Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 59 (2015) 520 – 529

Procedia Computer Science

International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

An Examination of Character Recognition on ID card using Template Matching Approach

Michael Ryan, Novita Hanafiah*

Bina Nusantara University, Jl. K.H Syahdan No.9, DKI Jakarta, 11480, Indonesia

Abstract

Identification card (ID cards) becomes the main reference in obtaining information of a citizen. Some business sectors require the information contained in the ID card to perform the registration process. In general, the registration process is still using a form to be filled in accordance with the data on the ID card, which will then be converted into digital data by means of retyping the information. The purpose of this research is to create a character recognition system on the ID card where the character recognition process included into four stages: pre-processing, text-area extraction, segmentation and recognition. The experiment includes some tests of greyscale, binarization and segmentation algorithm, as well as the combination of those algorithms. Text area extractor showed satisfactory results of identifying text-area on the ID card, which can scope all the entire area that consist of text. In the segmentation stage, approximately 93% of character can be cut off correctly. The actual character will be mapped to the template character using two algorithms where the division grid of each of them is different. Nevertheless, the recognition process of applying the template method matching still needs to be improved back.

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Keywords: ID card; character; recognition; OCR; template;matching;

* Corresponding author. Tel.:+62-858-837-877-71.
E-mail address: nhanafiah@binus.edu

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)
doi:10.1016/j.procs.2015.07.534

1. Introduction

Technology development is growing rapidly in the last decade, and having an impact on the use of gadgets in the daily activities. As sophisticated technology is utilized, human needs are also increasingly changing. Manual work will be minimized and changed as much as possible by applying a computer. In various business areas, especially those with the registration process, customer data will be inputted into database manually, which means the customer fill the form and the staff will enter the data written on the paper into computer. One of the most important data used in Indonesia is Identity Card. In almost every aspect, customer will be required to shown the ID card and those information which will be processed. Nowadays, a new fashion is developed to extract that information without asking the customer or staff to type the data one by one into the computer.

Instead of manually typing the data, a system can be proposed to extract the information by giving the ID card image to be processed and produced data as the result. The computer is required to be able to detect the text in the ID card and recognize the text. This approach is known as optical character recognition (OCR). An OCR application built in Android mobile platform[1] which can recognize a text from a photograph and process the text to produce a voice (text-to-speech). The text recognition leverages Tesseract which is an open source OCR engine[2], meanwhile Google's language translation service and Android system's text-to-speech synthesizer is leveraged to have the text read aloud.

Several methods and the algorithms to detect text in a video or image are reviewed, as well as the performance evaluation[3]. Texts in a video image can be classified into caption text, scene text, etc. It is possible to use a connected component-based approach to detect a text in a color image. For the edge detection, the most caption or text in the scene text is designed to be read easily, therefore, resulting in strong edges at the boundaries of text and background. Sushma and Padmaja use a contour-based robust algorithm to detect text in color images[4]. The method extracts the character regions to make a readable image for OCR. The contour gradient is applied by studying the pixels of text contour which usually have a high contrast to their neighbor pixels. Each candidate text region is verified with texture features. Binarization is also applied in the method before the recognition part.

OCR system typically consists of some phases, which are detection phase, localization, tracking, extraction and enhancement, and recognition (OCR)[3]. Text tracking is performed to reduce the processing time for text localization to maintain the integrity of position across adjacent frames, meanwhile enhancement of the extracted text components is required because the text region usually has low-resolution and is prone to noise.

In camera based analysis of text and documents survey[5], there are some challenges of capturing a picture from a camera, such as low resolution, uneven lighting, perspective distortion, nonplanar surfaces, wide-angle-lens distortion, complex backgrounds, zooming and focusing, moving objects, intensity and color quantization, sensor noise, compression, and lightweight algorithm. There are two terms used to express the text appearances in an image, which are caption text and scene text. Caption text usually has the strong contrast background and the fixed font size and position, for example a subtitle in a movie. The producer wants those texts to stand out clearly. On the other hand, scene text is harder to localize, because the text appearances is arbitrarily in the images. Binarization is applied before the text can be recognized. Adaptive thresholding method is more appropriate to handle the camera-captured image due to the lighting variation. A number of adaptive thresholding algorithms are generated from NiBlack's method. Liang et al said in another survey concludes that the most effective algorithm to binarize general image is using NiBlack's method[6]. It is also known that NiBlack's is effective to extract text[7,8].

Khurshidet al.[9] makes a comparison of some binarization methods inspired by NiBlack algorithm for ancient document images. Experiment result shows NiBlack algorithm always identifies the test regions correctly as foreground, but tend to produce large amount of binarization noise in non-text regions. Sauvola algorithm improved the NiBlack algorithm by applying dynamic range for the threshold. Nevertheless, when the grey values of text and non-text are close to each other, the result decreases significantly. In addition, Wolf algorithm shows that the performance will reduce when there is a sharp change in background gray value across the image. Even a small noisy patch could significantly influence the range of gray value which caused calculating misleading binarization threshold.

Epshteinet al.[10] detects a text in a natural scene by applying Stroke Width Transform (SWT) method. The image data will be transformed from containing color value per pixel to containing the most likely stroke width. One feature that separates text from other elements of scene is its nearly constant stroke width. The paper presents how to compute the stroke width for each pixel, so that the method can also use to recover regions that are likely to contain text. Another approach to recognize scene text is introduced by Shi et al. in 2013[11]. The approach use part-based tree-structured to model each type of character to detect and recognize character at the same time. When the potential character has been detected, the CRF model will be constructed on the potential locations. In order to define the unary and pairwise function, character scores, spatial constraints and language model are used. At the end, the recognition result is produced by minimizing the cost function.

Mollahet al.[12] design an OCR system based on camera devices. They experimenting with business card images captured by phone camera and try to recognize the character. The text regions are binarized and segmented into lines and characters, and those characters are passed into recognition module. Template matching approach is used to identify the character and the experiment shows a good result which can achieve 92.74% of accuracy. In this paper, the experiment is done on ID card images capture by mobile phone. The experiment to extract the caption text from the image includes many kind of algorithm for greyscaling and binarization, such as NiBlack, Sauvola, Wolf and Kurshid, histogram for segmentation phase and template matching for recognizing the text.

2. Text Detection Mechanism

There are four general stages to identify and recognize the text on an ID card, which can be seen in Figure 1. Digital image captured by mobile phone or any camera goes to preprocessing step. Scaling and greyscaling are applied in the preprocessing phase, as well as binarization. Next, the black and white image is processed in the text area extraction. Detecting the appearances of text in the image by separating the area into two parts, text region and background region.The addition algorithm is used in the text area optimization, and the giving a result of each text in a line. Segmentation phase takes those lines to be segmented and produce each segmented text to be recognized by the character recognizer. Template matching algorithms are used to complete the recognition phase and give the recognized text as the output of the system.

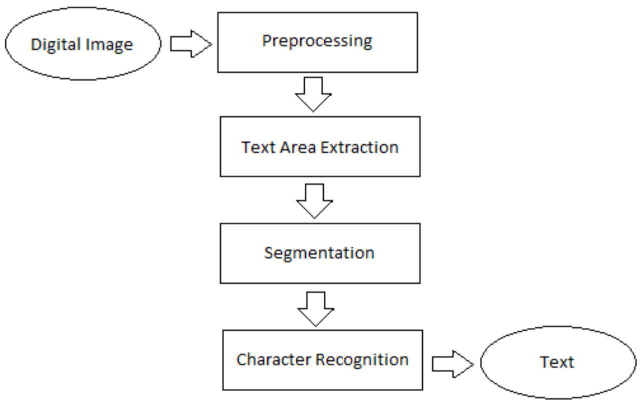

Fig. 1. Character Recognition on ID card scheme

2.1. Preprocessing

There are three smaller phases which are scaling, greyscaling, and binarization in this phase. At the scaling phase, images that have a length of more than 1024 is diminished in 1024 with the same ratio size. For example, a picture 3264x2448 (ratio 4: 3) entered andthat image is converted into a size of 1024x768 (the same ratio 4: 3). Scaling is used to speed up the process, because the larger the image, the more pixels that need to be processed.Image that have pass through scaling phase, is converted into gray image (greyscaling). In simple term, this phase change color image that have 3 color channels into gray image that only have 1 color channel. Greyscaling phase performed so that optimal binarization result will be obtained. There are 8 common algorithms used in this experiment, namely:

average (1), luminance (2), desaturation (3), maximum-decomposition (4), minimum-decomposition(5), single-channel colorred(6), single color channel green(7), single blue color channel(8).

$$f(x,y) = (R(x,y) + G(x,y) + B(x,y))/3 \quad (1)$$

$$f(x,y) = 0.299 \times R(x,y) + 0.587 \times G(x,y) + 0.114 \times B(x,y) \quad (2)$$

$$f(x,y) = \frac{\max(R(x,y) + G(x,y) + B(x,y)) + \min(R(x,y) + G(x,y) + B(x,y))}{2} \quad (3)$$

$$f(x,y) = \max(R(x,y) + G(x,y) + B(x,y)) \quad (4)$$

$$f(x,y) = \min(R(x,y) + G(x,y) + B(x,y) \quad (5)$$

$$f(x,y) = R(x,y) \quad (6)$$

$$f(x,y) = G(x,y) \quad (7)$$

$$f(x,y) = B(x,y) \quad (8)$$

The final phase in the preprocessing is binarization. The greyscaled image is transformed into a black and white image or binary image. Binarization divides image into 2 kinds of pixels (text pixel and background pixel). The division is done by finding the threshold value (t), which utilizes NiBlack, Sauvola, Wolf and Khurshid[9] in the experiment.

2.2. Text area extraction

In this phase image will be divided into 2 groups (text area and background area). First, image is divided into smaller parts (4x4 pixels). Next, the area that has between 6 to 93% of black pixel will be considered as text area. Then, all adjacent text area will be combined using the algorithm recursively 4 directions. Figure 2 shows the result of text are extraction phase on an ID card where the area inside the red-box is categorized as text region.

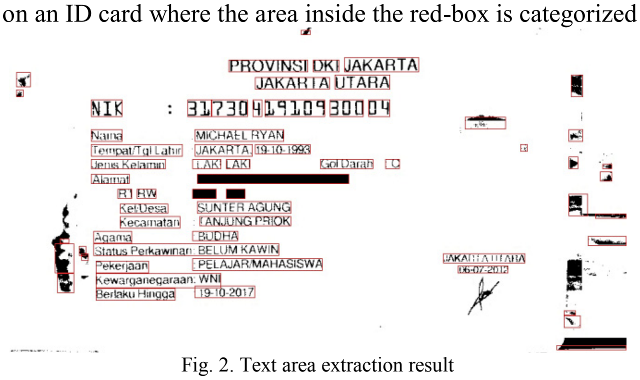

Fig. 2. Text area extraction result

2.3. Segmentation

Every text area is divided into smaller image for each character it holds. The segmentation is done horizontally (to dividing line) and vertically (to dividing characters). The division uses a histogram, will look for lines that do not have any first black pixel. Lines that do not have any black pixels, valued 1, are considered as a place to cut. The sample of segmentation result can be seen in figure 3.

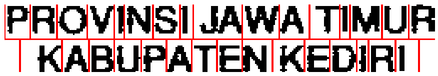

Fig. 3. Text area extraction result

2.4. Character Recognition

The output image from segmentation phase goes to character recognition phase that consist of several steps. At the beginning, the image is normalized by changing the image into standard size (the same size as the template size). Afterwards, thinning is conducted applying Zhange-suen algorithm[13]. When the thinning process has finished, the image is ready to be recognized. The normalization and thinning result is shown in figure 4.

(a) (b)

Fig 4. (a) Character normalization result, (b) Thinning result from (a).

There are two template matching algorithms used which are 3x3-algorithm and pixel-by-pixel algorithm. The idea of 3x3 algorithm is giving a template image (image of every character in a good position) for the system and divide that image into 9 squares (3x3 grids), afterwards code every squares based on the stroke type. We give an identification for each stroke types and create our pattern of recognizing the character. Several types of encoding used to deviate the pattern of a letter are:

- If the box has a pixel which only has one neighbor (tip pixel) is encoded as (1)
- If the black pixel is smaller than 3 then the box is considered as an empty box and coded as (0)
- If the box has a pixel that has four or more neighbors then the box is coded as (4)
- If the box is composed of neighboring pixels with 2 and 3 (usually the arch), then the box is coded as (3)
- If the box is only composed of pixels with 2 neighbors, then the box is coded as (2)

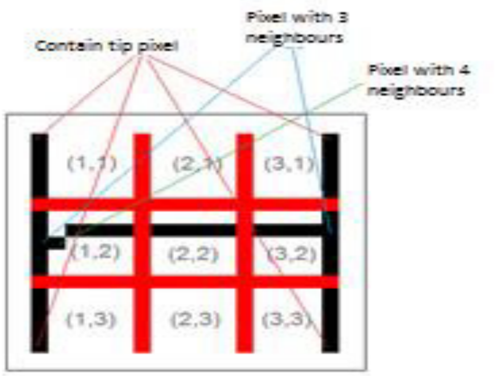

Fig 5. Example of encoded image using *3x3* algorithm

In figure 5 shows the example of encoded image produced from segmentation phase. By using the 3x3 algorithm the letter code is 101 323 101.

- Box (1.1), (3.1), (1.3), (3.3) has a tip pixel, it is coded as (1).
- Box (2.1) and (2.3) do not have a black pixel, it is coded as (0).
- Box (1.2) and (3.2) has a pixel consisting of pixels with neighboring 2 and 3, and 1 pixel with 4 neighbors, then encoded with (3).
- Box (2.2) consists of two neighboring pixels, then encoded as (2).

The value of each character (A-Z and other special characters) is stored in the database as our template to recognize the actual character. The image of an ID card that has passed the segmentation phase is need to be recognized, and we experiment with the 3x3 algorithm. That image is also divided into 3x3 grids and try to identify the pattern of each grid regarding to our pattern types above. Next, the encoded image is matched with the encoded data from the database in order to recognize the letter.

Unlike the 3x3 algorithm, the template of each letter is not divided into grids, but each pixel of the template image and the actual image produced from the segmentation phase is compared. Pixel-by-pixel algorithm match each pixel in the image and calculate the same number of pixels between the template and the tested images. Template with the highest similarity value will be considered as the answer.

3. Experiments Result

The algorithm's examination performed on preprocessing, text area extraction, and recognition phases, and at the end whole program is examined. The best combination of greyscale and binarization algorithms in the preprocessing phase is tried to be reached, with the best parameter on the greyscaling step in order to give the best input for binarization process. First, the combination of greyscale and binarization algorithms is tested on sample 1 image and the result is shown in table 1. It is shown that the algorithms give better results is Desaturation, Decomposition-max, Single Channel blue, Luminance.

(a) (b) (c) (d)

Fig 6. (a) sample 1, (b) sample 2, (c) sample 3, (d) sample 4.

The examination is continuing by test those combination algorithms for sample 2, 3 and 4 (Fig. 6). The experiment results obtained Decomposition-max algorithm and Wolf give the best results and 0:45 is the best *k* value. The window size used is 50x50 with the highest yield obtained was 93% (as result from Table 2). From the

320 total character on the ID card image, the system can segment about 296 characters correctly. A character that cuts correctly is a character that is not combined in one segment with another character or get separated in another segment (get the cut in the middle of the letter). The method used to cut the character into the segment is by finding a column of pixel that does not have any text pixel (black pixel). The example of segmentation results can be seen in fig 8. In fig 8(a), there should be 27 characters to be segmented but the system fails in one part which are "/" and "M", so it is counted that 25 characters are sliced correctly. In fig 8(b), there are some parts segmented incorrectly, such as "ca", "m", "n", and "T". It is calculated that the system can segment 16 characters correctly, out of 21 characters.

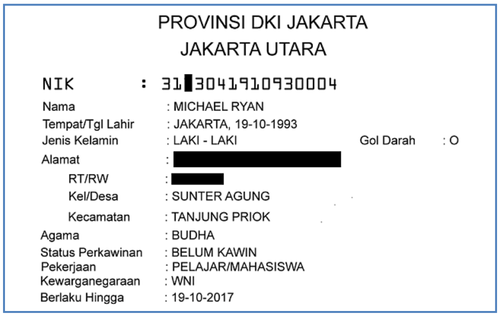

Fig. 7. Image used to test recognition algorithm

Table 1. Result of the combination of greyscale algorithm and binarization algorithm screening

No	*Greyscale Algoritm*	*Binarization algorithm*	Segmentation results			Noise category
			Character segmented correctly	Total character	%	
1	*Average*	*niblack*	0	0	0%	Many
2	*Average*	*Sauvola*	189	292	65%	Almost no noise
3	*Average*	*Wolf*	189	292	65%	Almost no noise
4	*Average*	*Khurshid*	203	292	70%	Almost no noise
5	*Lumanisce*	*niblack*	0	0	0%	Many
6	***Lumanisce***	***Sauvola***	**204**	**292**	**70%**	Few
7	***Lumanisce***	***Wolf***	**203**	**292**	**70%**	Few
8	***Lumanisce***	***Khurshid***	**210**	**292**	**72%**	Almost no noise
9	*Desaturation*	*niblack*	0	0	0%	Many
10	***Desaturation***	***Sauvola***	**202**	**292**	**69%**	Almost no noise
11	***Desaturation***	***Wolf***	**202**	**292**	**69%**	Almost no noise
12	***Desaturation***	***Khurshid***	**204**	**292**	**70%**	Almost no noise
13	*Decomp - Max*	*niblack*	0	0	0%	Many
14	***Decomp - Max***	***Sauvola***	**202**	**292**	**69%**	Almost no noise
15	***Decomp - Max***	***Wolf***	**202**	**292**	**69%**	Almost no noise
16	***Decomp - Max***	***Khurshid***	**218**	**292**	**75%**	Almost no noise
17	*Decomp - Min*	*niblack*	0	0	0%	Many
18	*Decomp - Min*	*Sauvola*	184	292	63%	Few
19	*Decomp - Min*	*Wolf*	184	292	63%	Few
20	*Decomp - Min*	*Khurshid*	194	292	66%	Almost no noise
21	*Single Channel - Red*	*niblack*	0	0	0%	Many
22	*Single Channel - Red*	*Sauvola*	184	292	63%	Almost no noise
23	*Single Channel - Red*	*Wolf*	184	292	63%	Almost no noise
24	*Single Channel - Red*	*Khurshid*	195	292	67%	Almost no noise
25	*Single Channel - Green*	*niblack*	0	0	0%	Many
26	*Single Channel - Green*	*Sauvola*	188	292	64%	Few
27	*Single Channel - Green*	*Wolf*	192	292	66%	Few
28	*Single Channel - Green*	*Khurshid*	203	292	70%	Almost no noise
29	*Single Channel - Blue*	*niblack*	0	0	0%	Many
30	***Single Channel - Blue***	***Sauvola***	**204**	**292**	**70%**	Few
31	***Single Channel - Blue***	***Wolf***	**204**	**292**	**70%**	Few

No	*Greyscale Algoritm*	*Binarization algorithm*	Segmentation results: Character segmented correctly	Segmentation results: Total character	Segmentation results: %	Noise category
32	***Single Channel - Blue***	***Khurshid***	**216**	**292**	**74%**	Few

(a) (b)

Fig 8. Segmentation result example

Afterwards, the examination in text area extraction stage looks for the size of the box that gives the best results in the division of the text area and background area. The results obtained are 4x4 box size and accuracy obtained is100% text area can be detected.

Table 2. Result of the parameter screening of greyscaling and binarization algorithms on Sample 2 image

Grayscale	Binarization	Window size	*k*	Result	Total characters	Percentage
Decomp - max	Wolf	20	0.5	276	320	86%
		50		284	320	89%
		60		280	320	88%
		40		277	320	87%
		50	0.55	266	320	83%
			0.45	294	320	**92%**
			0.4	296	320	**93%**
			0.35	285	320	89%
			0.3	276	320	86%
			0.25	259	320	81%
	Khurshid	20	0.5	275	320	86%
		50		279	320	87%
		60		282	320	88%
		70		283	320	88%
		80	0.55	267	320	83%
			0.45	290	320	**91%**
			0.4	288	320	**90%**
			0.35	288	320	90%

For the recognition algorithm performance, the system read the input images as shown in figure 7 and the result is shown in table 4 and 5. Algorithm pixel-by-pixel gives better results than the algorithm 3x3. This happens because the template that is used are noiseless that makes the template image and tested image easier to recognize using pixel-by-pixel. In order to test the whole system, start from preprocessing to recognizing process, sample 2 image is used as the input image and the result is presented in table 6 and 7.

Table 3. The sentence code representing the sentence on ID card

Sentences code	sentences	length
1	PROVINSI DKI JAKARTA	18
2	JAKARTA UTARA	12
3	31[X]3041910930004	20
4	Nama : Michael Ryan	16
5	Tempat/TglLahir : Jakarta, 19-10-1993	33
6	JenisKelamin : LAKI - LAKI	21
7	Gol. Darah : 0	10
8	Alamat : [….]	28
9	RT/RW : […..]	13
10	Kel/Desa : SUNTER AGUNG	20

11	Kecamatan : TANJUNG PRIOK	22
12	Agama : BUDHA	11
13	Status Perkawinan : BELUM KAWIN	27
14	Pekerjaan : PELAJAR/MAHASISWA	27
15	Kewarganegaraan : WNI	19
16	BerlakuHingga : 19-10-17	23
	Sum	320

Table 4. *3x3* algorithm to recognize specified input image

Sentence code	Result	Total letter	%
1	P??ViN?i?Kij?K?RT	12	61
2	j?K???U???	3	25
3	qIKi3??3Dh?91D?3oD0h	7	35
4	N?m??M?C4?EL??N	7	44
5	?m?????iL?h?rijA??T?1????n1???	10	30
6	???i?K?i?mi??LA?iLAK?	11	52
7	??i??rh?	2	20
8	?i????ix?M?Li?iBLO????N???	7	25
9	?T???DD??D1D	2	15
10	??i?????5uHTE??cuN?	5	25
11	????????nTHjuN?PRi??	8	36
12	??2m??Uoh?	2	18
13	??????P?????in?n?ELUM???N	9	33
14	P??????nPELAj?????hAg?5W	9	33
15	??N????n???r??niWHi	1	5
16	8??????h?n?????i1?i????	3	13
	Sum	97	30

Table5. *pixel-by-pixel* algorithm to recognize specified input image

Sentence code	Result	Total letter	%
1	PR0viUSi?KiJ?KART	14	78
2	JAKA?TUT?A	8	67
3	NTKi3TT3D4191D93D001	12	60
4	N?m?i?iO??ELR?N	7	44
5	Tmp?t?giL?hirij???Ti1?41011?9?	16	48
6	JcMi?Kci?miniL??iLA?i	11	52
7	?ciD?Lho	3	30
8	Ai???ti?Q??Li?i?LQ???4?Q??	7	25
9	RTiR?0?6i?10	7	54
10	?ciiDo???JNTERAGU??	8	40
11	??c?m?t??TNJ?NCFFi0?	9	41
12	A??????L?A	2	18
13	?t?tU?PcLk??iH?H?ELJ??A?N	10	37
14	??L??j?4??LAJ??i?A????sW	8	30
15	????L??u??????Ui?Ni	2	11
16	??Li?LJ?r????1?iJ0i?017	4	17
	Sum	128	39

In contrast to the result by using the test image in figure 7, *3x3* gives lightly better results than the pixel-by-pixel in sample 2. This happens because the algorithms *3x3* will take the form of general character and the noise does not impact too much compared on the pixel-by-pixel algorithm. Most characters cannot be read properly (symbolized by question '?' marks) because there are a lot of characters that is cut off and stuck with the other characters. Table 3 is the sentence appears in the ID card which should be recognized by the system. The information in sentence code 8 and 9 are not included for confidentiality.

Table 6. *3x3* algorithm to recognize sample 2 image

Sentence code	Result	Total Letter	%
1	??uV?N?i???j?N?PT?	5	28
2	jAK?Pr?UT?P?	5	42
3	?I?i317??4?????3???4	7	35
4	H????????h??L?Y?N	4	25
5	i???T??????LU?i?????PT??????i???	3	9
6	???1??K???????n?i?K?L?K?	5	24
7	?n?????????	0	0
8	????T?????nH??iP??L?????N???	4	14
9	?PT????????	1	8
10	x???????UHT?????4?	2	10
11	i????????rT??N?Uk??????	2	9
12	????????Dh?	2	18
13	???????u????2?i?????EL?jMK???N	6	22
14	????????n??LA?AP??h??????	5	19
15	?????i?????????ni?Ni	3	16
16	?????X?Hi?????????????	2	9
	Sum	56	18

Table 7. *pixel-by-pixel*algorithm to recognize sample 2 image

Sentence code	Result	Total letter	%
1	F?0V??Si???J??A?TA	7	39
2	jAKAPt?nT?P?	6	50
3	?T?i3T7Jn41pT?n3Enn4	7	35
4	???f4i?iCH?FL?Y??	5	31
5	ir?fii??fJ??jLi?J??????i1?T?iT??	2	6
6	J?L1i???fnL?iiMfii??r?L?	3	14
7	??iLir?tT?i	0	0
8	?i?f?1riLi???LinL?L?????N?1?	6	21
9	**Z**?????o??j?	0	0
10	??iL????U?T??????c	2	10
11	i??L?T?T?iiLANju??L?f0L	5	23
12	???L1??F?HA	1	9
13	**??i**itriJ??????iH?rf???i??K??iN	2	7
14	?????f??nFLL?j??i?HA????A	1	4
15	?????iu?????r???i?Ni	5	26
16	??ri?Lu?i?4???T?fo???z	3	13
	Sum	55	17

4. Conclusion

Detecting text on Indonesia ID card consists of some phases, which are pre-processing, extract the text regions, segmentation step and recognize the segmented area. We experience with some greyscaling algorithm and combine them with the binarization algorithm to find the best result as the input image of the next steps. Single Channel Blue algorithm has a good result due to the background color of Indonesian ID card is blue. Decomposition-Max and Wolf shows a good combination to produce better binarization image with k value is 0:45.Text area extractor showed satisfactory results of identifying text-area on the ID card, which can scope all the entire area that consist of text. In the segmentation stage, approximately 93% of character can be cut off correctly. The actual character will be mapped to the template character using two algorithms where the division grid of each of them is different. Nevertheless, the recognition process of applying the template method matching still needs to be improved back.

References

1. Mithe, Ravina, Supriya Indalkar, and Nilam Divekar. "Optical character recognition." *International Journal of Recent Technology and Engineering (IJRTE)* 2.1, 2013.
2. Smith, Ray. "An Overview of the Tesseract OCR Engine." *ICDAR*. Vol. 7. No. 1., 2007.
3. Jung, Keechul, Kwang In Kim, and Anil K. Jain. "Text information extraction in images and video: a survey." *Pattern recognition* 37.5 (2004): 977-997.
4. Sushma, J., and M. Padmaja. "Text detection in color images." *Intelligent Agent & Multi-Agent Systems, 2009. IAMA 2009. International Conference on*. IEEE, 2009.
5. Liang, Jian, David Doermann, and Huiping Li. "Camera-based analysis of text and documents: a survey." *International Journal of Document Analysis and Recognition (IJDAR)* 7.2-3 (2005): 84-104.
6. Trier, Øivind Due, and Torfinn Taxt. "Evaluation of binarization methods for document images." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.3, 1995, pp. 312-315.
7. Suen, H-M., and J-F. Wang. "Text string extraction from images of colour-printed documents." *IEE Proceedings-Vision, Image and Signal Processing*143.4 (1996): 210-216.
8. Wolf, Christian, J. Jolion, and Francoise Chassaing. "Text localization, enhancement and binarization in multimedia documents." *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Vol. 2. IEEE, 2002.
9. Khurshid, Khurram, et al. "Comparison of Niblack inspired Binarization methods for ancient documents." *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2009.
10. Epshtein, Boris, Eyal Ofek, and Yonatan Wexler. "Detecting text in natural scenes with stroke width transform." *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010.
11. Shi, Cunzhao, et al. "Scene text recognition using part-based tree-structured character detection." *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013.
12. Mollah, Ayatullah Faruk, et al. "Design of an optical character recognition system for camera-based handheld devices." *arXiv preprint arXiv:1109.3317*(2011).
13. Cheriet, M., Kharma, N., Liu, C.-L. & Suen, C. Y., 2007. *Character Recongition System: A Guide for Student and Practitioners.* New Jersey: John Wiley & Sons, Inc..