RSF Conference Proceeding Series: Engineering and Technology Vol. 4 No. 1 (2025) https://doi.org/10.31098/cset.v4i1.1028

Research Paper

Classification Of Merapi Volcano Images Based on HSV Color Feature Extraction and Local Binary Pattern Texture Feature Extraction Using The K-Nearest Neighbors Method

Awang Hendrianto Pratomo[1]*, Prize Isnan Khairi Attamimi [1], Agus Budi Santoso[2], Eko Teguh Paripurno[1], Johan Danu Prasetya[1], Mohd Sanusi Azmi[3]

[1] Universitas Pembangunan Nasional "Veteran" Yogyakarta, Indonesia
[2] Balai Penyelidikan dan Pengembangan Teknologi Kebencanaan Geologi (BPPTKG), Badan Geologi, Kementerian Energi dan Sumber Daya Mineral, Indonesia
[3] Universiti Teknikal Malaysia Melaka (UTeM), Malaysia

Received :	Revised :	Accepted :	Online : Oct 14, 2025

Abstract

The BPPTKG (Center for Volcanology and Geological Hazard Mitigation) routinely monitors Merapi Volcano's activity through visual imagery captured with DSLR lenses at several observation posts. However, not all recorded imagery can be used for analysis due to frequent cloud or fog cover. This not only makes it difficult for experts to accurately monitor Merapi's condition but also reduces the efficiency of data storage capacity. To examine the application of HSV color feature extraction, LBP texture feature extraction, and the K-Nearest Neighbor method for classifying Merapi Volcano images based on appearance. The dataset used consists of Merapi Volcano images captured from the Tunggularum observation post between October 1st and 10th, 2023, categorized into six classes based on the volcano's appearance. Preprocessing steps include cropping, masking, and image sharpening. Classification was performed using the K-Nearest Neighbor method to obtain the classification results of Mount Merapi images. Based our result, the classification method using HSV and LBP using the K-Nearest Neighbor method was successfully performed. The optimal value of k was 1, achieving an accuracy of 95%, while the worst value of k was 9, with an accuracy of 87%.

Keywords: *Image Processing, K-Nearest Neighbor, HSV, LBP, Mount Merapi*

INTRODUCTION

Mount Merapi (2968 m above sea level) is one of the most active volcanoes in Indonesia, attracting significant attention from various groups due to its frequent volcanic activity as well as its uniqueness from scientific and cultural perspectives [1]. The activity of Mount Merapi is routinely monitored by the Volcanology and Geological Disaster Mitigation Center (BPPTKG), a unit under the Geological Disaster Mitigation and Volcanology Center (PVMBG). The monitoring is done through visual images captured by DSLR cameras placed at several observation posts, one of which is the Tunggularum Post. These images are transmitted every minute to the BPPTKG monitoring room to observe the current condition of Mount Merapi.

However, not all images produced can be used for analysis, as most are obscured by clouds or mist. This situation leads to the accumulation of irrelevant data, which requires a manual process to filter and delete non-informative images. This not only complicates the work for experts but also impacts the efficiency of data storage capacity. Therefore, the development of an automatic system is needed to classify the clarity of Mount Merapi images so that only relevant images are stored, while those that do not meet the criteria can be automatically deleted.

Corresponding author's email: awang@upnyk.ac.id:

LITERATURE REVIEW

Afif (2020) conducted research related to image classification of Mount Merapi by comparing the Support Vector Machine and K-Nearest Neighbor methods using two feature selection methods: Correlation-based Feature Selection and Chi-Square, resulting in an accuracy of 90% [2]. Digital images are visual representations of an object consisting of pixels with certain intensity values [3]. Image processing has been widely used in research, one of which involves morphological operations to enhance object detection [4][5]. Preprocessing steps such as cropping, masking, and image sharpening aim to improve the quality of images before feature extraction [6][7]. Feature extraction is performed to obtain color and texture information from the image. In this study, the HSV color space is used as it effectively represents human color perception [8], and the Local Binary Pattern (LBP) method is used for texture due to its advantages in accuracy and computational efficiency [9][10]. The K-Nearest Neighbor (KNN) algorithm is used as a classification method due to its simplicity and effectiveness in grouping data based on proximity [11]. Model performance evaluation is conducted using a confusion matrix to calculate accuracy, precision, and recall values as classification performance indicators [12][13].

This research aims to develop a classification system for Mount Merapi images by utilizing a combination of HSV color feature extraction, Local Binary Pattern (LBP) texture feature extraction, and the K-Nearest Neighbor (KNN) algorithm. The HSV method is chosen because of its ability to separate color components into hue, saturation, and value, making it more efficient than other color models in image processing. Meanwhile, the LBP method has a relatively high accuracy in texture feature extraction, as demonstrated in various previous studies [11]. The KNN algorithm is chosen for its simplicity and effectiveness on large datasets [14], although its results are highly dependent on the selection of the nearest neighbor parameter (k) [15].

Several previous studies support the effectiveness of the methods used. A study by Lamasigi (2020) used a combination of LBP and KNN for medicinal plant leaf recognition, achieving an accuracy of 96%. Another study by Liantoni and Nugroho (2015) compared KNN with Naïve Bayes for herbal leaf classification, where KNN achieved an accuracy of 70.83%. These results indicate that the combination of these methods can be applied to various image data types to achieve optimal results.

In this research, the data used consists of 840 images of Mount Merapi from the Tunggularum Observation Post, taken from October 1 to 10, 2023. This data is divided into 672 training data and 168 testing data with a resolution of 1280 x 720 pixels in .jpg format. The classification process involves analyzing color features using the HSV color space and texture features using the LBP method, which are then classified using the KNN algorithm. It is expected that this system will improve the efficiency of managing Mount Merapi image data and contribute to the development of more effective and sustainable volcanic monitoring technology.

RESEARCH METHOD

The research methodology is a systematic procedure aimed at achieving the objectives of a study. Before conducting this research, various stages were outlined to design the research in order to ensure the research goals are achieved.

This study will employ a quantitative research method of development with secondary datasets using images of Mount Merapi for its analysis. The data was collected from the Tunggularum Observation Post from October 1 to 10, 2023. The data will then be processed and classified based on its visibility. The quantitative research method was chosen because the collected data will be analyzed mathematically and scientifically, allowing for well-founded conclusions. The type of research used is implementation research, as it applies, tests, and evaluates theories to solve research problems. Figure 1 illustrates the research stages, which are shown in a flowchart.

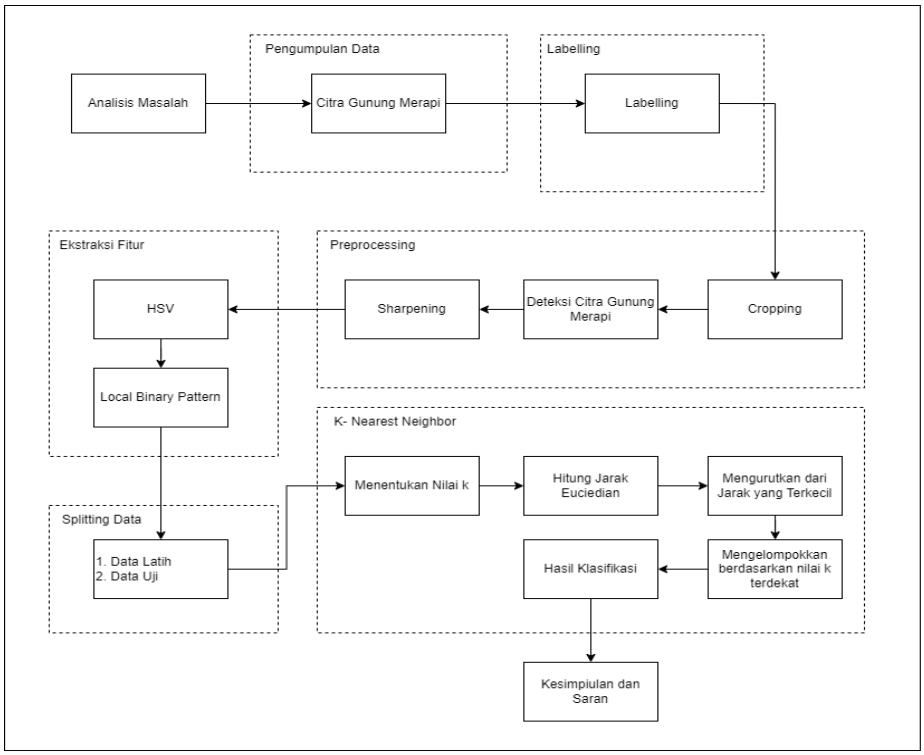


Figure 1. Research Methodology

Data Collection

Data collection was carried out to obtain data used for the research, where the data gathered will be processed and analyzed to extract useful information for solving the research problem. The data used in this study is secondary data, sourced from BPPTKG. The data was recorded from October 1, 2023, to October 10, 2023, with a total of 14,934 data points. However, only 5,187 data points were validated, due to some data being unsuitable for use and limitations of the equipment used in this study. The data used in this research consists of 840 images, with 140 images per class. Below in Figure 2 is an example of the data that will be used for the research.

		
Partial Night	Clear Night	Cloudy Night Tertutup
		
Partial Day	Clear Day	Cloudy Day

Figure 2. Data Example

Preprocessing

The preprocessing stage is the initial step in data processing, aiming to transform raw data into structured, ready-to-be-processed data for subsequent stages. For images, there is a specific preprocessing step called image preprocessing. Image preprocessing is a process of preparing images or visuals before they are processed in the next stage. The goal of image preprocessing is to improve the image quality, remove noise or disturbances, and adjust the image characteristics to better align with the processing objectives. The preprocessing steps used in this study include cropping and image sharpening using unsharp masking.

HSV Feature Extraction

After the image preprocessing process, the next step is feature extraction by obtaining values from the HSV. The HSV extraction method is used to convert colors into HSV. The purpose of converting colors to HSV is to obtain a more complex color mixture with a specific composition. The first step is to convert the R, G, and B values to H, S, and V values. The values will be normalized using the following equation:

$$r = \frac{50}{50+62+60} = 0{,}2907$$
$$g = \frac{62}{50+62+60} = 0{,}3605$$
$$b = \frac{60}{50+62+60} = 0{,}348$$

After obtaining the normalized R, G, and B values, the H, S, and V values can be calculated using the following formula:

Nilai V (*Value*)

$$V = \max([0.2907, 0.3605, 0.3488])$$
$$V = 0.3605$$

Nilai S (*Saturation*)

$$S = ([0.2907, 0.3605, 0.3488])$$
$$\delta = 0.3605 - 0.2907$$
$$\delta = 0.0698$$
$$S = \frac{\delta}{V}$$
$$S = \frac{0.0698}{0.3605}$$
$$S = 0.1936$$

H Value (*Hue*) with B = H, next

$$B = \frac{4+\frac{0{,}2907-0{,}3605}{0.0698}}{6}$$
$$B = \frac{4-\frac{0{,}0698}{0.0698}}{6}$$
$$B = \frac{4-1}{6}$$
$$B = 0{,}5$$

From the calculation of the RGB to HSV conversion, the results are obtained as shown in Table 1 below:

Table 1. HSV Value

Hue	***Saturation***	***Value***
0.5	0.1924	0.3797

LBP Feature Extraction

In this study, the extraction method used is Local Binary Pattern (LBP). Local Binary Pattern is a method with a high level of accuracy [1]. LBP is used to determine the texture size of grayscale images. The steps in the Local Binary Pattern method are to determine the matrix values, then find the central value of the matrix and its neighboring values, followed by the thresholding, encoding, and finally the histogram steps. The matrix values are determined by taking a 3x3 matrix and the central value of the matrix as the central threshold value. After obtaining the 3x3 matrix values, the process continues with the thresholding step as shown in Figure 3.

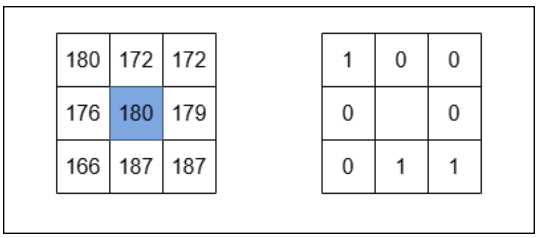


Figure 3. *Thresholding*

In the thresholding step, the values of the neighboring pixels in each pattern will be compared with the central value of the matrix, and then converted into binary values (0 and 1). If the value at the edge is greater than the value at the center point, it will be converted to 0. Conversely, if the edge value is smaller than the central value, it will be converted to binary value 1. This step is used to differentiate the local binary values in each part. The Encoding step can be seen in Figure 4.

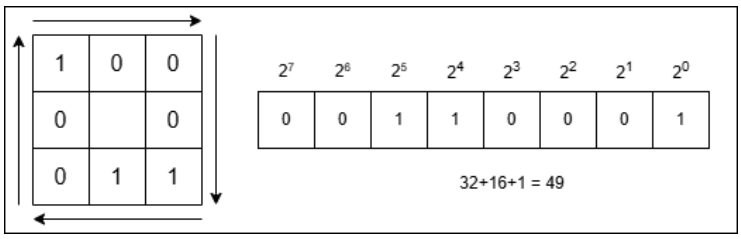


Figure 4. *Encoding*

The equation that can be used for calculations using the LBP method is as follows:

$$LBP_{P,R}(x_c, y_c) = \sum_{P=0}^{P=1} s\,(g_p - g_c)2^p \quad \text{(1)}$$

Explanation:

P = Number of neighboring pixels
R = Distance value / radius
gp = Value of neighboring pixel
gc = Value of pixel x and y
xc,yc = Central coordinates

And the function s(x) can be defined in the following equation (Equation 2):

$$(x) = \begin{cases} 1, x & \geq 0 \\ 0, x & < 0 \end{cases} \quad \text{(2)}$$

The next step is to create a histogram that displays the frequency of occurrence of the LBP values. These values are combined into an identification in the form of a histogram using the following equation:

$$H(k) = \sum_{i=1}^{N} \sum_{i=1}^{M} f\big(LBP_{P,R}(i,j)k\big), k \quad \text{(3)}$$

After obtaining the histogram values, the next step is to normalize them, with the goal of reducing the differences between feature values. The equation for the histogram normalization process is shown below.

$$X_{baru} = \frac{X_{lama}}{sum\ x} \quad \text{(4)}$$

K-Nearest Neighbor

The KNN algorithm is a classification algorithm that works by taking some k nearest neighbors as references to determine the class of a new data point. The classification of the KNN algorithm is based on the similarity or proximity between data points. KNN can also classify based on the number of data points that have the most similarity. The KNN algorithm has been used for statistical estimation and pattern recognition. The distance typically used in KNN is Euclidean Distance, which can be calculated according to two equations where xi represents the test data and yi represents the training data at the i-th index. In general, the Euclidean distance formula can be expressed as follows in equation (5).

$$d(x,y) = \sqrt{\sum_{i=1}^{n} \left(x_{training}^{i} - x_{testing}\right)^2} \quad \text{(5)}$$

Keterangan:

$x_{training}^{i}$: data training ke-i
$x_{testing}$: data testing
i : baris ke-i dari table
n : jumlah data training

The working process of the KNN algorithm is as follows:

1. Determine the value of k (the number of nearest neighbors).
2. Calculate the distance using the Euclidean distance formula for each training data point against the test data to be used.
3. Sort the calculation results based on the smallest values (ascending order).
4. Collect the category Y (KNN classification) based on the value of k.
5. The majority or most frequent KNN data will produce a new class of data, and this data will predict the value of the query instance calculated in the previous step.

FINDINS AND DISCUSSIONS

The testing was conducted using a dataset of 840 images of Mount Merapi with 6 different classes. The testing was performed by building a model using the K-Nearest Neighbor method.

Result Testing

The first step is to label the dataset. The dataset used consists of 840 data points, divided into 672 for training data and 168 for testing data. After labeling, the next step is preprocessing. The preprocessing stage includes several steps, such as image sharpening using unsharp masking. The algorithms used in the preprocessing stage are as follows:

Algorithm: Data Preprocessing

Input : Array img_dir citra gunung

Output : Array img_preprocessing hasil preprocessing citra

Procedure :

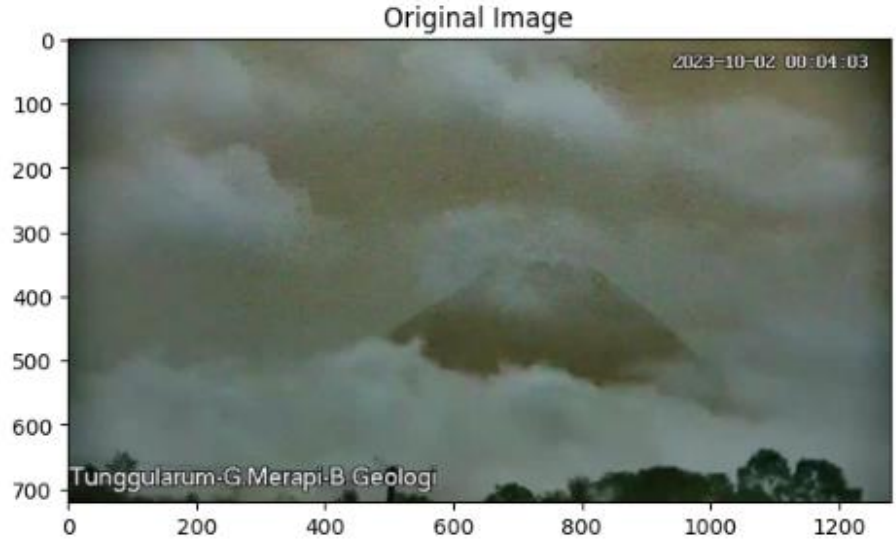
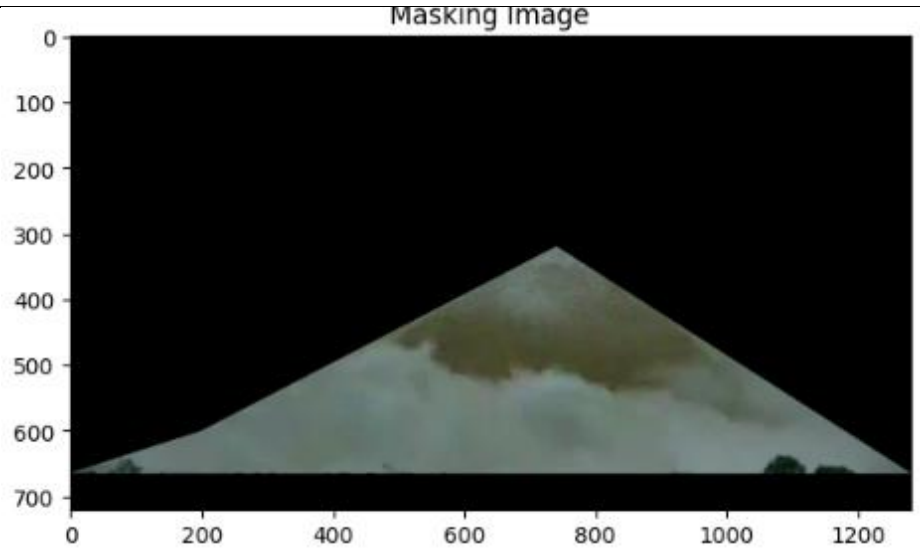
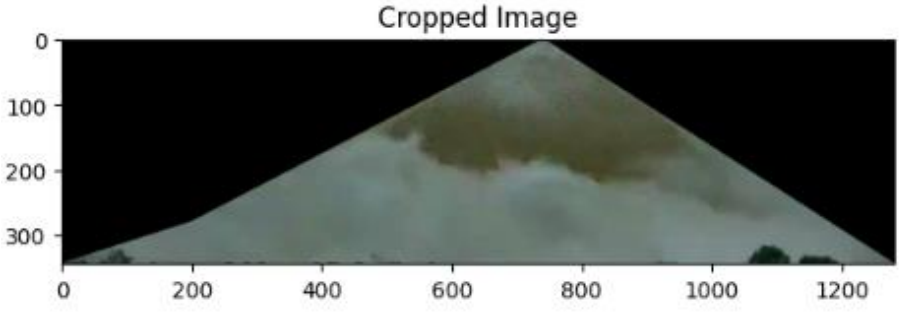
```
img_preprocessing = []
for i in img_dir
    img = read_image(i)
    height, width, channels = get_image_dimensions(img)
    pts = [[200, 600], [740, 320], [1280, 664], [0, 664] ]
    mask = create_black_image(height, width)
    fill_polygon(mask, pts)
    result = bitwise_and(img, mask)
    x, y, w, h = bounding_rectangle(pts)
    cropped_image = crop_image(result, x, y, w, h)
    blur = apply_gaussian_blur(cropped_image, sigma=3)
    sharpened = apply_sharpening(cropped_image, blur)
    append(img_preprocessing, sharpened)
    img = read_image(img_dir_sorted[n])
    height, width, channels = get_image_dimensions(img)
    pts = [ [200, 600], [740, 320], [1280, 664], [0, 664] ]
    mask = create_black_image(height, width)
    fill_polygon(mask, pts)
    result = bitwise_and(img, mask)
    x, y, w, h = bounding_rectangle(pts)
    cropped_image = crop_image(result, x, y, w, h)
    blur = apply_gaussian_blur(cropped_image, sigma=3)
    sharpened = apply_sharpening(cropped_image, blur)
```

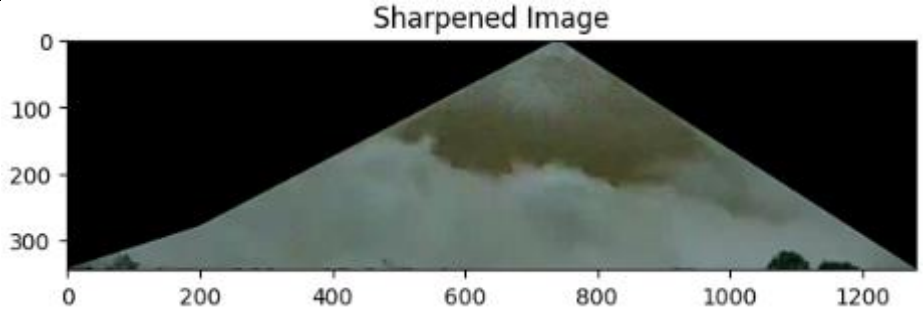
endfor

Algorithm 1. Data preprocessing

From the preprocessing process, an example of the results can be seen in Table 2.

Table 2. Image *Preprocessing*

Process	**Result**
Original Pictire	
Masking	
Cropping	

Process	Result
Sharpening	

After the preprocessing stage is completed, the next step is feature extraction. The feature extraction used includes color feature extraction using the HSV method and texture feature extraction using LBP. The algorithm used for the HSV color feature extraction is as follows:

Algorithm: Fiture Extraction HSV

Input : Array img_preprocessing hasil preprocessing
Output : Array hsv_list hasil ekstraksi fitur HSV
Procedure :

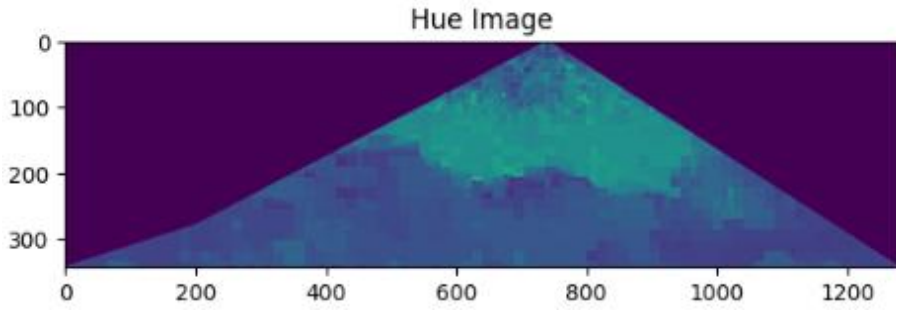
```
hsv_list = []
for i in preprocessing
    hsv_img = cv.cvtColor(i, cv.COLOR_RGB2HSV)h =
    hsv_img[:, :, 0]
    s = hsv_img[:, :, 1]
    v = hsv_img[:, :, 2]
    h_mean = np.mean(h)
    s_mean = np.mean(s)
    v_mean = np.mean(v)
    hsv_vektor = np.concatenate((h_mean.flatten(),s_mean.flatten(),
    v_mean.flatten())) hsv_list.append(hsv_vektor)
```

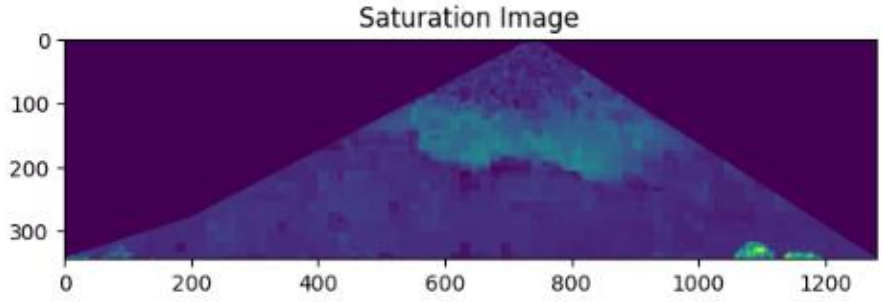
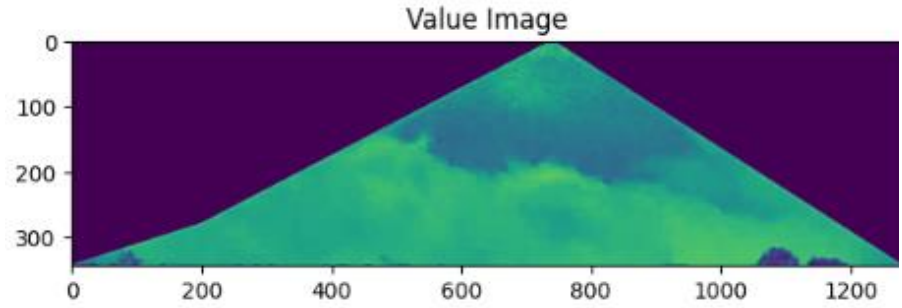
endfor

Algorithm 2. Fiture Extraction HSV

From the preprocessing process, an example of the results can be seen in Table 3, which shows the processed outcomes of color feature extraction, namely Hue, Saturation, and Value.

Table 3. Citra HSV

Result Extraction	Figure
Hue	

Result Extraction	**Figure**
Saturation	
Value	

After the HSV color feature extraction stage, the next step is to perform texture feature extraction using Local Binary Pattern with the following algorithm:

Algorithm : Feature Extraction Local Binary Pattern

```
Input     :  Array img_preprocessing  hasil preprocessing
Output    :  Array img_lbp  hasil ekstraksi fitur LBP
Procedure :  img_lbp2 = []
             for i in tqdm(img_dir_sorted, desc="load"):
              img = cv.imread(i)
              left, top, right, bottom = 0, 300, 1280, 600
              cropped_image = img[top:bottom, left:right]
              blur = cv.GaussianBlur(cropped_image, (0, 0), 3)
              sharpened = cv.addWeighted(cropped_image, 1.5, blur, -
              0.5, 0)
              gray = cv.cvtColor(sharpened, cv.COLOR_RGB2GRAY)
              array = np.array(gray)
              lbp = local_binary_pattern(array, 8, 1, method = 'uniform')
              hist, bins = np.histogram(lbp.ravel(), bins=np.arange(0, 11), range=(0, 10))
             hist = hist.astype("float")
              hist /= (hist.sum() + 1e-7)
              img_lbp2.append(hist)
```

Algorithm 3. Feature Extraction *Local Binary Pattern*

After the thresholding and encoding stages, LBP value normalization is performed to minimize the differences in feature values. Table 4 shows the results of the normalization process.

Table 4. LBP Normalization

	Fitur *Local Binary Pattern*										Kelas
	X1	**X2**	**X3**	**X4**	**X5**	**X6**	**X7**	**X8**	**X9**	**X10**	
1	0.014	0.041	0.019	0.080	0.065	0.231	0.063	0.052	0.362	0.074	Partial Night
2	0.014	0.041	0.021	0.087	0.069	0.237	0.068	0.054	0.334	0.074	Clear Night
3	0.015	0.036	0.019	0.060	0.044	0.185	0.046	0.045	0.485	0.064	Cloudy Night
4	0.008	0.028	0.009	0.064	0.048	0.254	0,049	0.037	0.430	0.073	Partial Day
5	0.008	0.026	0.06	0.054	0.033	0.233	0.042	0.037	0.450	0.065	Clear Day
6	0.010	0.032	0.009	0.069	0.044	0.270	0.056	0.043	0.378	0.090	Cloudy Day

After obtaining the feature values from the feature extraction process using HSV and LBP, the next step is to perform classification using the K-Nearest Neighbor method. The algorithm used in the KNN process is as follows:

Algorithm: *K-Nearest Neighbour*

Input : Array X_train data latih
Array y_train label data latihArray X_test
data uji
Array y_test label data latih
Array xopt nilai Gbest
Output : String prediction hasil klasifikasi
Procedure : X_train, X_test, y_train, y_test = train_test_split(X-data, labeling_total, test_size=0.2,random_state=1)
hasil_model = KNeighborsClassifier(n_neighbors=1)
hasil_model.fit(X_train, y_train)
prediction = model.predict(X_test)
acc = accuracy_score(y_test, prediction)*100

Algorithm 4. *K-Nearest Neighbors*

The first step in performing classification using the K-Nearest Neighbor method is to determine the value of k (the number of nearest neighbors). After determining the value of k, the next step is to calculate the Euclidean distance of the 168 test data points against the 672 training data points. After calculating the Euclidean distance, these values are sorted based on the closest or smallest distance. Then, several values are selected according to the determined k value. Table 5 shows the confusion matrix from the classification results using the K-Nearest Neighbor (KNN) method with k values of 1, 3, 5, 7, and 9. The rows represent the actual classes, while the columns show the predicted results from the model. Based on this table, one can observe how the classification performance changes with different k values for each time category and visual condition.

Table 5 Classification Result

Aktual	Prediksi																										
	Malam sebagian					Malam terlihat					Malam tertutup					Siang sebagian					Siang terlihat					Siang tertutup				
k	1	3	5	7	9	1	3	5	7	9	1	3	5	7	9	1	3	5	7	9	1	3	5	7	9	1	3	5	7	9
Malam sebagian	28	25	24	24	21	2	0	0	0	1	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Malam terlihat	0	1	2	2	5	24	30	28	28	24	1	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
Malam tertutup	5	3	0	2	2	1	0	0	0	0	15	24	25	25	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Siang sebagian	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	29	28	26	26	1	1	2	3	3	0	1	1	2	2
Siang terlihat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	32	26	27	27	27	0	0	0	0	0
Siang tertutup	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	2	1	2	1	0	0	0	0	25	26	25	25	24

The error graph on the training data shows that as the value of k decreases, the error tends to be lower, but it increases as the value of k increases. This indicates the importance of selecting an optimal k value to maintain a balance between accuracy and the model's generalization ability. The visualization of the relationship between k values and error can be seen in Figure 5.

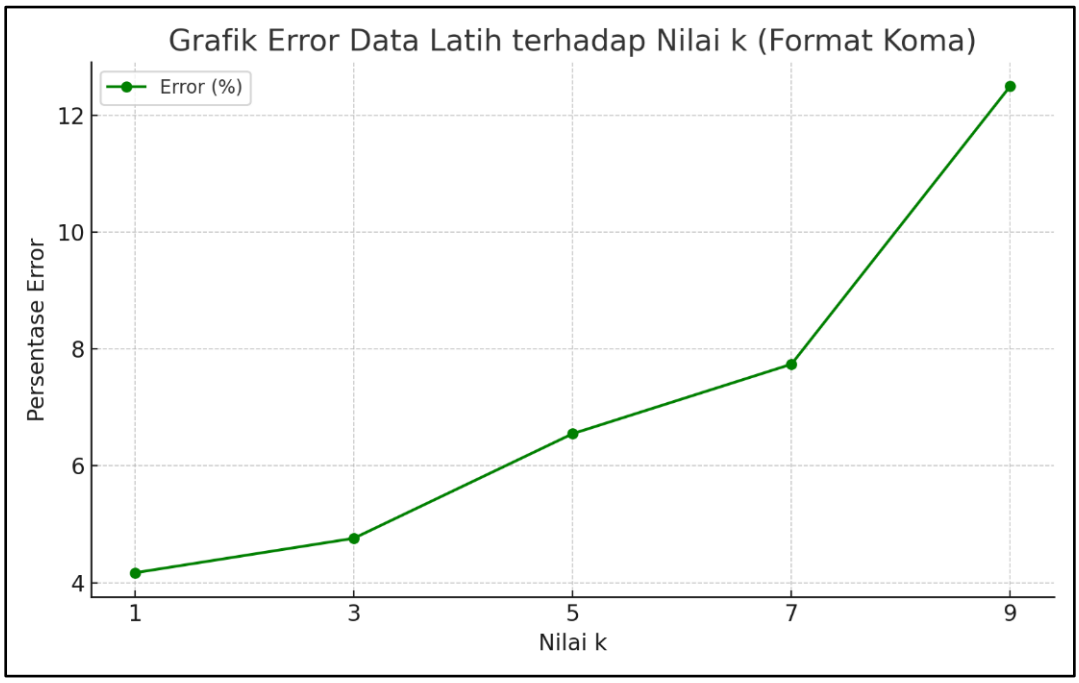


Figure 5. Training Data Error Graph

Discussion

This study is aimed at applying the K-Nearest Neighbor method using HSV color feature extraction and LBP texture feature extraction. The first step in this research is to analyze the issues that arise in this study. Image implementation is used to detect images of Mount Merapi by employing HSV color feature extraction and LBP texture feature extraction. The data used are primary data consisting of 840 images of Mount Merapi obtained from the Tunggularum Monitoring Post between October 1-10, 2023. Afterward, the Mount Merapi images will be classified based on the appearance of the images. The next step is to determine the appearance classes of Mount Merapi, which are divided into 6 classes: Clear Day, Cloudy Day, Partial Day, Clear Night, Cloudy

Night, and Partial Night. The labeling process is completed; the next step is image preprocessing. The preprocessing stage is divided into three steps: cropping, masking, and image sharpening using unsharp masking.

Feature extraction is the next step after preprocessing. The feature extraction used includes color feature extraction using HSV and texture feature extraction using LBP. The process of feature extraction involves obtaining RGB values from each image that has been preprocessed. Once the RGB values are obtained, they are transformed into HSV. After that, the average of the HSV values is taken and used as the color feature values. The next step, after preprocessing, is feature extraction. The texture feature extraction method used is Local Binary Pattern (LBP). The result is a set of 10 feature values.

After obtaining the feature values from the feature extraction process using the Local Binary Pattern method, the next step is to combine both values into a single dataframe that will be used for the classification process.

Classification using K-Nearest Neighbor will be performed once all the feature values are obtained. Before the classification stage, the feature extraction data and labeling will be split into two sets: training data (672 images) and testing data (168 images). A classification model is then created using K-Nearest Neighbor with the training data. The values of k used are 1, 3, 5, 7, and 9. The next step is to calculate the neighborhood distance based on the feature extraction values. After the classification model is built, testing is done using a multi-class confusion matrix. The best accuracy obtained is 95% using k = 1, while the worst accuracy is achieved with k = 9 at 87%. From these testing results, it can be concluded that the classification of Mount Merapi images using K-Nearest Neighbor, along with HSV color feature extraction and LBP texture feature extraction, can be successfully executed, yielding the best accuracy of 95% with k = 1.

CONCLUSIONS & FURTHER RESEARCH

Based on the results of the research on the classification of Mount Merapi images using HSV color feature extraction and LBP texture feature extraction implemented on the K-Nearest Neighbor algorithm, the following conclusions can be made:

1. The results of HSV color feature extraction and Local Binary Pattern texture feature extraction can be used as input values for the classification process of Mount Merapi images using the K-Nearest Neighbor method.
2. The implementation of the K-Nearest Neighbor method for classifying Mount Merapi images based on the reference values from HSV color feature extraction and Local Binary Pattern texture feature extraction works well. Testing results using a multi-class confusion matrix yielded the best accuracy rate of 95.83% with k=1, while the lowest accuracy was achieved with k=9, resulting in an accuracy of 87.5%.

Suggestions for future research, based on the limitations of this study, include adding real-time features for processing Mount Merapi images and adding a feature to select multiple data points simultaneously.

REFERENCES

Afif, R. (2020). *Pemodelan pemilahan citra Gunung Merapi secara otomatis pada BPPTKG Yogyakarta*. *AUTOMATA.*

Hermawati, F. A. (2013). *Data mining.* Yogyakarta: Penerbit Andi.

Hendro, E. P. (2018). Religiusitas Gunung Merapi. *Endogami: Jurnal Ilmiah Kajian Antropologi, 2*(1).

Lamasigi, Z. Y., Hasan, M., & Lasena, Y. (2020). Local binary pattern untuk pengenalan jenis daun tanaman obat menggunakan K-nearest neighbor. *ILKOM Jurnal Ilmiah, 12*(3), 208–218. https://doi.org/10.33096/ilkom.v12i3.667.208-218

Liantoni, F., & Nugroho, H. (2015). *Klasifikasi daun herbal menggunakan metode naïve Bayes classifier dan K-nearest neighbor.*

Lutfi, M. (2019). Implementasi metode K-nearest neighbor dan bagging untuk klasifikasi mutu produksi jagung. *Agromix, 10*(2), 130–137. https://doi.org/10.35891/agx.v10i2.1636

Maniswari, S. D., Rusdinar, A., & Purnama, B. (2015). Smart traffic light menggunakan image processing dan metode fuzzy logic. *e-Proceeding of Engineering, 2*(2), 2166–2170.

Mutrofin, S., Izzah, A., Kurniawardhani, A., & Masrur, M. (2014). Optimasi teknik klasifikasi modified K-nearest neighbor menggunakan algoritma genetika. *Jurnal GAMMA,* 130–134. https://doi.org/10.13140/RG.2.2.22330.08648

Nimah, F. S., Sutojo, T., & Setiadi, D. R. I. M. (2017). Identifikasi tumbuhan obat herbal berdasarkan citra daun menggunakan algoritma gray level co-occurrence matrix dan local binary pattern. *Jurnal Ilmu Komputer dan Informasi, 10*(2), 67–72.

Panggabean, A. K., Syahfaridzah, A., & Ardiningih, N. A. (2020). Mendeteksi objek berdasarkan warna dengan segmentasi warna HSV menggunakan aplikasi MATLAB. *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi, 4*(2), 94–97. https://doi.org/10.46880/jmika.Vol4No2.pp94-97

Priandini, H., Yundari, & Helmi. (2019). Monitoring deformasi Gunung Merapi menggunakan citra Sentinel-1A dengan teknologi differential interferometry SAR. *Jurnal Rekayasa Hijau, 8*(2), 92–100. https://doi.org/10.28932/jrh.v8i2.3741

Retnoningrum, D., Widodo, A. W., & Rahman, M. A. (2019). Ekstraksi ciri pada telapak tangan dengan metode local binary pattern (LBP). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 3*(3), 2611–2618.

Soeb, A. (2019). Perbaikan tingkat kekaburan gambar akibat pembesaran pada hasil screenshot dengan metode unsharp mask. *Jurnal Media Informatika Budidarma, 3*(2). https://doi.org/10.30865/mib.v3i2.1096

Utiarahman, S. A., & Pratama, A. M. M. (2024). Analisis perbandingan KNN, SVM, decision tree dan regresi logistik untuk klasifikasi obesitas multi kelas. *KLIK: Kajian Ilmiah Informatika dan Komputer, 4*(6), 3137–3146. https://doi.org/10.30865/klik.v4i6.1871

Wardhani, I. P., Putri, A. M., Irfan, & Widayati, S. (2021). Algoritma identifikasi ciri citra pegunungan dengan metode copping. *Jurnal Ilmiah Komputasi, 20*(2), 283–289. https://doi.org/10.32409/jikstik.20.2.2763