Journal of Robotics, Networking and Artificial Life
Vol. 11(3); March (2026), pp. 220-225
ISSN (Online): 2352-6386; ISSN (Print): 2405-9021
https://alife-robotics.org/jrnal.html

Research Article

Implementation of a Tool for Automatic Entry Fields Detection and for Labels Allocation to Generate Electronic Forms

Yuya Kimura[1], Nobuya Takahashi[1], Tetsuro Katayama[1*], Yoshihiro Kita[2], Hisaaki Yamaba[1], Kentaro Aburada[1], and Naonobu Okazaki[1]

[1]*Department of Computer Science and Systems Engineering, Faculty of Engineering, University of Miyazaki 1-1 Gakuen-kibanadai nishi, Miyazaki, 889-2192 Japan*
[2]*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*
Email: kimura@earth.cs.miyazaki-u.ac.jp, ntakahashi@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
[*]Corresponding Author

ARTICLE INFO

Article History

Received 03 December 2024
Accepted 25 January 2026

Keywords

Image processing
Electronic form
Entry fields
Region coordinates extraction
Label allocation

ABSTRACT

An effective approach to manage the content entered in forms is the use of electronic forms. However, if you use a paper form, it takes time to generate electronic forms. This paper has implemented a tool for automatic entry fields detection and for labels allocation to reduce the time required to place entry fields. As a result, we have verified that the implemented tool is useful to reduce the time to place entry fields.

1. Introduction

In April 2019, Electronic Books Maintenance Act was revised in Japan [1], and the digitalization of forms is being promoted. Paper forms can be converted into digital data by scanning or photographing them with a scanner or camera. This makes it easy, but it is required to see content visually in entry fields. An effective approach to manage the content entered in forms is the use of electronic forms. Some tools have been implemented to automatically generate electronic forms [2], [3]. However, if you use a paper form, it takes time to place entry fields because it requires placing entry fields on an electronic form by dragging with a mouse.

To solve this problem, this paper implements a tool for automatic entry fields detection and for labels allocation to generate electronic forms. Compared with previous studies that analyze form structures, the implemented tool requires no prior preparation, such as datasets. In addition, the implemented tool can automatically detect entry fields from images of both electronic documents, such as Word or text files created as digital information, and digitized documents, such as scanned paper documents saved in PDF or image formats.

The implemented tool receives form images as input that satisfy all the following conditions. In this tool, digitized documents are assumed to be form images captured with a smartphone camera.

- The image is an electronic document or a digitized document of a form.
- The background color of the form is white.
- The form is written in Japanese and in horizontal writing.
- The text is not handwritten.
- Entry fields are indicated by rectangles or underlines.

In addition, the implemented tool outputs the following two items:

- A JSON file that summarizes the acquired region coordinates, their corresponding labels, and unique IDs (numbers) for rectangular regions and underlined regions, respectively.
- A highlighted region image, in which the detected entry fields are visually emphasized by different colors, and the labels corresponding to each region coordinates are drawn as text.

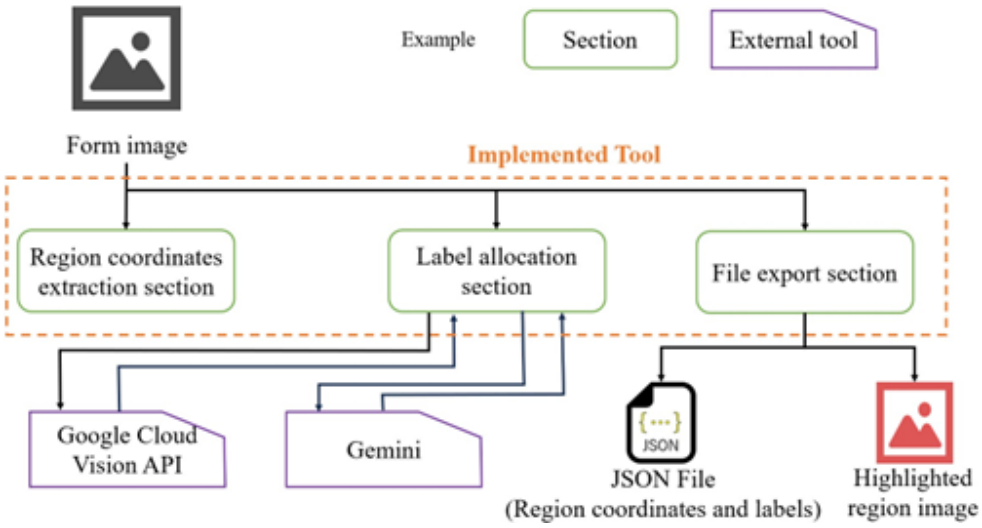

Figure 1 Structure of the implemented tool.

In this paper, the coordinates of detected entry fields are referred to as region coordinates, The regions indicated by rectangles are referred to as rectangular region coordinates, and those indicated by underlines are referred to as underlined region coordinates, respectively.

2. Previous Study

There are previous studies on form structure analysis that require no prior preparation. One of them is an image-processing-based approach [4], [5], [6].

As an example, Yuan et al. [6] proposed a method that recognizes table structures spanning multiple pages in form PDF files and outputs a CSV file containing the text within each cell along with their positional relationships. Their approach takes as input either PDF files or image files such as PNG and JPEG formats, and produces as output a CSV file containing the extracted text and a single image in which the recognized table structure is highlighted. While their method can recognize table structures across multiple pages, it does not support PDF files of digitized documents.

3. Implemented Tool

3.1. Functions

The implemented tool has the following two functions:

- Automatic region coordinates extraction function:
  This function receives a form image as input and extracts the coordinates of entry fields indicated by rectangles and underlines, as rectangular region coordinates and underlined region coordinates, respectively.
- Label allocation and result output function:
  This function receives a form image and region coordinates, and estimates the data type of the contents in the entry fields as labels. The data type is classified into one of three categories: string, number, or date. It estimates the data type of each detected character as an attribute, and assigns the attribute as a label to the corresponding region coordinate. After label allocation, this function outputs a JSON file containing both the region coordinates and their assigned labels. It outputs a PNG image in which the obtained region coordinates and their corresponding labels are highlighted on the input form image. The highlighted image makes it easy to verify visually the generated JSON file.

3.2. Structure

Figure 1 shows structure of the implemented tool. It is composed of the three following processing sections:

- Region coordinates extraction section
- Label allocation section
- File export section

3.2.1. Region coordinates extraction section

This section detects entry fields in the form image and extracts their coordinates. The input is the form image, and the output is region coordinates. These coordinates are sent to the Label allocation section. For rectangular regions, the output includes the x,y coordinates of the top-left vertex, width, and height. For underlined regions, the output includes the x,y coordinates of the leftmost point and width.

The following steps are performed in this section:

(i) Perform image processing on the form image, such as blur removal and binarization.

Figure 2 An example of highlighted region image showing detected rectangular regions.

List 1 An example of a JSON file showing detected rectangular regions.


```
{
    "id": 55,
    "label": "string",
    "coords" {
        "x": 275,
        "y": 2955,
        "width": 1033,
        "height": 272,
    }
},
{
    "id": 61,
    "label": "number",
    "coords" {
        "x": 275,
        "y": 2955,
        "width": 1033,
        "height": 272,
    }
},
```


(ii) Detect rectangles in the form image using the findContours function of OpenCV [7] and obtain rectangular region coordinates.
(iii) Sort the rectangular region coordinates in ascending order based on the x and y coordinates of their top-left vertices.
(iv) Detect horizontal lines in the form image using the HoughLinesP function of OpenCV and obtain underlined region coordinates.
(v) Sort the underlined region coordinates in ascending order based on the x and y coordinates of their leftmost endpoints.

3.2.2. *Label allocation section*

This section assigns labels to the acquired coordinates of region coordinates. It receives as input the region coordinates received from the Region coordinates extraction section and outputs pairs of region coordinates and their corresponding labels. The output of this section is sent to the File export section.

The following steps are performed in this section:

(i) Extract characters and their positions from the form image using Google Cloud Vision API [8].
(ii) Estimate the data type of each extracted character (date, string, or number), which are candidates of labels, using Gemini [9], and then obtain the result as its attribute.
(iii) Calculate the coordinates of the center point of each bounding box surrounding the extracted characters.
(iv) Sort the extracted characters in ascending order based on the y coordinates of their center points, and then by the x coordinates.
(v) Allocate labels to region coordinates when the region coordinate is greater than the center-point coordinate of each character's bounding box.

The labeling process overwrites the label of a field if a label has already been assigned to it. This is to reflect the typical writing order in Japanese forms and to prevent labels from being assigned to different region coordinates.

3.2.3. *File export session*

This section exports the following two files:

- A JSON file containing region coordinates paired with their corresponding labels.
- Highlighted region image, which is a PNG image highlighting the extracted entry fields.

It receives as input the pairs of region coordinates and labels received from the Label allocation section and outputs a JSON file and a highlighted region image.

4. Application Example

By using an application example, we verify that the implemented tool works correctly.

Figure 2 shows a part of an example of a highlighted region image showing detected rectangular regions. List 1 shows a part of an example of the output of JSON file showing detected rectangular regions.

From Figure 2, The fifty-fifth rectangular region is an entry filed for "備考" (remarks), and the sixtieth-first rectangular region is an entry field for "合計金額" (total amount). From List 1, The label of rectangular region with id 55 is "string", and the label of rectangular region with id 61 is "number". For each label in each rectangular region is "string" and "number". Therefore, we have confirmed they are the correct labels for "備考" and "合計金額". And, we have confirmed the region coordinates are correct.

a. Form image A

b. Form image B

Figure 3 Form images used in the experiment.

5. Evaluation

We conduct an experiment to evaluate the usefulness of the implemented tool. In addition, it is compared with previous studies.

5.1. Evaluation experiment

The implemented tool is applied to an existing electronic form generate tool, and an evaluation experiment is conducted. The electronic form making tool to which it is applied is a web application that supports generating electronic forms by placing entry fields using a GUI tool on a background image of the form. By applying the implemented tool, the tool automatically places the entry fields by referring to the outputted JSON file.

Figure 3 shows the two form images used in the experiment. Form image A is an image of an electronic document, and Form image B is an image of a digitized document. The numbers of entry fields in them are 77 for Form image A and 49 for Form image B.

We experiment with six students of University of Miyazaki. The following are the three measurement times used in the experiment:

- Execution Time: the time taken by the program from the start to the end of its execution.
- Placement Time: the time to place all entry fields on the electronic form correctly.
- Total Time: sum of Execution Time and Placement Time.

Table 1 shows the average measurement times for each form image. As shown in Table 1, the time was reduced by approximately 4 minutes and 25 seconds (about 48.2%) for Form image A, and by approximately 1 minute and 47 seconds (about 27.6%) for Form image B. These results indicate that the implemented tool can reduce the time to place entry fields in an electronic form.

Table 1 The average of measurement times it took the subjects to place entry fields in electronic forms. (min:sec)

Average measurement time	Form image A		Form image B	
	GUI only	With tool	GUI only	With tool
Execution	--	0:08.5	--	0:10.5
Placement	9:09.4	4:36.2	6:26.9	4:29.4
Total	9:09.4	4:44.7	6:26.9	4:39.9

5.2. Comparison with previous studies

Existing tools to generate electronic forms assist users in making them efficiently [2], [3]. However, when images of forms are used as input of them, they cannot detect entry fields, and they have to position the entry fields manually on the electronic forms by dragging them with a mouse. The implemented tool can detect entry fields and obtain the coordinates of them automatically.

We compare the implemented tool with the method proposed by Yuan et al. [6], which also uses an image-processing-based approach. For both Form image A and Form image B, we compare accuracy.

For accuracy, precision and recall are measured and compared to evaluate the correctness of both the detected region positions and the assigned labels. The equations for calculating precision and recall are shown below.

$$Precision = \frac{Entry\ Fields\ correctly\ detected}{Total\ Region\ Coordinates} \quad (1)$$

$$Recall = \frac{Entry\ Fields\ correctly\ detected}{Total\ Entry\ Fields} \quad (2)$$

Table 2 shows the results of comparing the accuracy of the method proposed by Yuan et al. and the implemented tool for Form image A and Form image B. Since the method of Yuan et al. does not include Label allocation function, the precision and recall of the implemented tool in Table 2 are shown in the form of "evaluation metrics that do not consider label errors (evaluation metrics that consider label errors)".

From Table 2, for Form image A, both the precision and recall of the implemented tool are higher compared to those of the method of Yuan et al. The reason why the recall of their method is lower than that of the implemented tool is that their method detects only rectangles and cannot detect underlines.

Furthermore, for Form image B, their method failed to detect any entry fields. We infer that this is because their method does not support form images of digitized documents as input.

Table 2 Accuracy comparison between the method of Yuan et al. and the implemented tool.

Evaluation metrics	Form image A		Form image B	
	Yuan et al.	Implemented tool	Yuan et al.	Implemented tool
Precision	81.48	87.80(87.80)	Not detected	84.00(66.00)
Recall	85.71	93.61(93.51)	Not detected	83.67(67.35)

From the above, the implemented tool is better than the method proposed by Yuan et al. in terms of automatically detecting entry fields in form images.

6. Conclusion

This paper has implemented a tool for automatic entry fields detection and for labels allocation to generate electronic forms. To verify the accuracy of the region coordinates and labels produced by the implemented tool, the precision and recall of both the implemented tool and the method of the previous study were evaluated. From the results of evaluation experiment, the implemented tool can reduce the time to place entry fields in an electronic form.

Therefore, the implemented tool is useful to reduce the time to place entry fields.

The future issues include the following.

- Improving the recognition accuracy of entry fields when processing colored form images.
- Distinguishing between entry fields and fields indicating the input content.
- Associating extracted entry fields with their corresponding input content.
- Expanding the evaluation with more diverse form images and more participants.
- Including additional comparison methods or baseline techniques for more reliable evaluation.

References

1. National tax agency, Special website for the Electronic Bookkeeping System. (in Japanese) (Accessed 2025-11-06)
2. System Integrator Corp., i-Reporter: Digital Form Solution for Converting Paper Forms into Electronic. (in Japanese) (Accessed 2025-11-06)
3. Infotec Inc., Create!Form: Electronic Form Generation Tool. (in Japanese) (Accessed 2025-11-06)
4. 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, pp. 704-708, 2001.
5. Manjunath, Akanksh and Nayak, Manjunath and Nishith, Santhanam and Pandit, Satish and Sunkad, Shreyas and Deenadhayalan, Pratiba and Gangadhara, Shobha, "Automated invoice data extraction using image processing", IAES International Journal of Artificial Intelligence (IJ-AI), Vol. 12, No. 2, pp. 514-521, 2023.
6. J. Yuan, H. Li, M. Wang, R. Liu, C. Li and B. Wang, "An OpenCV-based Framework for Table Information Extraction ", 2020 IEEE International Conference on Knowledge Graph (ICKG), pp. 621-628, 2020.
7. Bradski, G., The OpenCV Library, Dr. Dobb's Journal of Software Tools, 2008.
8. Google Cloud, Google Cloud Vision API, Google Cloud. (Accessed 2025-11-06)
9. Team Gemini: Gemini: A Family of Highly Capable Multimodal Models, 2023.

Authors Introduction

Mr. 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.

Dr. Nobuya Takahashi

He received B.S. and M.S. degrees in engineering from the University of Miyazaki, Japan, in 1994 and 1996, respectively, and Ph.D. degree in engineering from Kumamoto University, Japan, in 2006. He is currently an Assistant Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include linear control and machine learning. He is a member of SICE and JSME.

Prof. 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.

Prof. 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.

Prof. 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 Associate 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.

Prof. 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 a 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

Prof. 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.