama*, Yoshihiro Kita†,
Hisaaki Yamaba*, Kentaro Aburada*, and Naonobu Okazaki*
** Department of Computer Science and Systems Engineering, Faculty of Engineering, University of Miyazaki*
1-1 Gakuen-kibanadai nishi, Miyazaki, 889-2192 Japan
†Department of Information Security, Faculty of Information Systems, Siebold Campus, University of Nagasaki
1-1-1 Manabino, Nagayo-cho, Nishi-Sonogi-gun, Nagasaki, 851-2195 Japan
E-mail: kimura@earth.cs.miyazaki-u.ac.jp, kat@cs.miyazaki-u.ac.jp, kita@sun.ac.jp,
yamaba@cs.miyazaki-u.ac.jp, aburada@cs.miyazaki-u.ac.jp, oka@cs.miyazaki-u.ac.jp

Abstract

The digitalization of forms is being promoted. One of the effective ways to manage contents filled in fields is using electronic forms. Several tools have been developed to generate them automatically. However, when you use a paper form, the layout of the original form may change, and it takes time to generate electronic one because it is necessary to place fill-in fields on an electronic form by dragging with a mouse. This paper proposes a method for automatic fill-in fields detection and labels assignment to reduce time required to place fill-in fields without changing the layout. The proposed method can reduce the time to place fill-in fields.

Keywords: image processing, electronic form, fill-in fields, area coordinates obtainment, label assignment.

1. Introduction

In April 2019, Electronic Books Maintenance Act was amended [1], and the digitalization of them is being promoted. On the other hand, about half of Japanese companies still use paper form for their management system [2]. Forms can be digitized by taking pictures with a scanner or camera. This make it easy, but it is required to see contents filled in fields on manual. Using electronic forms is one of the ways to manage them efficiently. Some tools have been developed to generate electronic forms automatically. However, when using these tools, it has the following two problems.

- The layout of original forms may change.
- It takes time to place fill-in fields because it requires placing fill-in fields on an electronic form by dragging with a mouse.

To solve them, this paper proposes a method for automatic fill-in fields detection and for labels assignment to generate electronic forms. There is a previous study to detect rectangular fill-in fields to recognize the category of form documents and to store them [3]. Based on it, our proposed method is performed image-based fill-in fields detection.

2. Proposed method

In this chapter, we present the structure and the behavior of the proposed method. Fig. 1 shows the structure of the proposed method. It consists of four parts: Area coordinates obtainment part, Texts information obtainment part, Labels assignment part, and Files output part. It takes as input a form image, and outputs two form images highlighted fill-in fields and a JSON file including fill-in fields coordinates and labels.

An image of a digitized document and an image of an electronic document are received as input. Electronic documents refer to documents created as digital information, such as Word or text files. Digital documents refer to documents that have been scanned from paper documents and saved in PDF or image format. The image of the digitized document is assumed to be an image taken with a smartphone. In this paper, the fill-in fields obtained coordinates with the proposed method are called area coordinates, and ones indicated rectangle are called rectangular areas, and ones indicated underline are called underlined areas.

2.1. Area coordinates obtainment part

Area coordinates obtainment part obtains area coordinates for rectangular areas and underlined areas. The coordinates of rectangular area acquire four xy coordinates as rectangular area coordinates for each vertex. The coordinates of underlined area acquire two xy coordinates as underlined area coordinates for both endpoints. To obtain area coordinates, image processing is performed using OpenCV. Furthermore, we use DeblurGANv2 [4] to remove blur in the image. DeblurGANv2 is a tool that applies generative adversarial network (GAN) to deblurring. The following is the flow of Area coordinates obtainment part behavior.

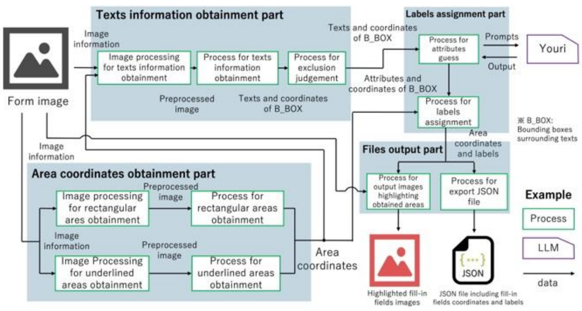


Fig. 1. The structure of the proposed method

(i) Process image to obtain rectangular area coordinates with imread, cvtColor, GaussianBlur, threshold, getStructuringElement, and dilate function of OpenCV.
(ii) Deblur the form image with DeblurGANv2.
(iii) Detect rectangle and obtain rectangular area coordinates with findContour function of OpenCV.
(iv) Process image to obtain underlined area coordinates with imread, cvtColor, and threshold function of OpenCV.
(v) Detect underline and underlined area coordinates with createFastLineDetector function of OpenCV Contrib, which is one of OpenCV's extra modules.

2.2. Texts information obtainment part

Texts information obtainment part obtains texts and the coordinates of the bounding box surrounding them as texts information using optical character recognition (OCR). Using OCR engine is Tesseract [5] with Japanese trained data and dictionary. In recognizing texts, the accuracy of texts recognition can be improved by binarizing the image. In this process, some pixels of fill-in fields affect the binarization for character recognition optimization. To solve this problem, the most frequent RGB values among all pixels in the image are obtained as the background color's RGB values. The following is the flow of Texts information obtainment part behavior.

(i) Obtain the area coordinates from Area coordinates obtainment part.
(ii) Load the image with imread function of OpenCV.
(iii) Deblur the loaded image with DeblurGANv2.
(iv) Obtain the most frequent RGB values with histogram function of Numpy.
(v) Fill the area with the background color referenced from the obtained area coordinates.
(vi) Generate an image after process (v).
(vii) Obtain texts information with OCR from the output image of process (vi) with Tesseract.

2.3. Labels assignment part

Labels assignment part assigns labels to obtained area coordinates. By assigning labels, it is possible to add information required for validation. In this study, one is selected from three data types: "date", "number", or "string". It uses Youri [6], Japanese large language model (LLM) to predict data types as attributes of texts to be filled in from the recognized texts. The data type is linked as the label by referencing the coordinates of the bounding box surrounding the texts. The following is the flow of Labels assignment part.

(i) Input prompt including obtained texts to guess the data type of them into Youri.
(ii) Obtain the answers from Youri as attributes.
(iii) Initialize labels for all area coordinates as string.
(iv) Assign labels of rectangular areas to attributes of the bounding box if the x, y coordinates of the bounding box center are greater than ones of the rectangular region center.
(v) Assign labels of underlined areas to attributes of the bounding box if the x, y coordinates of the bounding box center are greater than ones of the rectangular region center.

2.4. Files output part

Files output part outputs two form images and one JSON file. One of the form images highlights rectangular areas, and the other highlights underlined areas. The JSON file includes area coordinates and labels of them. The following is the JSON file composition.

- A unique ID for each area coordinate.
- The label assigned to each area coordinate.
- The object indicating each area coordinate.

The following is the flow of File output part behavior.

(i) Obtain area coordinates and labels of them from Labels assignment part.
(ii) Generate two form images A and B with copy function of Python.
(iii) Generate three random integer values and determine a random RGB color from each color space with randint function of Numpy.
(iv) For form image A, draw rectangles colored the RGB color generated in process (iii) by referencing the rectangular area coordinates in the form image with drawContours function of OpenCV.
(v) For form image B, draw green lines by referencing the underlined area coordinates in the form image with line function of OpenCV.
(vi) Draw the values of id key and label key at the top left of each area coordinate in the form images with putText function of OpenCV.
(vii) Save form image A as highlighting rectangular area image and form image B as highlighting underlined area image with imwrite function of OpenCV.

3. Application Example

In this chapter, we confirm that the proposed method works correctly by using an implemented prototype. A part of an example of the output of JSON file is shown in List. 1, and a part of an example of a highlighted rectangular areas image is shown in Fig. 2.

From List. 1, The label is “string” for the rectangular area with id 4, and the label is “number” for the one with id 5. From Fig. 2, The fourth rectangular area is a fill-in field for "品名", and the fifth rectangular area is a fill-in field for "数量". Here, “品名” is the Japanese word for the item name, and “数量” is the Japanese word for the quantity. For each label in each rectangular area is “string” and “number”. Therefore, we have confirmed they are the correct labels for “品名” and “数量”. And, we have confirmed their area coordinates are correct.

List. 1. A part of an example of output JSON file


```
{
  "id": 4,
  "label": "string",
  "coords": {
    "top_left": {
      "x": 275,
      "y": 817
    },
    "bottom_left": {
      "x": 275,
      "y": 903
    },
    "bottom_right": {
      "x": 1008,
      "y": 903
    },
    "top_right": {
      "x": 1008,
      "y": 817
    }
  }
},
{
  "id": 5,
  "label": "number",
  "coords": {
    "top_left": {
      "x": 1016,
      "y": 817
    },
    "bottom_left": {
      "x": 1016,
      "y": 903
    },
    "bottom_right": {
      "x": 1308,
      "y": 903
    },
    "top_right": {
      "x": 1308,
      "y": 817
    }
  }
},
```


Fig. 2. A part of an example of a highlighted rectangular areas image

4. Discussion

In this chapter, to evaluate the usefulness of the proposed method, we experiment with six students of University of Miyazaki. For the experiment, we used a GUI tool to generate electronic forms.

There are three measuring times: execution time, placement time, and total time. The following is the meaning of each measurement time.

- Execution Time: the time taken by the program from the start to the end of its execution.
- Placement Time: the time to place all fill-in fields on the electronic form with the GUI tool.
- Total Time: sum of execution time and placement time.

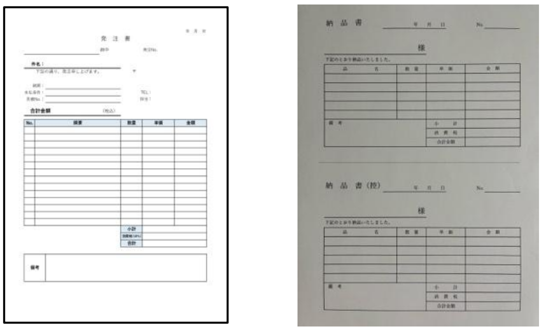
The steps of the experiment regarding evaluation placing fill-in fields with the GUI tool on an electronic form are shown below.

(i) The experimenter prepares two form images: a form image of an electronic document as Form image A, and a form image of a digitized document as Form image B.
(ii) The experimenter instructs the participants to place the fill-in fields in the electronic form with the GUI tool and starts measuring the placement time.
(iii) The participants place the fill-in fields as instructed

Fig. 3 shows the two form images used in the experiment. There are 86 fill-in fields to place in Form image A, and 54 ones to place in Form image B.

4.1. Evaluation on accuracy of area coordinates

Table 1 shows the precision rate and recall rate about area coordinates. We consider the reason why the recall rate of A is lower than that of B is because there are six filled-in fields to be filled in that is not indicated by either the rectangle or the underline, and it cannot detect them.



a. Form image A b. Form image B

Fig. 3. Form images used in the experiment

Table 1. The precision rate and recall rate about area coordinates obtainment (percent)

Precision Rate		Recall Rate	
Form image A	Form image B	Form image A	Form image B
91.49	81.82	93.48	100.00

4.2. Evaluation on accuracy of assigned labels

Table 2 shows the precision rate and recall rate about assigned labels. The area coordinates which have correct labels is correct too about the coordinates. We considered the reason why two rates in Form image A are less than that of Form image B is that a few texts are not recognized, so an attribute of other texts are assigned as the label incorrectly.

4.3. Evaluation on time to place fill-in fields with a GUI tool

There are two cases below for participants to place fill-in fields in the electronic form.

- Case A: using only the GUI tool to Form image A, and using it with proposed method to Form image B.
- Case B: using only the GUI tool to Form image B, and using it with proposed method to Form image A.

Table 3 shows the average of each measurement time for each form image. From Table 3, it reduced 3 minutes 54.5 seconds (about 31.4 %) of Form image A, and reduced 2 minutes 16.4 seconds (about 27.0 %) of Form image B. It shows that proposed method can reduce the time to place fill-in fields in an electronic form.

Table 2. The precision rate and recall rate about assigned labels (percent)

Precision Rate		Recall Rate	
Form image A	Form image B	Form image A	Form image B
72.22	73.13	81.25	90.74

Table 3. The measurement times it took the participants to place fill-in fields in electronic form

Average Measurement time	Form image A		Form image B	
	Case A	Case B	Case A	Case B
Execution	-	2:54.3	2:28.4	-
Placement	12:26.0	5:37.2	3:40.6	8:25.4
Total	12:26.0	**8:31.5**	**6:09.0**	8:25.4

5. Conclusion

This paper has reduced the time to place fill-in fields on an electronic form to generate it without changing the layout. We experimented with two form images. As a result, the proposed method can reduce the time by 31.4% and 27.0 %. Therefore, the proposed method has solved the two problems.

The future works are as follows.

- Improving recognition accuracy of fill-in fields and texts when using a colored form.
- Judging between filled-in fields themselves and fields indicating the contents to be filled in.
- Dealing with various shooting environment.

References

1. National tax agency, Special website for the Electronic Bookkeeping System (in Japanese) (Accessed 2024-12-13)
2. Keyman's Net, Status of digitalization of forms (2022) / Part 1 (in Japanese) (Accessed 2024-12-13)
3. Kuo-Chin Fan, Yuan-Kai Wang and Mei-Lin Chang, "Form document identification using line structure based features," Proceedings of Sixth International Conference on Document Analysis and Recognition, Seattle, WA, USA, 2001, pp. 704-708
4. O. Kupyn, T. Martyniuk, J. Wu and Z. Wang, "DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 8877-8886
5. GitHub, Tesseract, *tesseract*, https://github.com/tesseract-ocr/tesseract (Accessed 2024-12-13)
6. rinna, Rinna releases "Youri 7B", a Japanese language pre-learning model for Llama 2 (in Japanese) (Accessed 2024-12-13)

Authors Introduction

Yuya Kimura

Yuya Kimura received the Bachelor's degree in engineering (computer science and systems engineering) from the University of Miyazaki, Japan in 2024. He is currently a Master's student in Graduate School of Engineering at the University of Miyazaki, Japan. His research interests image processing.

Tetsuro Katayama

Tetsuro Katayama received a Ph.D. degree in engineering from Kyushu University, Fukuoka, Japan, in 1996. From 1996 to 2000, he has been a Research Associate at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. Since 2000 he has been an Associate Professor at the Faculty of Engineering, Miyazaki University, Japan. He is currently a Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include software testing and quality. He is a member of the IPSJ, IEICE, and JSSST.

Yoshihiro Kita

Yoshihiro Kita received a Ph.D. degree in systems engineering from the University of Miyazaki, Japan, in 2011. He is currently an Associate Professor with the Faculty of Information Systems, University of Nagasaki, Japan. His research interests include software testing and biometrics authentication.

Hisaaki Yamaba

Hisaaki Yamaba received the B.S. and M.S. degrees in chemical engineering from the Tokyo Institute of Technology, Japan, in 1988 and 1990, respectively, and the Ph D. degree in systems engineering from the University of Miyazaki, Japan in 2011. He is currently an Assistant Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include network security and user authentication. He is a member of SICE and SCEJ.

Kentaro Aburada

Kentaro Aburada received the B.S., M.S, and Ph.D. degrees in computer science and system engineering from the University of Miyazaki, Japan, in 2003, 2005, and 2009, respectively. He is currently an Associate Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include computer networks and security. He is a member of IPSJ and IEICE.

Naonobu Okazaki

Naonobu Okazaki received his B.S, M.S., and Ph.D. degrees in electrical and communication engineering from Tohoku University, Japan, in 1986, 1988 and 1992, respectively. He joined the Information Technology Research and Development Center, Mitsubishi Electric Corporation in 1991. He is currently a Professor with the Faculty of Engineering, University of Miyazaki since 2002. His research interests include mobile network and network security. He is a member of IPSJ, IEICE and IEEE.