Scientific Journal of Informatics
Vol. 8, No. 2, November 2021

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Histogram of Gradient in K-Nearest Neighbor for Javanese Alphabet Classification

Ajib Susanto[1], Christy Atika Sari[2*], Ibnu Utomo Wahyu Mulyono[3], Mohamed Doheir[4]
[1,2,3]Informatics Engineering Department, Faculty of Computer Sciences, Dian Nuswantoro University, Indonesia
[4]Software Engineering, Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Malaysia

Abstract.
Purpose: The Javanese script generally has a basic script or is commonly referred to as the "carakan" script. The script consists of 20 letters with different levels of difficulty. Some letters have similarities, so research is needed to make it easier to detect the image of Javanese characters.
Methods: This study proposes recognizing Hiragana's writing characters using the K-Nearest Neighbor (K-NN) method. In the preprocessing stage, the segmentation process is carried out using the thresholding method to perform segmentation, followed by the Histogram of Gradient (HOG) feature extraction process and noise removal using median filtering. Histogram of Gradient (HoG) is one of the features used in computer vision and image processing in detecting an object in the form of a descriptor feature. There are 1000 data divided into 20 classes. Each class represents one letter of the basic Javanese script.
Result: Based on data collection using the writings of 50 respondents where each respondent writes 20 basic Javanese characters, the highest accuracy was obtained at K = 1, namely 98.5%.
Novelty: Using several preprocessing such as cropping, median filtering, otsu thresholding and HOG feature extraction before do classification, this experiment yields a good accuracy.

Keywords: Javanese character, K-NN, HOG, median filter
Received September 2021 / **Revised** October 2021 / **Accepted** November 2021

INTRODUCTION

Indonesia has several types of historical heritage in the form of buildings, inscriptions, literary works and so on as a form of inheritance from ancestors. One of them is the Javanese script which has received international recognition, namely Unicode (an institution under the auspices of UNESCO). The institution recognized Javanese script on October 2, 2009 which Ki Demang Sokowaten previously registered from Yogyakarta. This recognition certainly has a good impact, such as being protected from claims by other parties and avoiding the threat of extinction. However, the problem now is that people no longer use Javanese scripts in their daily activities, especially Javanese people. It causes many people who have not been able to write Javanese script correctly because they have forgotten and are difficult to memorize [1]–[3]. The Javanese script itself has a total of 20 types of basic letter characters [4]. The basic script is also commonly referred to as the "carakan" or "nglegena" script.

In writing activities, everyone will have written results in various forms. It is because each person has their style and mood when writing [5]. The various variations or characteristics of the writing often create confusion in recognizing letters in an article. It causes misunderstanding and misperception for those who read it. This impact can be exacerbated when writing letters not commonly written every day, such as Javanese scripts. There are Javanese script letters that have the same shape, and each letter also has different complexity or complexity [6] [7]. Therefore, most people have difficulty writing Javanese scripts, not even a few people who have not recognized or read Javanese script letters [8].

An image processing-based approach known as Handwritten Character Recognition makes it easier for people to recognize Javanese characters. Handwritten Character Recognition is a research topic that

[*] Corresponding author
Email addresses: ajib.susanto@dsn.dinus.ac.id (Susanto), christy.atika.sari@dsn.dinus.ac.id (Sari), ibnu.utomo.wm@dsn.dinus.ac.id (Mulyono), jeruslem.20088@gmail.com (Doheir)
DOI: 10.15294/sji.v8i2.30788

discusses how computers can recognize and process a character in physical or printed media such as documents, images, and other media [9]. The research topic is generally better known as Optical Character Recognition (OCR). Basically, Optical Character Recognition (OCR) is a computer's ability to detect and convert text in printed documents or from handwriting into text formats that can be manipulated or changed [10]. Character Recognition (CR) is divided into 2 types, namely online recognition and offline recognition. Online recognition is a character recognition process where the character is obtained from the movement of a pen or finger on a touchscreen device or pen tablet. While offline recognition is the process of recognizing characters obtained from the results of a scanner or camera device.

In offline recognition, the captured image will undergo a computerized process using several methods or algorithms in digital image processing. One of them is a classification algorithm. The method works by predicting test data into class labels that have undergone a learning process. Some examples of classification algorithms that can be used are K-Nearest Neighbor (K-NN) [11][12][10], Support Vector Machine (SVM) [13][14], Decision Tree (DT) [15], dan Naive Bayes (NB). Research on handwritten character recognition has been conducted by P. Kamble and R. Hegadi using K-Nearest Neighbor (K-NN) as a classification method [16]. The data used is the image of handwritten Marathi characters using four classes of labels, namely numerals, vowels, consonants, and mixed. This study obtained an average accuracy of 88.56% from each test result of 4 types of class labels.

Based on these researches, the purpose of this paper is to optimize the classification of Javanese letters sing K-Nearest Neighbor (K-NN). Here, Histogram of Gradient (HoG) feature extraction is used as an optimization to define feature areas that group the pixel gradient values according to the orientation of each local part of the image. Then an experiment will be carried out to introduce handwritten Javanese letters based on the HoG descriptor feature as a parameter to calculate the distance proximity value using the K-NN classification algorithm.

METHODS

State of The Art

There is research related to handwritten character recognition, such as the recognition of numbers and letters. The research uses various methods in classifying data and extracting features or features as parameters. From several previous studies, researchers will choose to be used as references and relevant related to the research to be carried out. According to M. Sudarma and I. Surya Darma [18]. Papyrus manuscripts are Balinese heritages that need to be preserved and cared for because some manuscripts are damaged or fragile. In the papyrus, the script contains writing using Balinese characters. So, research was conducted to identify Balinese characters using K-Nearest Neighbor (K-NN) and semantic feature extraction. The semantic feature shows the shape of the Balinese character, which consists of vertical lines, horizontal lines, endpoints of writing, lines that form iterations, total rows, and total columns. The study results obtained an accuracy of 88.89%, where there are still characters classified incorrectly because the features obtained from each character have similarities. An example between the letters Ca and Sa.

According to R. Mojtaba Mohammadpoor [19], Local Binary Pattern (LBP) and Histogram of Gradient (HoG) features can be used as features in handwritten recognition. His research is on handwritten Persian numeral recognition which has difficulty in pattern recognition because some Persian numerals have the same character shape. This problem is compounded by the use of handwritten Persian numeral data. The data used comes from the HODA database with a total of 80,000 images which are divided into 60,000 images for training data and 20,000 images for test data. The proposed method is to combine LBP and HoG features for each handwritten Persian numeral image. The classification used is Multi-class Support Vector Machine (M-SVM). From the research results obtained an accuracy of 99.3%.

According to R. Kaur and Priyadarhni [20], in their research on the introduction of serial numbers on banknotes to overcome the problem of counterfeit money and help regulate currency circulation. The Histogram of Gradient (HoG) is used to extract character and number features on banknotes and the K-Nearest Neighbor (K-NN) algorithm is used for feature classification. The paper money used as data is obtained from the scanner. Then the preprocessing process is carried out such as detecting and extracting the serial number based on ROI and bounding box. To adjust the contrast level of the image and minimize the intensity variation with the background, linear contrast stretching and morphological opening are used.

Cell size [4 4] was selected to extract HoG features. This results in a feature length of 324. The average accuracy is 83.81%.

Javanese Character

Javanese script is a historical relic in writing since the 17th century or during the reign of ancient Mataram. At that time people used Javanese script as their daily language. The Javanese script itself generally has a basic script or is also commonly referred to as the “carakan” or “nglegena” script [6].

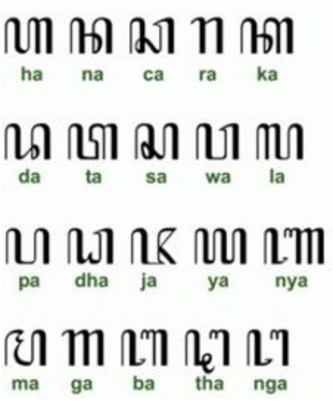


Figure 1. Javanese Character

Based on Figure 1, the script consists of 20 letters with different difficulty levels and some letters have similarities. In addition to the basic script / “carakan” , there are also murda script, sound script, fictitious script, sandhangan script, pre-sign script, and partner script.

HoG as Feature Extraction

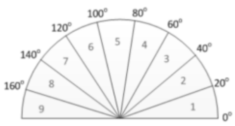
Feature extraction is a process for taking features or characteristics of an object based on certain characteristics of the object. So with the features or characteristics obtained, then an object can be classified easily by a classifier using the characteristics of each object based on certain classes. One of them is shape feature extraction which describes objects based on line and contour configurations. Shape features are categorized into 2, namely boundary-based and region-based. The boundary-based is used to describe the number of pixels that are on the object boundary. While region-based is the number of pixels that make up an object. Histogram of Gradient (HoG) is one of the features used in computer vision and image processing in detecting an object in the form of a descriptor feature. In forming the HoG feature, several steps are needed to divide the image into smaller, connected parts called cells. Each cell is calculated its HoG direction or edge orientation for the pixels in the cell. Based on HOG, calculate the x and y directions in the image f(x,y), where f(x,y) is the brightness value in the image has been shown in (1). After the x and y values are obtained, then the gradient magnitude (Arg) and gradient orientation (θ) can be searched according (2) and (3).

$$f(x,y) = \begin{cases} fx\,(x,y) = f(x+1,y) - f(x-1,y) \\ fy(x,y) = f(x,y+1) - f(x,y-1) \end{cases} \tag{1}$$

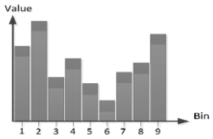
$$Arg(x,y) = \sqrt{fx(x,y)^2 + fy(x,y)^2} \tag{2}$$

$$(x,y) = arctan\,\frac{fx(x,y)}{fy(x,y)} \tag{3}$$

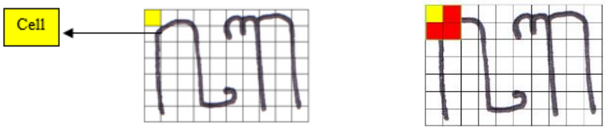
Calculate discrete values of each pixel in angular bins according to gradient orientation, because each cell contributes to the gradient weight in the corresponding angular bins. The angular bins image process is divided into 9 areas with an average binning angle of 20 degrees.



HoG Degrees



Histogram in HoG



HoG Cells Cell inside of Block

Figure 2. HoG Cells

Based on Figure 2, Adjacent cell groups are considered as spatial regions referred to as blocks. This grouping of cells into blocks is the basis for grouping and normalizing histograms. Histogram normalization is performed to form a histogram block. This set of block histograms represents descriptors. Calculating normalization is done by dividing the feature vector v by the appropriate L2-norm using (4). Where ε is a constant preventing the occurrence of a value of 0, which is generally 1. While v is a normalized vector.

$$V \rightarrow \frac{v}{\sqrt{||v||^2+\varepsilon^2}} \tag{4}$$

K-NN

Classification determines the class of a new object based on a previously designed class model. There is a learning process based on the characteristics of a training data [21]. So from this learning will produce a pattern that can be used in the classification process of the test data. This process is commonly referred to as supervised learning [22]–[24]. Classification itself has several techniques based on distance, statistics, neural networks, decision trees and rules. One example of a classification method based on distance is K-Nearest Neighbor (K-NN).

The K-Nearest Neighbor method is a classification algorithm that groups test data with the closest resemblance or distance to the learning data. So if a test data has the closest majority resemblance to a certain class among other classes, then the test data has the same label or category as that clasS. The advantages of K-NN itself are easy to understand because it is simple, effective for large amounts of data and many classes, resistant to noise in training data, and the training process is fast. This classification method has steps that can be described in Figure 3, with stages :

1) Determine the value of K.
2) Calculate the distance of the test data to all training data.
3) Label the test data closest to the training data based on the dominant class label.

Generaly, the value of K that is commonly used is an odd number (1,3,5,7...n). This is to avoid classifying the closest distance on the test data with the same number of class labels. Meanwhile, to calculate the proximity value, there are several techniques such as Euclidean, Manhattan, Minkowski and Chebychev. However, the most commonly used technique is Euclidean, because this technique is one of the distance calculations that use attributes of numeric type. The following equation formula in the Euclidean technique in (5) while *x = (x11, x12, ... , x1n)* and *y = (y11, y22, ... , y2n).*

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_{1,i} - y_{1,i})^2} \tag{5}$$

Where 'x' is the training data set and 'y' is the set of test data. Meanwhile, dis(x,y) is the calculation of the shortest distance between training data (x) and test data (y) based on dimension size (n) a number of data sequences (i). Training data is used as a classification model whose data has undergone a learning process and already has a clear class label as knowledge to predict new data. Meanwhile, test data is used to measure the performance or performance of the classifier's success in carrying out the classification results.

Data Collection

Several people with the same classification wrote samples of Javanese script handwritten data. They are selecting respondents to write Javanese characters on white HVS paper using a black ballpoint pen. Javanese script is written sequentially according to the example with a total of 20 Javanese characters. Each respondent writes each letter of the Javanese script as many as 20 characters. The results of the respondent's writing are then captured into an image in .JPG format using a scanner.

Proposed Method

Research on the use of the concept of K-Nearest Neighbor (K-NN) and feature extraction of Histogram of Gradient (HoG) for handwritten Javanese letter recognition can be seen in Figure 3.

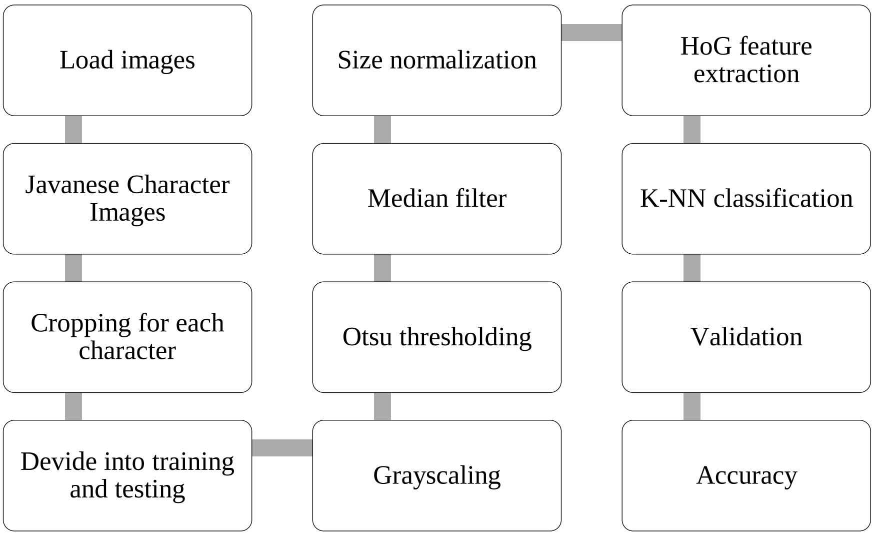


Figure 3. Proposed Method for Classify Javanese Character

Based on Figure 3, we made some indicators in HoG such as cell size, block, and num bins. Whereas, in K-NN we had been test for each K value (1,3,5,7,9,11,13,15) using Euclidean distances.

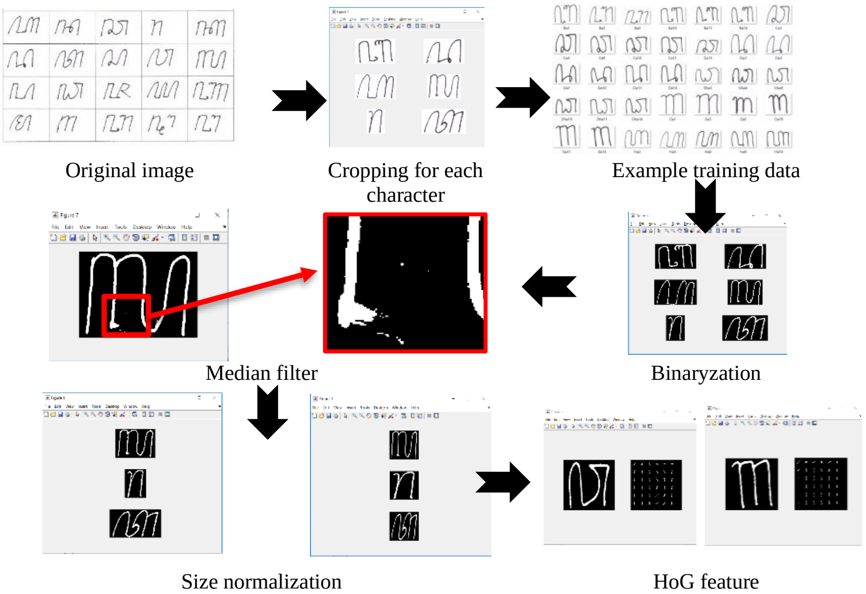
RESULT AND DISCUSSION

The first stage, preparing data in the form of handwritten Javanese script images. In this study, data were acquired using a scanner and stored in .JPG file format. The second stage is cropping the image of the handwritten Javanese script from the scanner containing 20 Javanese script letters. Where cropping is done to produce images per letter with various sizes as shown in Figure 4. The third stage is to group the cropped images into different folders. The first folder is the training data folder which contains 120 images as training/learning images. The second folder is the test data folder which contains 40 images for testing. The fourth stage, preprocessing all images that have been previously grouped into folders. The technique used at this stage can affect the subsequent process and the final stage.

The preprocessing stage includes:

1) Grayscaling to change the intensity of the RGB image into a single intensity or the image only has color based on the degree of gray level.
2) Binarization to change the color of the image to black and white. This process is done by determining the threshold value. Where the black pixel '0' as the background and the white pixel '1' indicates the character of the Javanese script. In this process, Otsu's thresholding will be used to produce a black and white image because this technique can determine the threshold value automatically [8].
3) Median filter for image enhancement, such as smoothing the image and removing noise at the same time [27]. This filter works by finding the middle value in the NxN pixel size that has been sorted in ascending order. After the middle value is obtained, the middle value will replace the center pixel value.
4) Size Normalization changes the image size on the various training and test data to the same size, 130x128 pixels.

The fifth step is to take or extract the features of each Javanese letter image using the Histogram of Gradient (HoG) using cell sizes, namely [20 19], block [2 2] and numbins is 9. The following is an example of the HoG feature on Javanese characters with the parameters used. The sixth stage is to recognize the test data using the K-NN classification method based on the value obtained from the feature extraction process. At this stage, the process of comparing a test data with all data on the training data is carried out which has a closeness value. The process of finding the value of the proximity using the Euclidean distance. After comparing the closest values, the closest value will be chosen according to the desired number of K. From the selected set of data values, the dominant class will be determined. So if a test data has a majority proximity to a certain class among other classes, then the test data has the same label or category as that class. The last stage is to evaluate using a confusion matrix. At this stage the aim is to calculate how much test data is recognized correctly and which is recognized incorrectly. So that it can be analyzed how accurate the method used in recognizing Javanese letters is. All of the stages has been shown in Figure 4.



Original image

Cropping for each character

Example training data

Median filter

Binaryzation

Size normalization

HoG feature

Figure 4. Copping stage

Tabel 1. K-NN results (%)

Dataset ratio	K values	K-NN	Median Filter – K-NN	Median Filter – HoG – K-NN
60:40	1	94	96	96.75
	3	93.25	95	96.25
	5	91.5	93.75	93
	7	89	92.64	89.25
	9	88	86	88
	11	82	78.25	84
	13	74.5	77	81
	15	71	73	78
70:30	1	95	96.5	97
	3	93.5	95	97.75
	5	91	92	94
	7	86	87.25	91
	9	85	86	89
	11	79.75	82	84.25
	13	74.25	80	82.25
	15	70	79	80
80:20	1	96	97	98.5
	3	93	93	96
	5	92.5	92	94.25
	7	90	88	91.25
	9	84.75	84	89
	11	80	79.25	86
	13	77	76.5	83.25
	15	73.75	74	80.75

Based on Table 1, it is known that the image has been classified using KNN through a combination of several preprocessing and feature extraction. The purpose of using the combination is to know the performance of K-NN. The value of K tested is quite long, namely from K = 1 to k = 15. The K value obtained from all experiments yielded an accuracy above 70%. HOG accuracy was higher than KNN alone without model optimization and KNN with median filter. The median filter is used to sharpen the image, while HoG is used to represent the image's shape. In essence, using the median filter – HoG and KNN together can result in higher accuracy. The highest KNN value is 96%, while the median filter – KNN produces a slightly higher accuracy of 97. KNN obtains the highest accuracy with the combination of the median filter and HoG at 98.5%.

On the other hand, we also analyze the accuracy to ratio dataset. We use 3 ratios, namely 40%, 30%, and 20% with a total data of 1000 Javanese script images. The image ratio of 60:40 produces the lowest accuracy compared to others. K values in each ratio tend to be high at low K values. The median filter – HoG – KNN still occupies the highest accuracy gain in small and large ratios. From all data, it is known that the highest accuracy is at K=1 from the combination of the median filter – HoG – KNN with 98.5%.

CONCLUSION

The process of classifying the image of Javanese characters using KNN has been completed in the analysis. The overall image is the image of the Javanese script, which is 1000 handwritten images from 50 respondents. KNN alone has produced accuracy between 73% to 96% on K values from 1 to 15. To increase accuracy, median filtering and Histogram of Gradient (HoG) were used in this study. Accuracy results are known to increase between 1 and 4% higher using selected preprocessing and feature extraction. In this study, we do not look at the optimal aspect of the algorithm based on travel time, but only based on accuracy. In future research, analysis based on travel time, recall, precision can be implemented to obtain results from other points of view.

REFERENCES

[1] I. Prihandi, I. Ranggadara, S. Dwiasnati, Y. S. Sari, and Suhendra, “Implementation of backpropagation method for identified javanese scripts,” *J. Phys. Conf. Ser.*, vol. 1477, no. 3, 2020.

[2] 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,” *Procedia Comput. Sci.*, vol. 157, pp. 499–505, 2019.

[3] S. Das and S. Banerjee, “An algorithm for japanese character recognition,” *Int. J. Image, Graph. Signal Process.*, vol. 7, no. 1, pp. 9–15, 2014.

[4] A. Susanto, D. Sinaga, C. A. Sari, E. H. Rachmawanto, and D. R. I. M. Setiadi, “A high performace of local binary pattern on classify javanese character classification,” *Sci. J. Informatics*, vol. 5, no. 1, p. 8, 2018.

[5] T. A. Zuraiyah and C. R. Pitoyo, “Optimization of feature extraction using combined image centroid zone and zone centroid zone method,” in *IORA International Conference on Operation Research*, 2016, pp. 27–30.

[6] C. K. Dewa, A. L. Fadhilah, and A. Afiahayati, “Convolutional neural networks for handwritten javanese character recognition,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 12, no. 1, p. 83, 2018.

[7] G. S. Budhi and R. Adipranata, “Handwritten Javanese character recognition using several artificial neural network methods,” *J. ICT Res. Appl.*, vol. 8, no. 3, pp. 195–212, 2015.

[8] G. H. Wibowo, R. Sigit, and A. Barakbah, “Javanese character feature extraction based on shape energy,” *Emit. Int. J. Eng. Technol.*, vol. 5, no. 1, pp. 154–170, 2017.

[9] L. D. Krisnawati and A. W. Mahastama, “Building classifier models for on-off javanese character recognition,” in *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, 2019, pp. 25–34.

[10] M. Rashad and N. A. Semary, “Using KNN and random forest tree classifiers,” Springer Int. Publ. Switz., pp. 11–17, 2014.

[11] Y. Sugianela and N. Suciati, “Javanese document image recognition using multiclass support vector machine,” *CommIT (Communication Inf. Technol. J.*, vol. 13, no. 1, p. 25, May 2019.

[12] V. Ong and D. Suhartono, “Using K-Nearest Neighbor in optical character recognition,” *ComTech Comput. Math. Eng. Appl.*, vol. 7, no. 1, p. 53, 2016.

[13] M. S. Kadhm, “Handwriting word recognition based on SVM classifier,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 11, pp. 64–68, 2015.

[14] G. Katiyar, A. Katiyar, and S. Mehfuz, “Offline handwritten character recognition system using support vector machine,” *Am. J. Neural Networks Appl.*, vol. 3, no. 2, pp. 22–28, 2017.

[15] H. N. K. Al-behadili, “Classification algorithms for determining handwritten digit,” *Iraq J. Electr. Electron. Eng.*, vol. 12, no. 1, pp. 96–102, 2016.

[16] P. M. Kamble and R. S. Hegadi, “Geometrical features extraction and knn based classification of handwritten marathi characters,” *Proc. - 2nd World Congr. Comput. Commun. Technol. WCCCT*

2017, pp. 219–222, 2017.

[18] M. Sudarma and I. W. A. Surya Darma, “The identification of balinese scripts characters based on semantic feature and k-Nearest Neighbor,” *Int. J. Comput. Appl.*, vol. 91, no. 1, pp. 14–18, 2014.

[19] R. T. Mojtaba Mohammadpoor, “Recognition of persian handwritten numbers using LBP-HOG descriptor,” *Bull. Environ. Pharmacol. Life Sci.*, vol. 4, no. 1, pp. 284–290, 2015.

[20] R. Kaur and Priyadarshni, “Serial number recognition in banknotes using HoG feature extraction and KNN classification,” *IOSR J. Comput. Eng.*, vol. 18, no. 6, pp. 41–49, 2016.

[21] A. Lamba and D. Kumar, “Survey on KNN and Its Variants,” *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 5, no. 5, pp. 430–435, 2016.

[22] R. Muralidharan and C. Chandrasekar, “Object Recognition using SVM-KNN based on geometric moment invariant,” *Int. J. Comput. Trends Technol. to Aug Issue*, no. September 2011, pp. 215–220, 2011.

[23] Z. E. Rasjid and R. Setiawan, “Performance comparison and optimization of text document classification using k-NN and Naïve Bayes classification techniques,” *Procedia Comput. Sci.*, vol. 116, pp. 107–112, 2017.

[24] M. Sholihin, “Classification of batik lamongan based on features of color, texture and shape,” *Kursor*, vol. 9, no. 1, pp. 25–32, 2018.