Jurnal ELTIKOM :
Jurnal Teknik Elektro, Teknologi Informasi dan Komputer

Vol. 7, Issue 1, June 2023, page. 29-37
ISSN 2598-3245 (Print), ISSN 2598-3288 (Online)
DOI: http://doi.org/10.31961/eltikom.v7i1.618
Available online at http://eltikom.poliban.ac.id

SIGNATURE IDENTIFICATION USING DIGITAL IMAGE PROCESSING AND MACHINE LEARNING METHODS

I Kadek Nurcahyo Putra*, Ni Putu Dita Ariani Sukma Dewi, Diah Ayu Pusparani, Dibi Ngabe Mupu
Department of Computer Science, Universitas Pendidikan Ganesha, Singaraja, Indonesia
e-mail: ikadeknurcahyoputra@gmail.com, dita.ariani.sukma@undiksha.ac.id, diah.ayu.pusparani@undiksha.ac.id, dibi@undiksha.ac.id

Received: October 12, 2022 – Revised: December 6, 2022 – Accepted: December 7, 2022

ABSTRACT

Signature is used to legally approve an agreement, treaty, and state administrative activities. Identification of the signature is required to ensure ownership of a signature and to prevent things like forgery from happening to the owner of the signature. In this study, data signatures were obtained from 25 people over the age of 50. The signers provided 20 signatures and were free to choose the stationery used to write the signature on white paper. The total data obtained in this study was 500 signature data. The obtained signature was scanned to create a signature image, which was then pre-processed to prepare it for feature extraction, which can characterize the signature images. The HOG method was used to extract features, resulting in a dataset with 4,536 feature vectors for each signature image. To identify the signature image, the classification methods SVM, Decision Tree, Nave Bayes, and K-NN were compared. SVM achieved the highest accuracy, which is 100%. When K=5, the K-NN method achieved a fairly good accuracy of 97.3%. Meanwhile, Naive Bayes and Decision Tree achieved accuracy significantly lower than K-NN (61%). Because the HOG method produced a large feature vector for each signature, it is recommended that important features that represent signatures be optimized or extracted to produce smaller features to speed up computation without sacrificing accuracy, and that the HOG method be compared to other extraction feature methods to obtain a better model in future research.

Keywords*: HOG, KNN, Naive Bayes, Signature, SVM.*

I. INTRODUCTION

SIGNATURES are generally used to legally approve an agreement, contract, and state administrative activities [1]. The signature must be confirmed as authentic at the agreement ratification. To avoid undesirable outcomes such as forgery, it is critical to recognize signatures. The similarity of the valid signature with the newly written signature can be used to recognize the signature [2],[3].

Signatures are generally identified manually by humans by directly comparing a valid signature pattern with a signature written at the time [4]. Frequently written signatures will be identical to each other, but not always the same. These changes can be influenced by the position of the writing, size, and writing tools used. Age and mental condition of a person also affect these changes [5]. A person can be classified as elderly when he reaches the age of 50 years. Changes that occur in the elderly include physiological and motor skills [6].

Signature identification can now be done not only manually, but also by computer, thanks to recent technological advancements. Computers identify things objectively, allowing them to assist humans in making decisions, whereas humans can be subjective at times. However, the computer cannot directly identify objects, in this case signatures, but must first go through some pattern recognition processes, which can be accomplished by extracting signature features [5].

There have been several previous studies that explain signature identification. The first is a study from [7] which identified the signatures of 15 (fifteen) people with 150 signature datasets. They extract texture features in signature images using the Gray Level Co-Occurrence Matrix (GLCM) and Local Binary

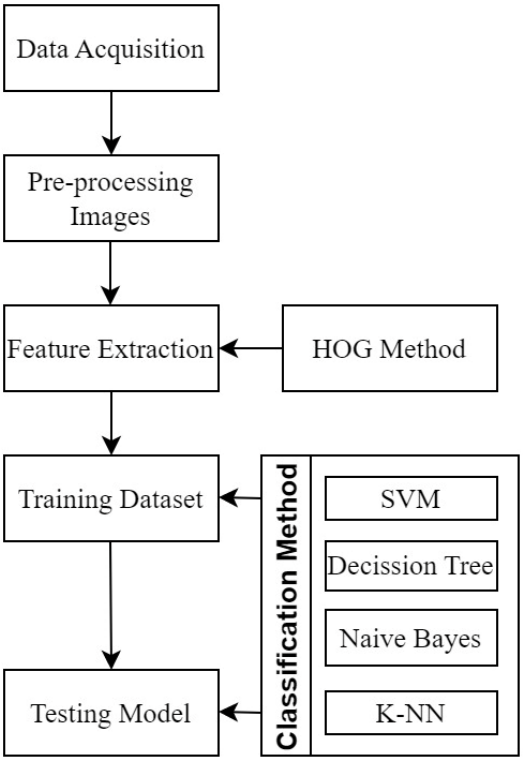

Figure 1. Research Flow Diagram

Pattern (LBP) methods. The Support Machine (SVM) classification method is used to build a classification model. The test results show that the dataset extracted using the GLCM method obtained an accuracy of 86.67% outperforming the dataset produced by feature extraction using the LBP method, which is 80%.

The second is a study from [8] using artificial neural network backpropagation to perform signature identification. They use 50 nodes, 1 hidden layer, learning rate 0.3 after doing various experiments to get maximum results. 150 out of 10 respondents obtained 95% accuracy. From the use of HOG as a signature feature extraction method combined with the classification method of artificial neural networks in [9], an accuracy of 98.33% was obtained.

This study collected signature data from people who had passed the age of 50. The dataset is built by extracting signature features using the Histogram of Oriented Gradient method, then the dataset will be divided into training data and test data. Several Machine Learning methods such as Support Vector Machine (SVM) which in [7] are combined with the GLCM feature extraction method get the highest accuracy in identifying signatures. This study combines SVM with HOG feature extraction in the hope of getting an increase in accuracy. Several machine learning methods such as K-Nearest Neighbors, Naive Bayes, and Decision Tree are also used to compare SVM models to obtain a good classification method for recognizing signatures.

II. RESEARCH METHOD

This section describes the HOG method, the classifier method used to identify signature images, and the study's research flow. The research method in this study included four stages: data acquisition, feature extraction, training, and testing. Figure 1 shows the flow of this research, beginning with the data collection process from 25 people over the age of 50, and ending with the signatures obtained being scanned to produce signature images. The signature image was pre-processed so that the HOG method could extract features from it. The feature extraction dataset was divided into two parts: 70% of the data was used for training to build a classification model, and the remaining 30% was used to test the classification model built with various classification methods such as SVM, Decision Tree, Naive Bayes, and K-NN to find the best model in identifying signatures in this study.

TABLE 6
HOG BINS STORAGE

0	20	40	60	80	100	120	140	160

A. Histogram of Oriented Gradient

Histogram of Oriented Gradient (HOG) is a feature descriptor based on edges and directions (orientation) widely used in image processing and computer vision. HOG is often used in facial recognition, animals, vehicle image detection, handwriting, and others [10]. HOG feature extraction is done by cal culating the gradient orientation in an area localized in the image [11],[12].

The formation of HOG features in each cell was carried out by accumulating gradient magnitudes that had the same direction orientation. The gradient directions were grouped into sections called bins. In this study, 8x8 cells were used and the gradient direction was grouped into 9 bins which [13] stated that the number of bins was 9 which provided more optimal detection results.

The HOG method starts by computing the gradient on the object's horizontal and vertical axes (x,y) [14]. In an 8-bit grayscale image (0-255), for example, the horizontal gradient calculation (x-axis) moves from left to right, from the image's boundary (background) to the right to meet an object with a lame pixel intensity value (large magnitude difference) from a small value to a large value (positive gradient), and then moves straight until it reaches the back-ground (negative gradient) as Equation 1. In the vertical calculation (y), the gradient moves from top to bottom.

$$Gradient\ (x, y) = |positive\ gradient - negative\ gradient| \tag{1}$$

The horizontal and vertical gradient calculations are combined, then the magnitude of the gradient and direction (orientation) are calculated by the Equation 2 and 3.

$$Gradient\ Magnitude = \sqrt{x^2 + y^2} \tag{2}$$

$$Gradient\ Direction = arc\ tan\frac{y}{x} \tag{3}$$

To calculate a histogram of oriented gradients (HOG), segment the image into smaller cells and calculate the gradient magnitude and gradient direction. In this study we used 8x8 cells. The results of the calculation of the magnitude and direction of the gradient are stored in the specified 9 bin as shown in Table 1.

If a certain pixel gives a gradient direction of 30°, then the gradient magnitude value will be divided and stored in bins 20° and 40°. Adding the magnitude gradients in each bin yields 9 feature vectors in each cell.

B. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a technique in machine learning to make predictions, both in the case of classification and regression [15]. SVM has the basic principle of a linear classifier, namely classification cases that can be linearly separated. However, SVM has been developed to work on nonlinear problems by incorporating kernel concepts into high-dimensional workspaces. In this high-dimensional space, a hyperplane will be sought that can maximize the distance (margin) between data classes [16].

The SVM concept can be explained simply as an attempt to determine the best hyperplane value that serves as a separator between two classes in the input space. SVM builds a hyperplane on a multidimensional space to separate different classes. SVM generates the optimal hyperplane iteratively which is used to minimize errors. The core idea of SVM is to find the Maximum Marginal Hyperplane (MMH) value that divides the dataset the most into several classes [17].

The main purpose of SVM is to share a given dataset in the most possible way. The distance between the two closest points is called the margin. The goal is to select the hyperplane with the largest possible margin between the support vectors in the given data set.

C. Decision Tree C4.5 Classifier

The C4.5 Decision Tree Classification Method converts data into trees with rules that influence predictive decisions. A decision tree is made up of three types of nodes: the root, which serves as the decision tree's starting point, the branch, which contains the classification question, and the leaf, which contains the decision tree's final decision or target class. The formation of a decision tree begins with calculating the dataset's Entropy to determine how informative a node is (Equation 4). Based on Equation 4, *S* is dataset, *k* is classes, and *Pj* is feature probability. Next, calculate the gain of each feature and use the highest gain of the feature as the root of the decision tree using Equation 5 where *Entropy (S)* is entropy dataset, *A* is feature, *k* is class, |*Si*| is Proportion of *Si* to *S*, and |*S*| is number of cases in *S*.

$$Entropy(S) = \sum_{j=1}^{k} - pj\ log2\ pj \tag{4}$$

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{k} \frac{|Si|}{|S|}\ x\ Entropi(Si) \tag{5}$$

D. Naïve Bayes Classifier

The Naive Bayes Classification Method is a machine learning method that utilizes probability and statistical calculations proposed by the British scientist Thomas Bayes, which predicts future probabilities based on previous experience [18].

The Naive Bayes algorithm assumes that the effect of a feature value on the class label is not related to the values of other features. This assumption is called a class of free/unbound conditions. Bayes provides learning methods based on existing evidence. The algorithm studies the available evidence by calculating the correlation between the desired variable and all other variables [19].

Naive Bayes is stated as the Maximum A Posteriori hypothesis. For example, if there are several alternative hypotheses h, then the hypothesis that has the highest probability will be sought when a set of evidence appears (Max Probability) [19]. Naïve bayes is calculated using Equation 6 where *P*(*H*|*E*) is the probability that hypothesis *H* occurs when evidence *E* appears, *P*(*E*|*H*) is the probability of evidence *E* that will affect the hypothesis *H*, *P*(*H*) is the initial probability of *H*, and *P*(*E*) is the initial probability of *E*.

$$P(H|E) = \frac{P(E|H) \times P(H)}{P(E)} \tag{6}$$

E. K-Nearest Neighbors Classifier

KNN algorithm is the simplest algorithm among any machine algorithms [7]. KNN is included in supervised learning which is used for the classification of new objects based on their closest objects. The results of the new instance query will be classified based on the most number or the majority of the categories in the KNN. It can also be interpreted that the most frequent class will be used as a classification class [8],[20].

Here are the steps to perform classification using the K-Nearest Neighbor (KNN) algorithm [11]. First, specify the *K* parameter according to used data. The minimum *K* value is 1 and the maximum value is amount of training data. Second, calculate the distance between test data and training data. To calculate the distance in the calculation of the KNN algorithm, use Euclidean distance with the Equation 7 where *pi* is data train, *qi* is data test, *i* is data variable, and *n* is data dimension.

$$Euclidean = \sqrt{\sum_{i=1}^{n} (pi - qi)^2} \tag{7}$$

Third, the distance is sorted from the largest to the smallest. Fourth, determining the closest distance up to the *K* parameter. Fifth, pair the appropriate classes.

F. Data Acquisition

Signature data was obtained from people over the age of 50. When signing on white paper, signers could use any writing utensil they wanted. Signers were asked to write 20 signatures, with each paper containing four signatures. Following receipt of the signature data, it was scanned and saved in jpg format.

G. Feature Extraction

Scanned signature images with four signatures in one image were cropped so that each image only had one signature. Cropped images were labeled in the Alphabet_No format, such as A_01. This represents A's first signature. The alphabet was used to replace the signer's name in this study.

Images were resized to 80*120 pixels to ensure that they were all the same size. The original RGB image was then converted to a gray level image in order to perform feature extraction using the HOG method. The feature extraction process produced a dataset that was saved in csv format.

H. Training Data

The dataset was broken down into two parts, namely training data and testing data with a ratio of 70:30. Data training was used to train data and build classification models. Machine Learning models can later identify new data.

I. Testing

The test data was used to validate the classification model developed with the training data in the previous step. The confusion matrix was used to assess the classification model's performance.

III. RESULT AND DISCUSSION

A. Data Acquisition

This study collected 500 data points from 25 people over the age of 50. The signer either uses their own pen or uses stationery provided by the researcher. On white paper, each person signs 20 times. During the data collection process, it was interesting to note that almost all signers paused to write their signature on the next piece of paper. The signature data on the paper is scanned and saved in color and jpg format. Figure 2 depicts the data acquisition results.

B. Feature Extraction

Researchers carry out several processes before performing feature extraction on signature images. Figure 3 shows a signature image that has been cut and given the name of each class. The image is cropped so that one image contains one signature. Each signature image is labeled according to a predefined format. The labeled image is cropped to minimize the blank in the background, taking only the signature object. Figure 5 shows the results of resizing the original image, each image being resized to be uniform. The next step is to convert the original RGB image to a gray level so that the image is ready for feature extraction using the HOG method. The results of converting the RGB image to a grayscale image are shown in Figure 6.

Figure 7 shows an illustration of the HOG method of visualizing the direction of signature strokes that distinguishes one person's signature from another. Table 2 describes the number of feature vectors in each signature image. The 9 values in each cell represent 9 gradient magnitude values in the HOG feature storage. Each cell generates 9 feature vectors, using a 2*2 cell block, the total HOG feature extraction results are 9*14*36 = 4,536 feature vectors for each signature image.

C. Training Data

The feature extraction dataset is divided into training data and testing data. 70% of the data from the dataset (350 signature images) are used as training data to build a classification model. This study uses C = 10 and rbf kernel to build SVM classification model. The K-NN model uses Euclidean distances to determine data proximity, the Naive Bayes Algorithm calculates the average value and deviation of training data to calculate feature probabilities with the normal distribution formula, and Decision tree

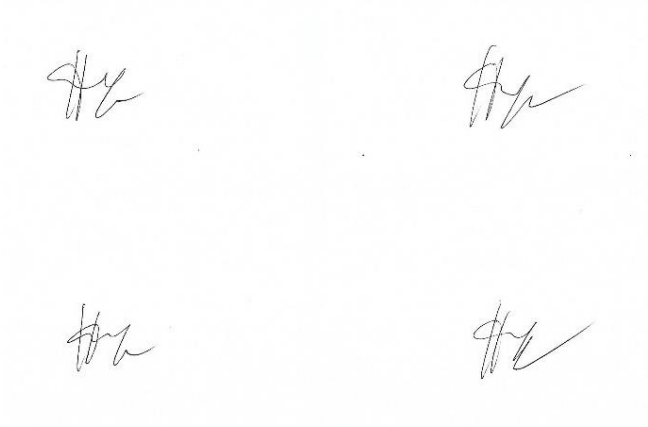

Figure 2. Raw Signature Data

Figure 3. Signature Images Data

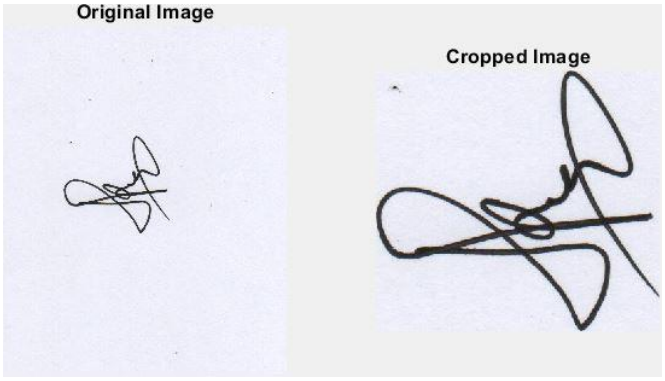

Figure 4. Cropping Image

build builds a classification model from calculating data entropy and obtaining information from each feature.

D. Testing Models

The classification model that has been built uses 350 training data, then is tested with 150 data or 30% of the data from the feature extraction dataset. Naive Bayes, Decision Tree, SVM, and K-NN are compared to classify signature images to get a machine learning model that gives the best performance.

TABLE 2
ILLUSTRATION RESULT OF FEATURE EXTRACTION

9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

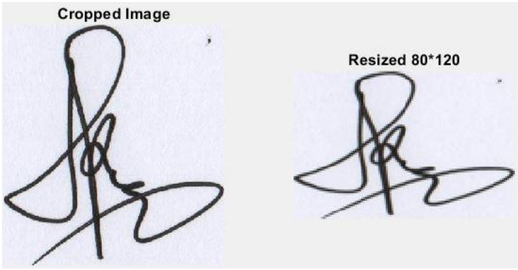

Figure 5 Resizing Signature Image

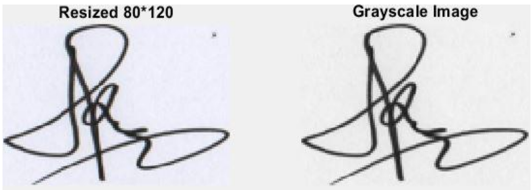

Figure 6 Grayscaling Signature Image

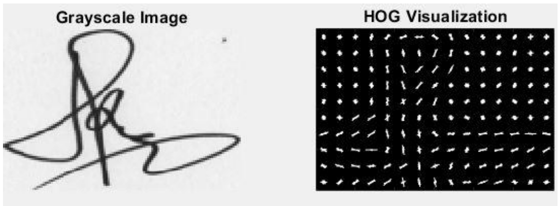

Figure 7 HOG Method Visualization

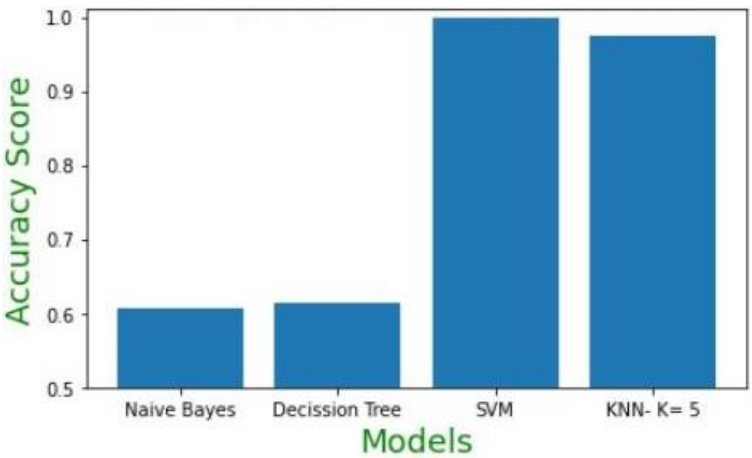

Figure 8 Accuracy Comparison between Four Model Classification

Accuracy which represents the percentage of an algorithm that can correctly predict test data based on actual data is used to compare models. Figure 8 shows the results of the comparison of the accuracy of the four classification methods used in this study.

Figure 8 shows that SVM obtained perfect accuracy in this study, namely 100%. These results indicate that the SVM classification method combined with the HOG feature extraction method yields better results than research [7] which used the GLCM and LBP feature extraction methods. The K-NN Classifier using the parameter K=5 obtains an accuracy value of 97.3% which indicates that the solid HOG extraction method is used in extracting signature features. Meanwhile, the accuracy obtained using the Decision Tree C4.5 and Naive Bayes methods was quite far below K-NN, namely 61.3% and 60.6% respectively. The different accuracy results between SVM, Decision Tree, and Naive Bayes can occur because the Decision Tree classification model is built from training data and forms a tree that has patent rules. Therefore, when new data is entered it will immediately follow the rules. Numerical data from the signature image consists of more than 4000 feature vectors, and there is a possibility that feature values outside the tree rules may be classified incorrectly. Naive Bayes has a weakness when there is a feature value of 0 then the probability will be 0, so it is not suitable for data that has large feature vectors.

IV. CONCLUSION

Four classification methods have been successfully applied to classify signature images in this study. The SVM Classification Method which was built using the Rbf Kernel and a value of C = 10 obtained the highest accuracy in this study, namely 100%. K-NN Classifier obtains a fairly good accuracy of 97.3% using the K=5 parameter. Decision Tree C4.5 obtained an accuracy of 61.3%, and the Naive Bayes algorithm obtained the lowest accuracy in this study, namely 60.6%. In this study, it can be concluded that SVM and K-NN combined with the HOG feature extraction method are good for identifying signatures. The relatively high accuracy indicates that the dataset resulting from feature extraction using the Histogram of Oriented Gradient method can accurately characterize each different person's signature. Because the HOG method generates 4,536 feature vectors for each signature image, more research is needed to optimize or extract important features that represent signatures in order to produce smaller features that can be computed faster without sacrificing accuracy. Because signatures are commonly used when approving an agreement, agreement, and state administration activities, speed and accuracy in identifying and distinguishing one person's signature from another is critical.

REFERENCES

[1] B. C. Octariadi, "Pengenalan Pola Tanda Tangan Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation," J. Teknoinfo, vol. 14, no. 1, pp. 15–21, 2020.

[2] G. Novandra, M. Z. Nafâ, and T. G. Laksana, "Perancangan aplikasi android identifikasi tanda tangan menggunakan multi layer perceptron," JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform., vol. 3, no. 1, 2018.

[3] J. Arifin and M. Z. Naf'an, "Verifikasi Tanda Tangan Asli Atau Palsu Berdasarkan Sifat Keacakan (Entropi)," J. Infotel, vol. 9, no. 1, pp. 130–135, 2017.

[4] M. S. Simanjuntak, R. Rosnelly, and W. Wanayumini, "Identifikasi Tanda Tangan menggunakan Metode Fitur Ekstrasi Biner dan K Nearest Neighbor," CSRID (Computer Sci. Res. Its Dev. Journal), vol. 12, no. 3, pp. 191–200, 2021.

[5] W. Fitriani, M. Z. Naf'an, and E. Usada, "Ekstraksi Fitur pada Citra Tanda Tangan Sebagai Ciri Identitas Pemiliknya Menggunakan Discrete Fourier Transform," 2018.

[6] A. Afrizal, "Permasalahan Yang Dialami Lansia Dalam Menyesuaikan Diri Terhadap Penguasaan Tugas-Tugas Perkembangannya," Islam. Couns. J. Bimbing. dan Konseling Islam, vol. 2, no. 2, pp. 91–106, 2018.

[7] P. Mudjirahardjo and A. Basuki, "Identifikasi Tanda Tangan dengan Ekstraksi Ciri GLCM dan LBP," J. EECCIS (Electrics, Electron. Commun. Control. Informatics, Syst., vol. 13, no. 1, pp. 6–10, 2019.

[8] A. Hidayatno, R. R. Isnanto, and D. K. W. Buana, "Identifikasi Tanda-Tangan Menggunakan Jaringan Saraf Tiruan Perambatan-Balik (Backpropagation)," J. Teknol., vol. 1, no. 2, pp. 100–106, 2008.

[9] M. Taşkiran and Z. G. Çam, "Offline signature identification via HOG features and artificial neural networks," in 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 2017, pp. 83–86.

[10] Y. Sugianela and N. Suciati, "Ekstraksi fitur pada pengenalan karakter Aksara Jawa berbasis Histogram of Oriented Gradient," JUTI J. Ilm. Teknol. Inf., vol. 17, no. 1, pp. 64–72, 2019.

[11] H. Patel, S. Desai, P. Desai, and A. Damani, "Review on Offline Signature Recognition and Verification Techniques," Int. J. Comput. Appl., vol. 179, no. 53, pp. 35–41, 2018.

[12] A. Singh, "Feature engineering for images: a valuable introduction to the HOG feature descriptor," Mediu. Anal. Vidhya, vol. 4, 2019.

[13] T. Kobayashi, A. Hidaka, and T. Kurita, "Selection of histograms of oriented gradients features for pedestrian detection," in Neural Information Processing: 14th International Conference, ICONIP 2007, Kitakyushu, Japan, November 13-16, 2007, Revised Selected Papers, Part II 14, 2008, pp. 598–607.

[14] R. Y. Endra, A. Cucus, F. N. Afandi, and M. B. Syahputra, "Deteksi Objek Menggunakan Histogram Of Oriented Gradient (Hog) Untuk Model Smart Room," Explor. J. Sist. Inf. dan Telemat. (Telekomunikasi, Multimed. dan Inform., vol. 9, no. 2, 2018.

[15] F. S. Jumeilah, "Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian," J. RESTI (Rekayasa Sist. Dan Teknol. Informasi), vol. 1, no. 1, pp. 19–25, 2017.

[16] I. C. R. Drajana, "Metode support vector machine dan forward selection prediksi pembayaran pembelian bahan baku kopra," Ilk. J. Ilm., vol. 9, no. 2, pp. 116–123, 2017.

[17] L. Priyambodo et al., "Klasifikasi Kematangan Tanaman Hidroponik Pakcoy Menggunakan Metode SVM," J. RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 6, no. 1, pp. 153–160, 2022.

[18] M. M. Saritas and A. Yasar, "Performance analysis of ANN and Naive Bayes classification algorithm for data classification," Int. J. Intell. Syst. Appl. Eng., vol. 7, no. 2, pp. 88–91, 2019.

[19] T. Wahyono, Fundamental of Python for Machine Learning: Dasar-Dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan. Yogyakarta: Gava Media, 2018.

[20] S. Ibrahim and N. A. N. Samlan, "Histogram of oriented gradient (HOG) for off-line handwritten signature authentication," Int. J. Emerg. Trends Eng. Res., vol. 8, no. 1 1.1 Special Issue, pp. 102–107, 2020, doi: 10.30534/ijeter/2020/1681.12020.