JOIN (Jurnal Online Informatika)
Volume 5 No.1 | June 2020: 73-80
DOI: 10.15575/join.v5i1.580

p-ISSN: 2528-1682
e-ISSN: 2527-9165

Comparison of Template Matching Algorithm and Feature Extraction Algorithm in Sundanese Script Transliteration Application using Optical Character Recognition

Yana Aditia Gerhana[1], Muhammad Farid Padilah[2], Aldy Rialdy Atmadja[3]
[1,2,3]Department of Informatics, UIN Sunan Gunung Djati Bandung, Indonesia

Article Info

Article history:

Received May 16, 2019
Revised June 28, 2020
Accepted July 14, 2020
Published July 15, 2020

Keywords:

Feature Extraction Algorithm
Luminosity Algorithm
Matching Template Algorithm
Sundanese Script
Thresholding Algorithm

ABSTRACT

The phenomenon that occurs in the area of West Java Province is that the people do not preserve their culture, especially regional literature, namely Sundanese script, in this digital era there is research on Sundanese script combined with applications using Feature Extraction algorithm, but there is no comparison with other algorithms and cannot recognize Sundanese numbers. Therefore, to develop the research a Sundanese script application was made with the implementation of OCR (Optical Character Recognition) using the Template Matching algorithm and the Feature Extraction algorithm that was modified with the pre-processing stages including using luminosity and thresholding algorithms, from the two algorithms compared to the accuracy and time values the process of recognizing digital writing and handwriting, the results of testing digital writing algorithm Matching algorithm has a value of 87% word recognition accuracy with 236 ms processing time and 97.6% character recognition accuracy with 227 ms processing time, Feature Extraction has 98% word recognition accuracy with 73.6 ms processing time and 100% character recognition accuracy with 66 ms processing time, for handwriting recognition in feature extraction character recognition has 83% accuracy and 75% word recognition , while template matching in character recognition has an accuracy of 70% and word recognition has an accuracy of 66%.

Corresponding Author:

Yana Aditia Gerhana,
Department of Informatics,
UIN Sunan Gunung Djati Bandung,
Jl. AH. Nasution No. 105 Bandung, Indonesia
Email: yanagerhana@uinsgd.ac.id

1. INTRODUCTION

Every country has some wealth that can be preserved. One of the wealth owned by a country is culture. Indonesia has a country that has a very diverse culture, including script, language, and literature. At present, culture is required to keep pace with technological developments, so that culture is not used up by time. Therefore, the role of digitalization is very influential on current cultural development. Identifying scripts, documents, images, and videos are commonly done, along with the development of digital technology. This also underlies some research in digital image recognition. This image recognition is also embedded in devices such as scanner, digital camera, and smartphone [1].

Optical Character Recognition (OCR) is the classification process of optical patterns to recognize characters from a digital image [2][3][4][5][6][7][8][9]. OCR is replicate human functions and belongs to the family of machine recognition techniques performing automatic identification. OCR technology enables to transform documents such as scanned paper, pdf files, or images from the digital camera [10]. After transforming data, that can be reprocessed again, so the document is going to be editable and searchable data [11]. With the presence of OCR, text that is scanned with OCR can be identified as a case or word which can then be translated according to training data, and every text can be manipulated, replaced, or given a

barcode[12]. To recognize the pattern, OCR is doing the steps of segmentation, feature extraction, and classification[10].

On the other hand, the development of the introduction of letters or handwriting among them is doing optimization so that it adds a component of artificial intelligence with a view to increase accuracy in writing recognition. The research for recognition in Arabic font-written script is also done by Rashwan, Fakhr, Attia, El-Mahallawy. This research explores the HMM-based approach to building Arabic font-written OCR. The experiment is done with real-life documents that have more punctuation and special symbols. The data consists of 540 pages of the text scanned at 600 dpi with two conditions: font-dependent and multi-font. The experimental result showed that Mudir font has lower Word Error Rate (WER) compare to Simplified and Traditional font[13].

Many types of research have been done for recognizing text using OCR. Another research proposed a method for recognizing English character with many fonts. The neural network is the method for solving the problem. Binary images of English character with different font are used as database. This database includes 237 samples of each character. There are five steps to classify character: pre-processing, feature extraction, calculating similarity measure to select feature, classification using genetic algorithm and SOM network, and evaluation. The result showed that the proposed method is able to recognize English characters with 98.5% accuracy[14].

In addition, OCR also implements for the Qatari number plate. This research-based on feature extraction and template matching algorithms. In this paper, four algorithms are applied to Qatari number plates. The proposed algorithms are based on feature extraction (vector crossing, zoning, combined zoning, and vector crossing) and template matching techniques. All algorithms have been tested by MATLAB. In this experiment, 2790 Qatari binary character images were used to test the algorithms. The template matching based algorithm showed the perfect recognition rate of 99.5% with an average time of 1.95 ms per character. However, this paper focused on comparing feature extraction and template matching for recognizing the Sundanese character and translate it into Latin character. There are several improvements to optimize in recognizing the Sundanese numbers. This paper includes 4 sections. First, the introduction discusses OCR and some research that is relevant to OCR. The second part explains the methods are used in this study, including the luminosity algorithm, thresholding, feature extraction, and template matching. The following part focused on testing and the experimental results by comparing two algorithms. Last, the conclusions and suggestions that support future research related to recognizing character by OCR image.

2. METHOD

This section explains about the method that is implemented in this research. The research contains the following steps. First, the luminosity method is applied to convert RGB color to grayscale. Next, the Grayscale Image must set into a binary value. This way makes an image is easier to distinguish structural features. So, an image with binary value can make it easy to distinguish objects from the background[15][16]. Threshold values are used to set binary values that can be generated from an image. Furthermore, the binary images are read by OCR and adjusted by the Feature Extraction and Template Matching Algorithm[9]. Next, the algorithm has to match training data, so it can generate Latin character from the Sundanese character inputs. All of the steps described in the following figure 1.

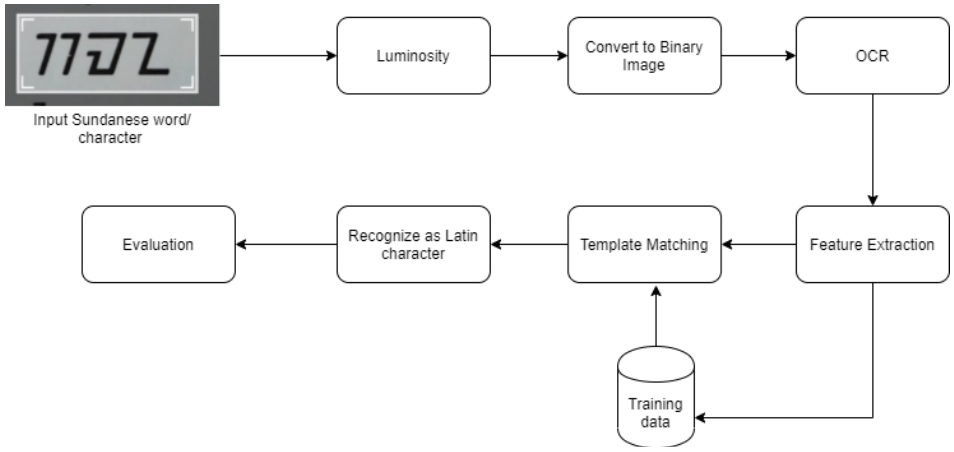

Figure 1. Steps to Recognize Sundanese character

2.1. Luminosity

The intensity of a pixel has a range between a minimum and a maximum. This range is represented in an abstract way as a range from 0 (total absence, black) and 1 (total presence, white), with any fractional values in between. To converting RGB to grayscale/intensity, it must have specific weights to R, G, and B colors that are to be applied. These weights are: 0.2989, 0.5870, 0.1140[17]. In these steps, input images with RGB color are converted into grayscale images. Grayscale images are images that only have gray images, which means the intensity level of each color is the same, has an intensity of 0 to 255. The equation as in follows:

$$Y = (0.299*R) + (0.587*G) + (0.114*B) \quad (1)$$

R = Red Color Component
G = Green Color Component
B = Blue Color Component

2.2. Convert to Binary Image

There is a way to convert Greyscale images to Binary images. It can apply with a Threshold value. Threshold value is the value of greyscale images valued from 0 to 255, by determining the threshold value we can set which images will be included in the value 0 (White) and which images will be included in the value 1 (Black). The process of changing the image can be formulated with the following equation:

$$b(i) = \begin{cases} 0, & i \geq a \\ 1, & i < a \end{cases} \qquad b(i) = \begin{cases} 0, & i \geq a \\ 1, & i < a \end{cases} \quad (2)$$

a = Threshold value
I = Pixel intensity
b = Binary value (0,1)

2.3. Feature Extraction

Feature extraction is a method to construct a smaller set of features by non-linearly combining existing features. In contrast, feature selection methods assign each feature, which has an importance value. This method is used to filter the set of features[18]. This is an example of feature selection based on the extraction of feature such as comparison of width and height (Ratio), Number of Horizontal lines, Number of vertical lines, open character image (top, bottom, right, left), the intersection of vertical lines in the middle of the image, intersection of horizontal lines in the middle of the image. The illustration extraction of features is shown in the figure 2. K-Nearest Neighbor (KNN) algorithm is used to recognize the character, and the result is compared with the closest neighbors (the closest data template).

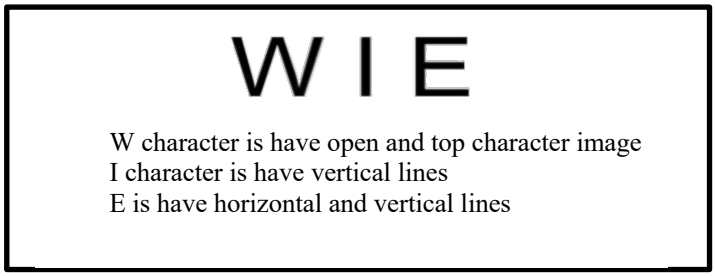

Figure 2. Illustration of Extraction of Feature

2.4. Template Matching

Template matching is an algorithm that has the concept of matching every pixel of a binary image to the template from the training data. The binary image that is inputted will be calculated the correlation value (match value) the largest correlation value that is considered to be in accordance with the template[19][20]. Template matching recognizes by calculating the correlation value between the input image and the template image.

$$r = \frac{\sum_{k=1}^{n}(x_{ik}-\bar{x}_i)\cdot(x_{jk}-\bar{x}_j)}{\sqrt{\left[\sum_{k=1}^{n}(x_{ik}-\bar{x}_i)^2.\sum_{k=1}^{n}(x_{jk}-\bar{x}_j)^2\right]}} \quad (3)$$

p-ISSN: 2528-1682
e-ISSN: 2527-9165

n = number of pixel in a matrices
r = correlation value of two matrices
X_{ij} = pixel value of matrices with cell in i x j (matrices template)
X_{ik} = pixel value of matrices with cell in i x k
X_j = average value of a pixel in matrices (j)
X_i = average value of a pixel in matrices (i)

In figure 3 showed the example of data. The data has converted into a binary image. This binary image is known as a binary template. The template will be compared to another template.

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	0	1	1	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3. The example of binary template

The recognition process is done with the Template Matching Correlation. This method is useful for finding images that match the input image. The template matching is done for comparing the binary template with the available data template. The correlation value obtained from the template image with the input data is 0.634012. Figure 4 is shown the comparison between a binary template and input image as a data template.

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	0	1	1	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	1	1	1	1	0
0	0	1	1	1	1	0	0	0	1	1	1	1	0
0	0	0	0	1	1	0	0	0	0	0	1	1	0
0	0	0	1	1	0	0	0	0	0	1	1	0	0
0	0	1	1	0	0	0	0	0	1	1	0	0	0
0	1	1	1	0	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	1	1	1	0	0	0	0
1	1	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4. The Comparation between binary template and image data template (1)

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	0	1	1	0	0	0	0	1	1	0
0	0	0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	0	1	1	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5. The Comparation between binary template and image data template (2)

From figure 2, the calculation of input images compares to the binary template have got a correlation value of 0.86711. The result showed that the image data template 2 has a higher value than image data template 1. So, the most suitable data is the data which have the largest correlation value, namely the second image data template. Then, the input letters are recognized as the second input data.

3. RESULTS AND DISCUSSION

The aim is based on the variation of Sundanese characters and speech. The typeface used in digital writing testing is Sundaness.ttf, while the handwriting test is done by taking three different handwritten sample data with the image format used is png. The test parameters used are the processing time and the level of accuracy of the algorithm in recognizing the Sundanese script. Test data used for digital writing were 42 Sundanese characters and 30 Sundanese words. Whereas ten characters Sundanese dan handwriting and ten Sundanese words, with three different examples of writing.

3.1. Digital writing test results of the Sundanese script introduction.

The test is based on Sundanese characters, vowels, consonants, and numbers of 42 characters. The test results show that the accuracy rate of the feature extraction algorithm is 100%, while template matching is 97.6%, or with an error rate of 2.4%. While the average processing time of Feature Extraction is 66,023 ms, and Template Matching is 220,547 ms. Figure 2 explains the comparison of the time of the digital writing process of character writing between the Matching Template and Feature Extraction, and the Feature Extraction processing time tends to be faster than the Matching Template.

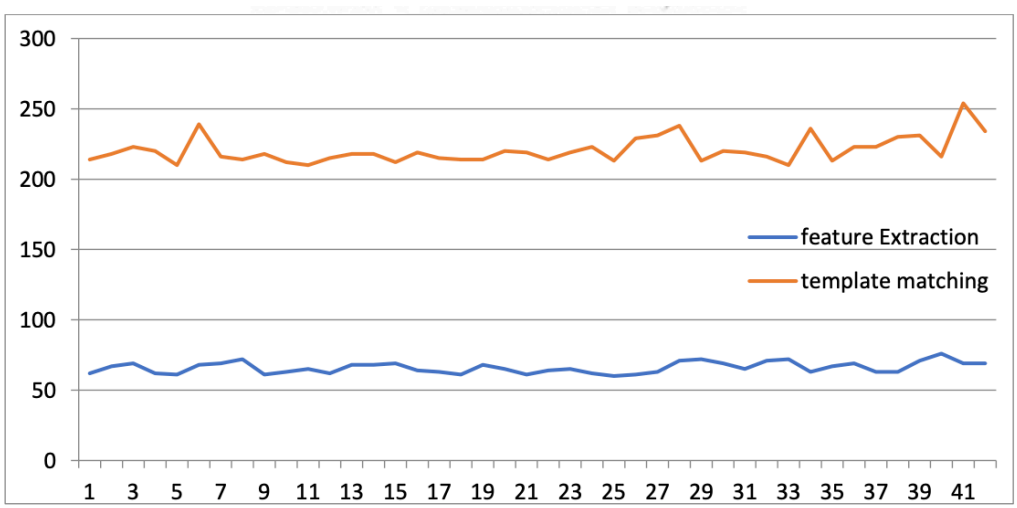

Figure 6. Comparison of processing time for digital writing

The next test is based on the number of words in the Sundanese script using 30 sample words. Based on the test results, the accuracy obtained by Feature Extraction is 98%, and Template Matching is 87%, or with an error rate of 13%. While the average processing time of Feature Extraction is 73.6 ms, and Matching Templates is 236 ms. Figure 3 explains the comparison of the time of the digital writing process between the Matching Template and Feature Extraction, and the tendency of the Feature Extraction processing time is faster than the Matching Template.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

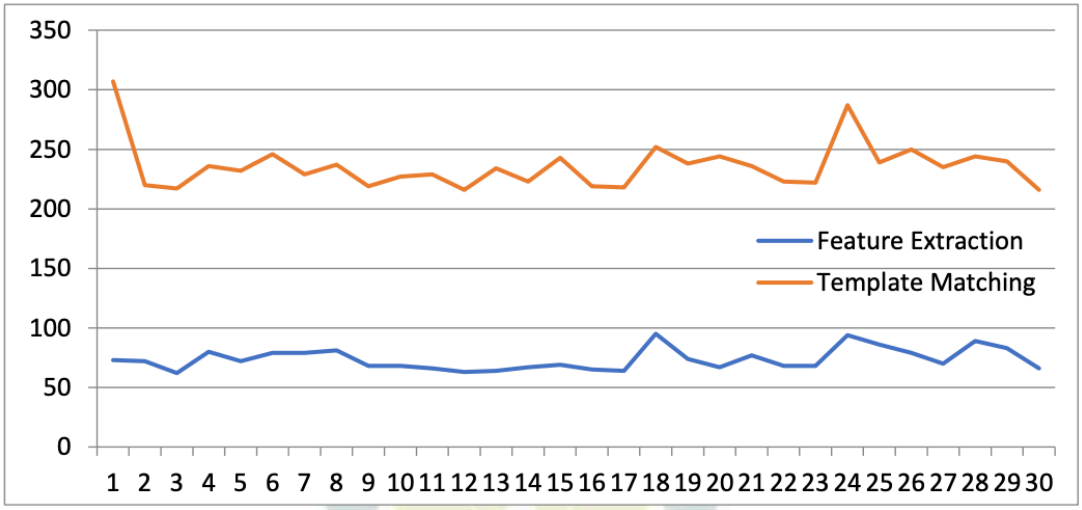

Figure 7. Comparison of processing time for digital writing

3.2. Testing of Sundanese script handwriting recognition

Handwriting testing is done by using three data samples from different handwriting, ten data for character recognition, and ten data for word recognition. The test results, the accuracy of the feature extraction algorithm is 83%, while template matching is 70%, or with an error rate of 30%. While the average processing time of Feature Extraction is 71.5 ms, and Template Matching is 226 ms. Figure 4 explains the comparison of the time of the character's Sundanese handwriting process between the Matching Template and Feature Extraction, and the Feature Extraction processing time tends to be faster than the Template Matching.

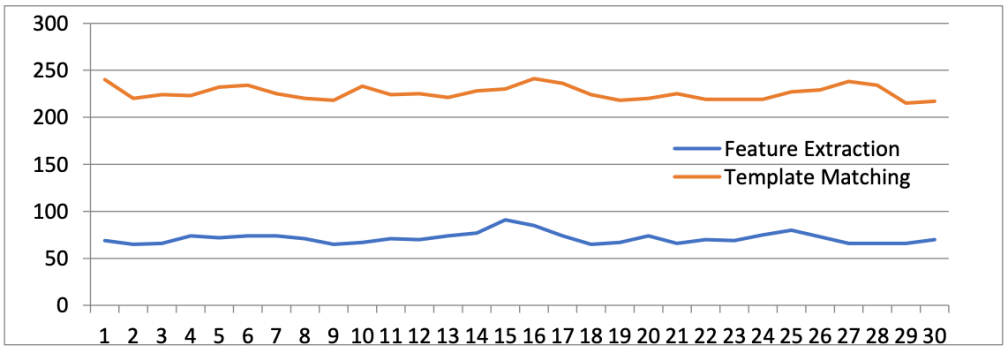

Figure 8. Comparison of processing time for handwriting

The next test is based on the number of words in the Sundanese script, using 30 sample words. Based on the results of testing, the accuracy obtained by Feature Extraction is 75%, and Template Matching is 66%, or with an error rate of 34%. While the average processing time of Feature Extraction is 99.83 ms, and Template Matching is 257.4 ms. Figure 5 explains the comparison of the time of the handwriting process between the Matching Template and Feature Extraction, and the tendency of the Feature Extraction processing time to be faster than the Matching Template.

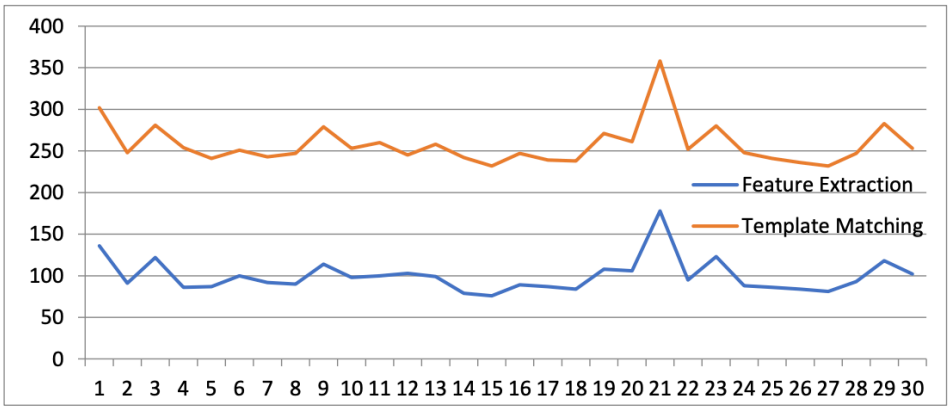

Figure 9. Comparison of processing time for handwriting

4. CONCLUSION

Based on the test results of the introduction of Sundanese script using digital writing or handwriting, the accuracy of reading Sundanese characters and words using the feature extraction algorithm is higher than the template matching algorithm. Similarly, the average processing time required for feature extraction algorithms is faster compared to template matching algorithms. In order to avoid noise and be able to maximize the pattern recognition process, further research in the pre-processing process can use other OCR algorithms, such as Otsu Threshold. The amount of training data is reproduced by using a high-resolution camera.

ACKNOWLEDGEMENTS

This research was supported/partially supported by department of informatics UIN Sunan Gunung Djati Bandung. We thank our colleagues from department of informatics UIN Sunan gunung Djati Bandung who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper.

5. REFERENCES

[1] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: A survey," *Int. J. Doc. Anal. Recognit.*, vol. 7, no. 2–3, pp. 84–104, 2005, doi: 10.1007/s10032-004-0138-z.

[2] R. D. Zarro and M. A. Anwer, "Recognition-based online Kurdish character recognition using hidden Markov model and harmony search," *Eng. Sci. Technol. an Int. J.*, vol. 20, no. 2, pp. 783–794, 2017, doi: 10.1016/j.jestch.2016.11.016.

[3] M. Soua, R. Kachouri, and M. Akil, "Efficient multiscale and multifont optical character recognition system based on robust feature description," *5th Int. Conf. Image Process. Theory, Tools Appl. 2015, IPTA 2015*, pp. 575–580, 2015, doi: 10.1109/IPTA.2015.7367214.

[4] S. Desai and A. Singh, "Optical character recognition using template matching and back propagation algorithm," *Proc. Int. Conf. Inven. Comput. Technol. ICICT 2016*, vol. 2016, 2016, doi: 10.1109/INVENTIVE.2016.7830161.

[5] D. Kalina and R. Golovanov, "Application of template matching for optical character recognition," *Proc. 2019 IEEE Conf. Russ. Young Res. Electr. Electron. Eng. ElConRus 2019*, pp. 2213–2217, 2019, doi: 10.1109/EIConRus.2019.8657297.

[6] K. El Gajoui, F. A. Allah, and M. Oumsis, "Diacritical Language OCR Based on Neural Network: Case of Amazigh Language," *Procedia Comput. Sci.*, vol. 73, no. Awict, pp. 298–305, 2015, doi: 10.1016/j.procs.2015.12.035.

[7] A. Farhat *et al.*, "OCR based feature extraction and template matching algorithms for Qatari number plate," *2016 Int. Conf. Ind. Informatics Comput. Syst. CIICS 2016*, 2016, doi: 10.1109/ICCSII.2016.7462419.

[8] P. Ahmed and Y. Al-Ohali, "Arabic Character Recognition: Progress and Challenges," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 12, pp. 85–116, 2000, doi: 10.1016/s1319-1578(00)80004-x.

[9] M. Ryan and N. Hanafiah, "An Examination of Character Recognition on ID cardusing Template Matching Approach," *Procedia Comput. Sci.*, vol. 59, pp. 520–529, 2015, doi: 10.1016/j.procs.2015.07.534.

[10] A. Chaudhuri, K. Mandaviya, P. Badelia, and S. K Ghosh, *Optical Character Recognition Systems for Different Languages with Soft Computing*, vol. 352, no. December. 2017.

[11] D. B. Alamsyah, "Implementasi Text Recognition untuk mendeteksi Digital Writing dan Handwriting dalam Alfabet Latin menggunakan OCR (Optical Character Recognition)," Aug. 2018.

[12] A. E. Utami, O. D. Nurhayati, and K. T. Martono, "Aplikasi Penerjemah Bahasa Inggris – Indonesia dengan Optical Character Recognition Berbasis Android," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 1, p. 167, Jan. 2016, doi: 10.14710/jtsiskom.4.1.2016.167-177.

[13] M. A. A. Rashwan, M. W. T. Fakhr, M. Attia, and M. El-Mahallawy, "Arabic OCR system analogous to HMM-based ASR systems; implementation and evaluation," *J. Eng. Appl. Sci.*, vol. 54, no. 6, pp. 653–672, 2007.
[14] N. Samadiani and H. Hassanpour, "A neural network-based approach for recognizing multi-font printed English characters," *J. Electr. Syst. Inf. Technol.*, vol. 2, no. 2, pp. 207–218, 2015, doi: 10.1016/j.jesit.2015.06.003.
[15] C. Saravanan, "Color image to grayscale image conversion," *2010 2nd Int. Conf. Comput. Eng. Appl. ICCEA 2010*, vol. 2, no. April 2010, pp. 196–199, 2010, doi: 10.1109/ICCEA.2010.192.
[16] S. R. Nayak and J. Mishra, "An Improved Method to Estimate the Fractal Dimension of Color Images," *Perspect. Sci.*, vol. 8, pp. 412–416, 2016, doi: 10.1016/j.pisc.2016.04.092.
[17] G. Jyothi, C. Sushma, and D. S. S. Veeresh, "Luminance Based Conversion of Gray Scale Image to RGB Image," *Int. J. Comput. Sci. Inf. Technol. Res.*, vol. 3, no. 3, pp. 279–283, 2015.
[18] S. Pölsterl, S. Conceti, N. Navab, and A. Katouzian, "Survival analysis for high-dimensional, heterogeneous medical data: Exploring feature extraction as an alternative to feature selection," *Artif. Intell. Med.*, vol. Volume 72, pp. 1–11, 2016, doi: 10.1016/j.artmed.2016.07.004.
[19] Y. Jisung, S. H. Sung, D. K. Seong, S. K. Myung, and C. Jihun, "Scale-invariant template matching using histogram of dominant gradients," *Pattern Recognit.*, vol. 47, no. 9, 2014, doi: 10.1016/j.patcog.2014.02.016.
[20] J.Weber and S.Levre, "Spatial and spectral morphological template matching," *Image Vis. Comput.*, vol. 30, no. 12, pp. 934–945, 2012, doi: 10.1016/j.imavis.2012.07.002.