Computer Engineering and Applications Vol. 10, No. 3, October 2021

The Effectiveness of Image Preprocessing on Digital Handwritten Scripts Recognition with The Implementation of OCR Tesseract

Lily Rojabiyati Mursari*, Antoni Wibowo

Computer Science Department, Bina Nusantara University, Jakarta, 11480, Indonesia
**lily.mursari@binus.ac.id*

ABSTRACT

Optical Character Recognition (OCR) has been widely discussed in various topics in the rise of robotics, artificial intelligence and computer vision. OCR has become a solution in extracting characters from the image into machine-encoded text. This research aims to discuss character recognition from digital handwritten image. However, characters recognition problems using OCR has been more or less solved. OCR mainly implemented in reading characters from scanned of printed documents. In this research, image preprocessing including convert to grayscale, morphological operations and noise removal has been successfully boost the accuracy score of OCR performance. The average success outcome resulted to 79.26% in reading characters from the image.

Keywords: Optical Character Recognition, Digital Handwritten, Tesseract, Image Processing.

1. INTRODUCTION

Optical Character Recognition (OCR) is the process of reading text from printed or scanned-handwritten document. Nowadays, OCR already implemented in extracting filled forms, passports, ID cards recognition, license plate, etc. [1][2]. Not only in desktop application, by uploading or capturing the image, OCR has been widely implemented in smartphones. The quality of image-input become a major factors of recognition performance and affects the accuracy rate of OCR [3][4].

Even though the data in digitization format, there are challenging issues such as the image noise, different handwriting style, different type and font sizes, not aligned, etc. is become the major problems in recognizing characters. There are many OCR engine available and consistently updated newer version such as Tesseract, Google OCR, ABBYY FineReader, FreeOCR, etc., however, Tesseract stands out of all OCR engines [5]. Tesseract supports in Windows, Linux and MacOS. Tesseract provides 120+ models data trained language data files which already applied the new LSTM neural net-based engine. Tesseract also can process right-to-left text such as Arabic or Hebrew [6].

Tesseract is an open-source OCR engine that was developed by HP between 1984 and 1994. Tesseract was first started as a PhD research project. In the year 2005, Tesseract was released by Hewlett Packard and University of Nevada, Las Vegas. Now, partially Tesseract engine funded by Google and released under the Apache license, version 2.0. The lastest version, Tesseract 4.x is released in December, 2019. Tesseract engine implemented the key functional modules such as:

line and word finder, word recognition, static character classifier, linguistic analysis and adaptive classifier. However, Tesseract's output will result very poor quality if the images are not preprocessed [4][7][8].

Image processing is a form of signal processing with image as an input and the output may be the characterisics or features related with the input. There are two methods can be perform related to image processing namely, analogue image and digital image processing. Analogue image can be used for hardcopies document such as photographs, newspaper, etc. Digital image processing is a technique in manipulation digital images using computers. Digital image processing has three general phases techniques: preprocessing, enhancement and information extraction [9]. Preprocessing is one of important aspect in image processing. Image processing used in many fileds including recognition text inside images namely OCR as the topic of this research.

Historically, OCR has been implemented mainly in recognizing characters from scanned documents both printed document and handwritten document [4]. This research aims in exploration using digital handwritten image as the data input. Even in digital format, the data input contains pixel-noise, fuzzy, grainy which will affect the recognition performance. Moreover, this research developed the combination of preprocessing steps including convert into grayscale, morphological operations, and noise removal. Applying the preprocessing will improve the quality input for the OCR engine.

The scope of the research are deployed in web based application, Java programming language as the main algorithm including openCV as the image processing and Tesseract 4.x as the OCR engine. The dataset of the research is developed by providing canvas in the website, Figure 1. shows the ability of free-flow writing any characters and convert them into a .png format.

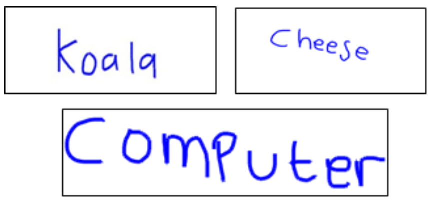


FIGURE 1. Digital Handwritten Image

This paper is organized as follows: Section 1 explained briefly about Optical Character Recognition (OCR). In the Section 2 we discussed a brief overview of related work and also we explained the proposed method of our research in Section 3, the experiments of the research are described and discussed in Section 4 before we conclude in Section 5.

2. RELATED WORK

Many researchers implemented various techniques of combination preprocessing for different types of character. For non-latin characters, in 2020, Anuradha et al. consists three main stages of recognizing Sinhala characters: preprocessing, recognition process and postprocessing. Their preprocessing proposed the techniques such as image binarization, noise removing, skew detection and correction, normalization and segmentation. The accuracy rate of reading Sinhala characters including the complex characters has achieved between range 80% - 90% [10]. In 2016, Qi Li et al. using five steps: trainned chinese characters data, dilation, erotion, image enlargement and reduction in the image. Their proposed methods using scanned handwritten chinese characters that reach the highest accuracy of all samples 64%. Meanwhile, the recognition accuracy rate of each chinese characters is over 92% [11]. In 2011, Kumar et al. implemented digitization the indian scripts as their input being converted into scanned document. The preprocessing using 3 steps: skew detection/correction, skeletonization, and noise removal/reduction. And also, the important steps such as: segmentation, feature extraction and classification applied [12].

Moreover for latin characters, in [4] use captured label of counterfeited products as their dataset. The preprocessed methods they used are : image binarization, noise reduction, image sharpening and image cropping. They calculate the accuracy for each methods, noise reduction: 78%, sharpening image: 76%, image cropping: 73% and binarization: 82%. In different methods, Sandip et al. applied labeling training data and segmentation in scanned Roman scripts. They test for total 1844 training sets and reach highest accuracy 83.50% [13].

Furthermore, character recognition problems using OCR has been more or less solved. The reserachers combining different type of technique preprocessing considering can produce high quality input for OCR prosess. This research aims to process in recognizing handwritting latin scripts with the combination of preprocessing technique such as: convert into grayscale, morphological operations, and noise removal. The details of proposed methods will be discussed in the next section.

3. PROPOSED METHOD

Figure 2. Illustrates the proposed method of this research was implemented in the following stages. The method is divided into three phases, i.e., input, process and output.

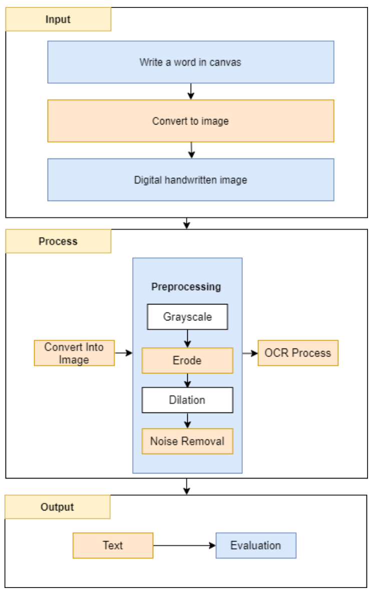


FIGURE 2. Proposed Methods

3.1 DATASET PREPARATION

This research developed a canvas in order to make a dataset of digital handwritten text. This feature capable of doing free-flow writing in website. This feature is developed because the author facing limitation in finding digital handwritten text dataset collection. In the previous study, the researchers used scanned of handwriting image as their dataset [3][12][14][15]. With total of 50 sets of image format digital handwritten image including the total of 323 characters as the dataset of the research. The image contains pixels, fuzzy, grainy in the edge, to achieve the high qualities input for OCR process, we implemented the preprocessing step that described in Section 3.2.

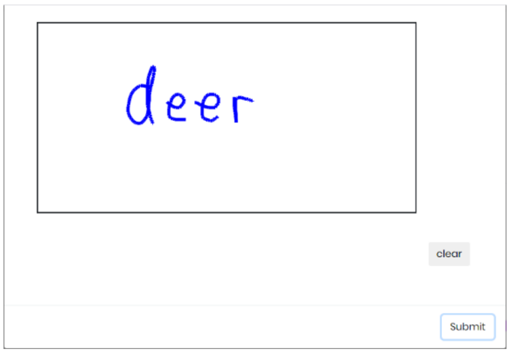


FIGURE 3. Writing Canvas

3.2 PREPROCESSING

Preprocessing is subjected as the initial stage of image processing before sent to the OCR process. We investigated the preprocessing techniques including convert into grayscale, basic morphological operations: erode and dilate, and noise removal need to be performed.

3.2.1 CONVERT INTO GRAYSCALE

First, converting the image into grayscale from the blue color text on the image. By setting the gray-level in averaging the intensity level of three colors pixel.

$$Gray = \frac{Red+Green+Blue}{3} \qquad (1)$$

This process aims to differentiate the pixel noise in the image. The grayscale image is easier to work in morphological operations than color image [16].

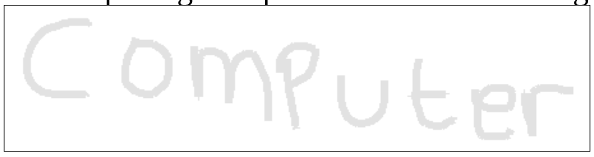


FIGURE 4. Grayscale Image

3.2.2 MORPHOLOGICAL OPERATION

Morphological operation is to filter the image by replacing the convolution operation by the logical operation. The basic set of morphological operations are erosion and dilation that processed the images based on shapes. This process aims to isolate the individual elements and joining disparate elements in the image, finding intensity bumps and holes in the characters, smoothing contours, thin characters, and extracting the boundaries [17].

ALGORITHM 1. Morphological Operations

```
1: #erode
2: Mat element = Imgproc.getStructuringElement(Imgproc.MORPH_RECT, new Size(3, 3));
3: Imgproc.erode(source, destination, element);
4:
5: #dilate
6: Mat element = Imgproc.getStructuringElement(Imgproc.MORPH_RECT, new Size(3, 3));
7: Imgproc.dilate(source, destination, element);
```

Moreover, morphological operations can be successfully implemented in handwritten text due to erratic hand and movement and different handwriting style [18].

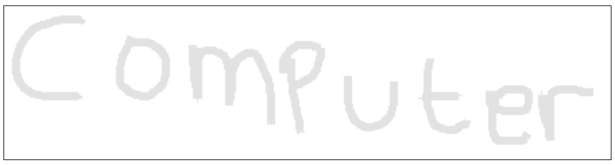


FIGURE 5. Erode

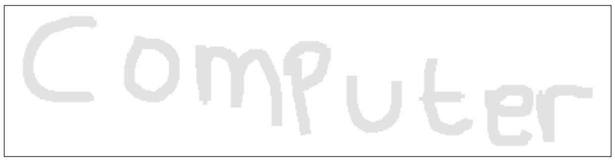


FIGURE 6. Dilate

In details, Figure 4. the words of ‘O’ rounded by holes in the edge of the word. After the process of erode and dilate (Figure 5. and Figure 6.), the holes and bumps are reduced. The output of morphological operations made the word into an accurate shape than before preprocessing.

3.2.3 NOISE REMOVAL

The last step of preprocessing is noise removal. The output of morphological operations still contains noise, so that the pixels that are not needed can be eliminated. Using median blur operations, the central element of image is replaced by the median of all pixels in the kernel area.

```
ALGORITHM 2. Noise Removal
1: #noise removal
2: Imgproc.medianBlur(source,destination,5);
```

In details of Figure 7. The character of ‘M’ shows a bump at the bottom. Performing noise removal operations, the bumps eliminated and produce an accurate shape of ‘M’.

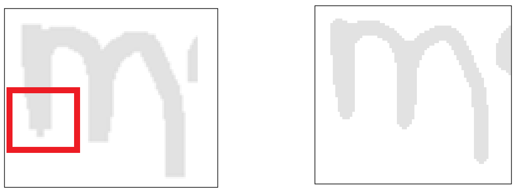


FIGURE 7. Before and After Noise Removal.

Furthermore, the usability of image preprocessing improves the character level which has an important role in OCR process recognizing in word level. Tesseract engines provide additional language regarding the language used. This research implemented the tessdata/eng.trainned data trained model as the dictionary of existing word in English.

3.2 OCR PROCESS

Figure 8 shows the process of Tesseract engine [5]. First, page layout analysis is for detecting the text area in the image. Second step, dividing the detected text area into a series of blobs. The blob is categorized as a classifiable unit, which may contain several characters or may contain a part of some character. The third step is to define the text lines and to combine the blobs into a series of words according to the blank space [19]. These three steps are prepared for the word recognition. The next step is to recognize each word in two passes. On each pass, Tesseract conducts various splitting and merging operations into blobs in one word and form a series of character outlines to be recognized [20]. On the first pass, Tesseract recognizes the characters outlines with static classifier based on the features library. If the recognition’s result is high confidence, the process will be continue to the adaptive classifier as training data. On the second pass, Tesseract tries in recognizes again those words with low confidence from the first pass, in order to omprove the recognition accuracy. The last step, Tesseract chooses only one result with the highest confidence as the output [5][19].

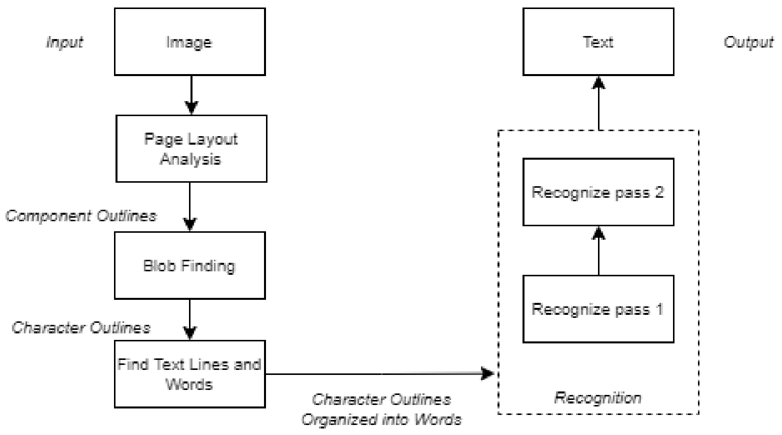


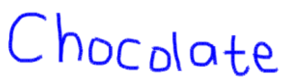
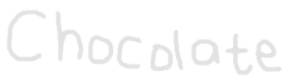
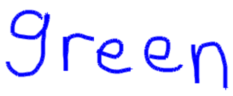
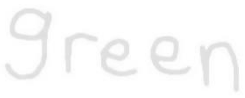
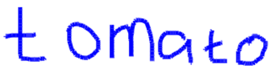
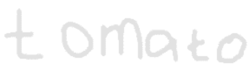
FIGURE 8. Tesseract engine process

4. RESULT AND DISCUSSION

A total 50 sets digital handwritten image and resulting in 323 characters. As the initial, the image contains pixels, fuzzy, and grainy. Preprocessing has been applied for each image including convert into grayscale in defining noise at the characters, basic morphological operations such as erode and dilate for joining disparate elements and finding intensity bumps and holes in the characters. The last step of preprocessing is noise removal, using median filter, unnecessary elements in the image can be omitted. The image preprocessing has been executed using OpenCV library which support in Java programming languages for this research. The comparation several results of OCR between original image and preprocessing presented in Table 1.

TABLE 1

Comparison Results of OCR Between Original and Preprocessing

No	Original	Preprocessing	OCR	
			Original	Preprocessing
1			Oe plate	Chocolate
2			SIRI	Ireen
3			SN PT	+OMato

OCR result with preprocessing shows significantly return nearly correct output compared with original image. Similarity characters (data no. 3) occurred between “t” and “+”, in the other results returning into numbers e.g. “Z” to ‘2’, “g” into “9”.

The calculation of accuracy rate of OCR on each image is determined on the number of character extraction results. In 1995, Rice et al. presented The 4th Annual Test of OCR Accuracy rate [21].

$$\frac{character\ count - character\ error}{character\ count}\ x\ 100\% \qquad (2)$$

OCR cannot produce a 100% accuracy rate for characters recognition, it depends on the applied technique approach for the OCR [20].

TABLE 2
Experiment results

Rate (%)	Original	Preprocessing
Average success	30.03%	79.26%

The test has been done and calculated the accuracy using Equals 2. for each image. Table 2 presents the average success of reading each character from handwritten image between original and preprocessing. From a total 50 sets images, an average of 30.03% of the characters on the original image can be extracted using Tesseract 4.x. Using the same OCR engine, an average of 79.26% can be read on the preprocessing image.

5. CONCLUSION

The research results in demonstrate recognizing characters from the digital handwritten image with preprocessing steps including grayscale, morphological operations and noise removal implemented before sent to OCR engine has improved the OCR performance. Using OCR engine Tesseract 4.x, original image reached the percentage of success to 30.03% in reading characters from the image. This is because Tesseract already applied the static character classifier, linguistic analysis and adaptive classifier in recognizing characters and words [8]. Meanwhile, preprocessing reach the percentage of success to 79.26% and successfully boost the OCR performance. Therefore, further studies need to explore in implementing more advance methods such as normalization, segmentation and skeletonization of preprocessing in order to get higher accuracy in recognizing digital handwritten both for latin and non-latin text.

REFERENCES

[1] C. Patel, A. Patel, and D. Patel, “Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study,” Int. J. Comput. Appl., 2012, doi: 10.5120/8794-2784.
[2] I. Kissos and N. Dershowitz, “OCR Error Correction Using Character Correction and Feature-Based Word Classification,” 2016, doi: 10.1109/DAS.2016.44.
[3] S. A. Alsuhibany and M. T. Parvez, “Secure Arabic handwritten CAPTCHA generation using OCR operations,” 2016, doi: 10.1109/ICFHR.2016.0035.

[4] J. BJÖRKMAN, Evaluation of the Effects of Different Preprocessing Methods on OCR Results from Images with Varying Quality, Degree project in computer science and engineering (Stockholm, Sweden, 2019).

[5] R. Smith, D. Antonova, and D. S. Lee, "Adapting the Tesseract open source OCR engine for multilingual OCR," 2009, doi: 10.1145/1577802.1577804.

[6] C. Geetha, "Optical Character Recognition with Tesseract," *J. Mech. Contin. Math. Sci.*, 2019, doi: 10.26782/jmcms.spl.2019.08.00006.

[7] R. Pramanik and S. Bag, "Shape decomposition-based handwritten compound character recognition for Bangla OCR," J. Vis. Commun. Image Represent., 2018, doi: 10.1016/j.jvcir.2017.11.016.

[8] R. Smith, "An overview of the tesseract OCR engine," 2007, doi: 10.1109/ICDAR.2007.4376991.

[9] M. Kaur, N. S. Randhawa, and V. Garg, "Proposed Approach for Layout & Handwritten Character Recognization in OCR," Int. Res. J. Eng. Technol., 2019.

[10] Anuradha, C. Liyanage, H. Wijayawardhana, and R. Weerasinghe, "Deep learning based sinhala Optical Character Recognition (OCR)," 2020, doi: 10.1109/ICTer51097.2020.9325428.

[11] Q. Li, W. An, A. Zhou, and L. Ma, "Recognition of offline handwritten Chinese characters using the tesseract open source OCR engine," 2016, doi: 10.1109/IHMSC.2016.239.

[12] M. Kumar, M. K. Jindal, and R. K. Sharma, "Review on OCR for handwritten indian scripts character recognition," 2011, doi: 10.1007/978-3-642-24055-3_28.

[13] S. Rakshit, S. Basu, "Recognition of Handwritten Roman Script Using Tesseract Open source OCR Engine", Proc. National Conference on NAQC, 2008.

[14] G. Abdul Robby, A. Tandra, I. Susanto, J. Harefa, and A. Chowanda, "Implementation of optical character recognition using tesseract with the javanese script target in android application," 2019, doi: 10.1016/j.procs.2019.09.006.

[15] R. KumarNath and M. Rastogi, "Improving Various Offline Techniques used for Handwritten Character Recognition : A Review," Int. J. Comput. Appl., 2012, doi: 10.5120/7726-1136.

[16] C. Kanan and G. W. Cottrell, "Color-to-grayscale: Does the method matter in image recognition?," PLoS One, 2012, doi: 10.1371/journal.pone.0029740.

[17] I. Saito, K. Sadamitsu, H. Asano, and Y. Matsuo, "Morphological Analysis for Japanese Noisy Text based on Extraction of Character Transformation Patterns and Lexical Normalization," J. Nat. Lang. Process., 2017, doi: 10.5715/jnlp.24.297.

[18] J. Singh, G. Singh, R. Singh, and P. Singh, "Morphological evaluation and sentiment analysis of Punjabi text using deep learning classification," J. King Saud Univ. - Comput. Inf. Sci., 2021, doi: 10.1016/j.jksuci.2018.04.003.

[19] R. Smith, D. Antonova, and D. S. Lee, "Adapting the Tesseract open source OCR engine for multilingual OCR," 2009, doi: 10.1145/1577802.1577804.

[20] H. Singh and A. Sachan, "A Proposed Approach for Character Recognition Using Document Analysis with OCR," 2019, doi: 10.1109/ICCONS.2018.8663011.

[21] P. Pangestu, D. Gunawan, and S. Hansun, “Histogram equalization implementation in the preprocessing phase on optical character recognition,” Int. J. Technol., 2017, doi: 10.14716/ijtech.v8i5.877.
[22] K. R. Sanjuna and K. Dinakaran, “A multi-object feature selection based text detection and extraction using Skeletonized Region optical character recognition in-text images,” Int. J. Eng. Technol., 2018, doi: 10.14419/ijet.v7i3.6.16009.
[23] E. Promin and P. Suriyachai, “Improvement of Scanned Medical Document Management System,” 2019, doi: 10.1109/KST.2019.8687398.
[24] M. G. Marne, P. R. Futane, S. B. Kolekar, A. D. Lakhadive, and S. K. Marathe, “Identification of Optimal Optical Character Recognition (OCR) Engine for Proposed System,” 2018, doi: 10.1109/ICCUBEA.2018.8697487.
[25] M. Koistinen, K. Kettunen, and J. Kervinen, “How to Improve Optical Character Recognition of Historical Finnish Newspapers Using Open Source Tesseract OCR Engine – Final Notes on Development and Evaluation,” 2020, doi: 10.1007/978-3-030-66527-2_2.
[26] I. M. A. Mahawan and I. P. A. E. Darma Udayana, “Implementation of Average Filter and Median Filter for OCR Pre Processing of Incoming Letters Image,” 2020, doi: 10.1088/1757-899X/846/1/012021.
[27] D. Sporici, E. Cuşnir, and C. A. Boiangiu, “Improving the accuracy of Tesseract 4.0 OCR engine using convolution-based preprocessing,” Symmetry (Basel)., 2020, doi: 10.3390/SYM12050715.
[28] M. K. Sahu and D. N. K. Dewangan, “A Survey on Handwritten Character Recognition,” IARJSET, 2017, doi: 10.17148/iarjset.2017.4120.