Acceleration, Quantum, Information Technology and Algorithm Journal (AQILA)
Vol. 3, No. 1, June 2026, pp 18-28
ISSN: 3062-8555, DOI: 10.62123/aqila.v3i1.174

Application In Distinguishing Artificial Intelligence-Makened Images And Original Images With Visual Feature Extraction Using Ensemble Learning Algorithm

Moh Hafiz Naufal[1], Al-Khowarizmi[1*]

[1]Department of InformationTechnology, Universitas Muhammadiyah Sumatera Utara, Indonesia

*Corresponding Email: alkhowarizmi@umsu.ac.id

DOI : 10.6213/aqila.v3i1.174

Received : May 29, 2026
Revised : June 15, 2026
Accepted : June 16, 2026

Keywords:
Artificial Intelligence
Visual Feature Extraction
Ensemble Learning

ABSTRACT

The development of generative Artificial Intelligence (AI) technology enables computer systems to produce highly realistic images that closely resemble real photographs. This condition creates challenges in distinguishing AI-generated images from real images visually. This research aims to develop an image classification system capable of distinguishing AI-generated images and real images using visual feature extraction and ensemble learning algorithms.The research method consists of several stages including image preprocessing by resizing images to 256 × 256 pixels, visual feature extraction including RGB color histogram, grayscale intensity distribution, texture features using Gray Level Co-occurrence Matrix (GLCM), and edge features using the Canny Edge Detection method. The extracted features are then used as input for several classification algorithms such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Random Forest. Furthermore, model combination is performed using an ensemble learning method with a hard voting technique.The experimental results show that the Random Forest model achieved an accuracy of 65.71%, while the ensemble learning method achieved an accuracy of 65.00% with an F1-score of 0.6918. The developed system is also implemented as a web-based application using the Streamlit framework, allowing users to upload images and obtain prediction results directly. The results indicate that the combination of visual feature extraction and ensemble learning can be used as an approach to help identify AI-generated images and real images.

1. INTRODUCTION

Artificial Intelligence (AI) technology has experienced rapid progress, particularly in the field of digital image processing. Generative AI models such as the Generative Adversarial Network (GAN) [1] and other generative AI models are capable of producing images with highly realistic visual quality that closely resemble the original camera image. This capability is widely utilized in various fields, such as the creative industry, graphic design, advertising, and digital media [2].

However, with the advancement of AI technology[3], AI-generated images are becoming increasingly difficult to visually distinguish from the original. This situation has the potential to lead to misunderstandings in the public, especially for lay users such as parents who tend to trust digital visual content without verification. Most currently available detection systems use complex, black-box deep learning-based approaches, making the model's decision-making process difficult to understand. Therefore, a scientific approach is needed that not only produces classification but also can more transparently explain the visual characteristics that differentiate AI-generated images from the original.

As these problems develop, various studies have been conducted to distinguish AI-generated images from the original using image processing and machine learning techniques [4][5]. One commonly used approach is visual extraction, the process of extracting important characteristics from images such as texture, color, edge patterns, and pixel distribution. These visual features have been shown to represent the characteristic differences between AI-generated images and real images, although they are visually difficult to distinguish by the human eye [6].

However, using a single classification algorithm often fails to provide optimal results, especially when faced with increasingly complex data variations. Therefore, methods are needed that can improve classification accuracy and stability [7][8]. The ensemble learning algorithm is an approach that combines multiple learning models to produce more accurate decisions than a single model [9]. Based on these issues, this study proposes the implementation of a system to distinguish between AI-generated images and real images by extracting visual features using the Ensemble Learning algorithm [10]. This research is expected to

facilitate the automatic identification of image authenticity and serve as a solution to minimize the spread of image-based hoaxes and improve digital literacy in the community.

2. RESEARCH METHODOLOGY

This research uses a quantitative approach with computational experimental methods to develop a classification system capable of distinguishing between AI-generated images and real images. The research framework falls under the category of supervised learning, utilizing labeled datasets. The system development process follows a systematic process, including data collection, preprocessing, feature extraction, model training, performance evaluation, and validation. The data used consists of digital images obtained from open sources and categorized into two main classes: AI-generated images and real images. Using programming tools such as Python and supporting libraries like OpenCV, this research aims to produce a prototype detection system that can classify images with high accuracy.

In the research, image data, after undergoing preprocessing and visual feature extraction, is then used to train a model using an ensemble learning approach. The evaluation process involves testing the model on separate test data, with results measured using evaluation metrics such as accuracy, precision, recall, and F1-score. Validation of the classification results is conducted to ensure the model can reliably distinguish between AI-generated and real images. Classification error analysis is also conducted to improve the method and results in future research. Thus, this research not only produces theory, but also system implementation that can be evaluated objectively and contributes to the field of digital image processing and machine learning [11][12].

3. RESULTS AND DISCUSSION

The dataset used in this study consists of digital images divided into two categories: AI-generated images and real images. The total dataset used was 560 images, with a balanced distribution across each class. The dataset details are as follows: AI Images: 280 images and Real Images: 280 images. The dataset was then divided into two parts: training data and testing data, with separate divisions made from the beginning of the data collection process. The training data consisted of 210 AI-generated images and 210 real images. The test data consisted of 140 images, consisting of 70 AI images and 70 real images.

The dataset was divided equally between the two classes to avoid bias in the model toward one category. With this balanced data composition, it is hoped that the classification model will perform optimally in distinguishing between AI-generated images and real images.

3.1. AI Image Data Source

The AI image data in this study was obtained through an image generation process using the Grok artificial intelligence platform developed by xAI. The data collection process involved inputting specific prompts to generate a variety of images according to the research needs. A screenshot of the AI image generation process is shown in Figure 1 as evidence of the data collection process.

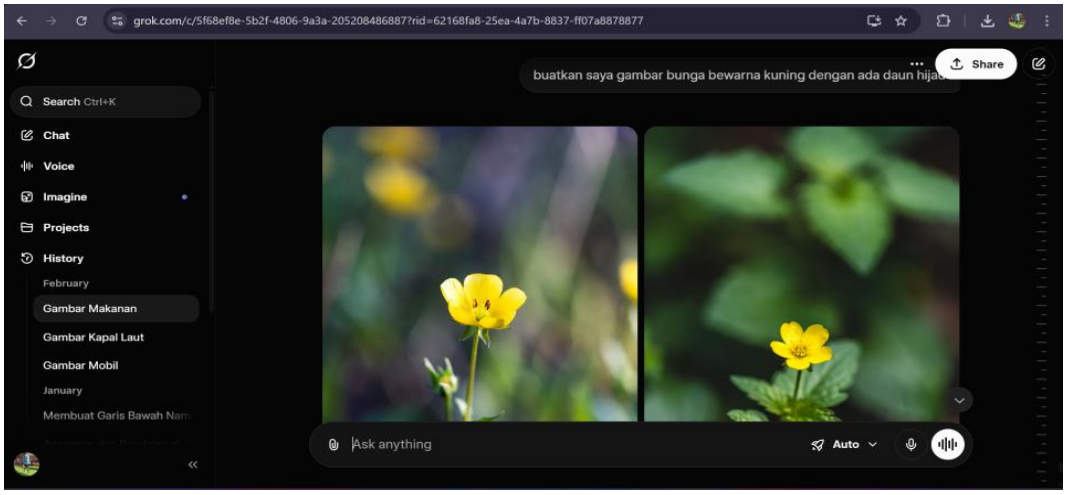

Figure 1. Example of AI image data retrieval on the Grok platform

3.2. Original Image Data Source

The original image data (real images) used in this study were obtained from the license-free photo hosting platform Unsplashed. This platform provides high-quality photographic images uploaded by professional photographers and the global community. Data collection was carried out by selecting images with natural visual characteristics, such as realistic lighting, texture variations, and general photographic composition. This selection was made selectively to ensure that the images used were truly real photographs and not generated by artificial intelligence. A screenshot of the image collection process from Unsplash is shown in Figure 2 as documentation of the data source.

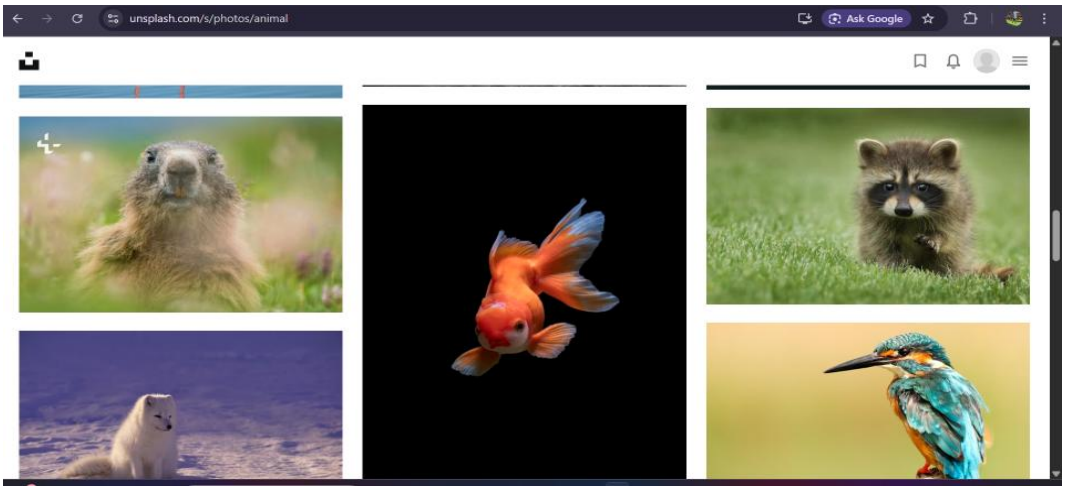

Figure 2. Examples of original image data retrieval

3.3. Data Preprocessing Results

The preprocessing stage is carried out to ensure that all images have uniform characteristics before the feature extraction process [13]. The datasets used in this study vary in size and resolution, requiring an adjustment process to ensure consistent processing by the classification system. Preprocessing is a crucial stage in digital image processing because it can improve the quality of data representation and help the model learn visual patterns more stably [14]. One common step at this stage is standardizing the image size so that each image has the same dimensions, allowing for more consistent processing by the machine learning algorithm.

In this study, all images in the dataset were converted to a size of 256 x 256 pixels using the OpenCV library [15] in the Python programming language. The resizing process aims to standardize the image dimensions so that differences in resolution between images do not affect the feature extraction and classification processes. By resizing, all images in the dataset have a uniform size, allowing for more effective processing during the visual feature extraction stage. This process helps the system analyze image characteristics more consistently, regardless of image size variations [16].

Figure 3. Results of standardizing image size

3.4. Data Preprocessing Results Visual Feature Extraction Results

At this stage, visual features are extracted from preprocessed images. Feature extraction aims to transform digital images into numerical representations that can be processed by classification algorithms. Visual features play a crucial role in image recognition systems because they represent the visual characteristics of an object in the form of numerical data that can be analyzed by machine learning models [17]. In this study, the visual features used include color, texture, edge, and pixel intensity distribution. The feature extraction process was performed using the Python programming language with the help of the OpenCV and Scikit-image libraries. In Figure 4, standardized to 256×256 pixels, was then processed to generate a feature vector representing the image's visual characteristics [18].

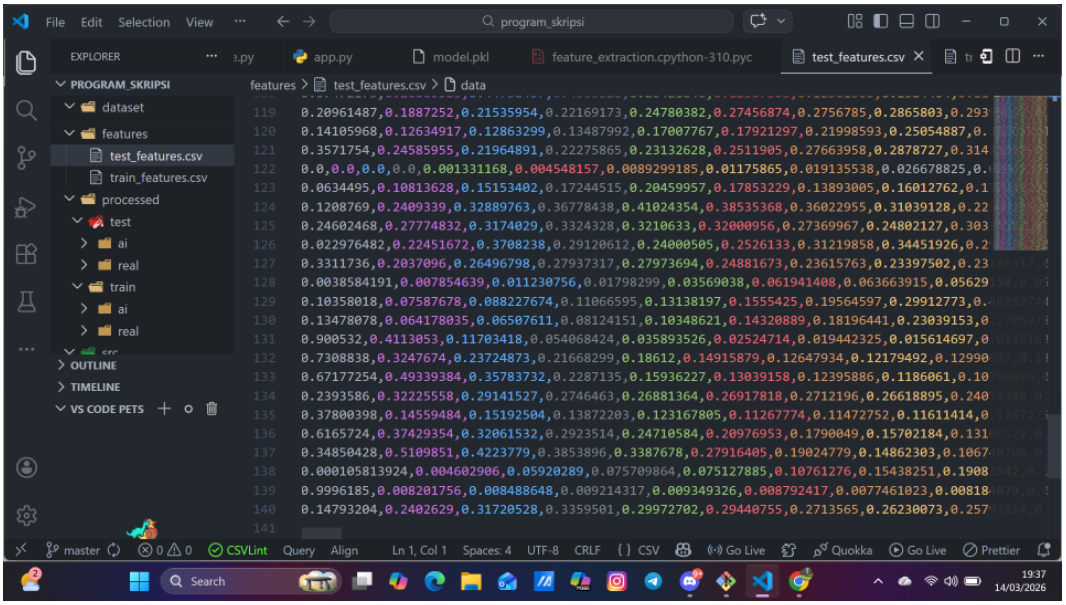

Figure 4. Visual Feature Extraction Results

3.5. Colour Feature Extraction

Colour features are obtained using a color histogram in the RGB color space. A color histogram is used to describe the distribution of red, green, and blue color intensities in an image. This approach is widely used in image analysis because it effectively represents color characteristics in the form of statistical distributions [19].

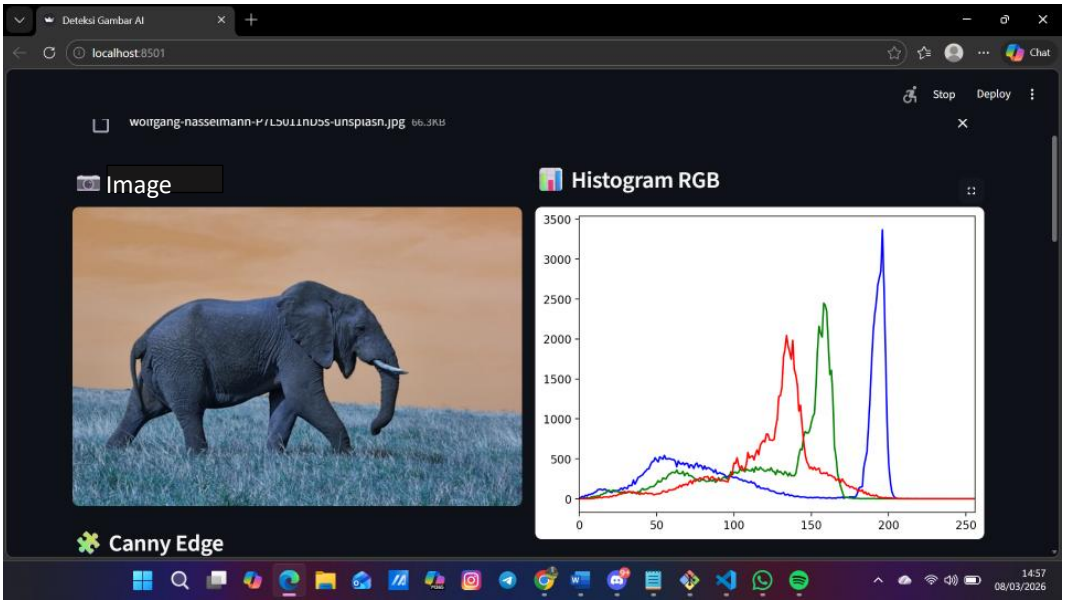

Figure 5. Histogram of RGB color distribution

In this process Figure 5, each color channel is analyzed to calculate the pixel intensity distribution. The resulting histogram values are then normalized to produce a more stable feature representation that can be used as input in the classification process.

3.6. Texture Features

The texture features in this study were obtained using the Gray Level Co-occurrence Matrix (GLCM) method. This method is used to analyze the spatial relationships between pixels in grayscale images to describe the texture patterns contained in the image. In Figure 6, from the GLCM matrix several statistical parameters are calculated as texture features, namely contrast, correlation, energy, and homogeneity. These parameters are used to represent the complexity of the texture and the intensity relationships between pixels in the image [20]. These texture feature values are then used as part of the feature vector that will be processed in the classification stage.

Figure 6. Results of texture feature extraction using GLCM

3.7. Edge Features

Edge features are obtained using the Canny Edge Detection method. This method is used to detect significant changes in pixel intensity to identify object boundaries or contours in an image. Edge information can help the system recognize object structures in an image [21].

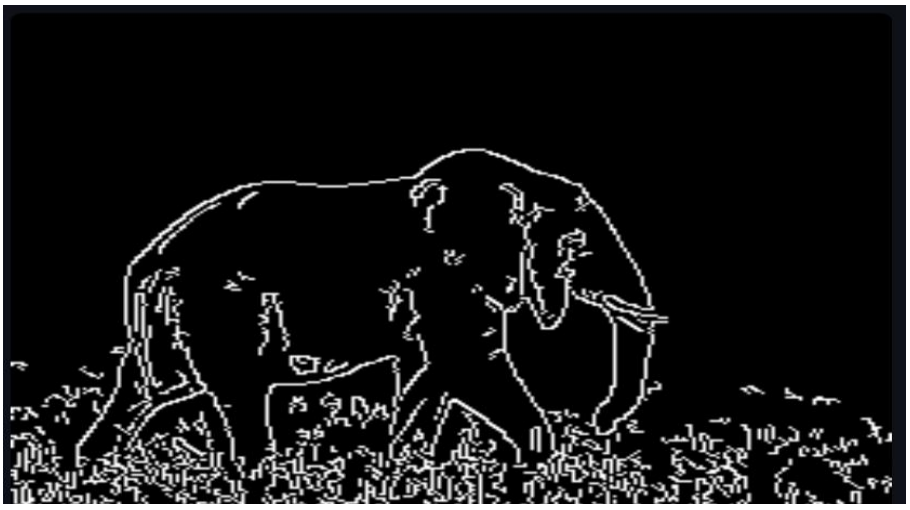

Figure 7. Edge detection results using the Canny method

In this study figure 7, in addition to generating edge images, the ratio of the number of edge pixels to the total pixels in the image, known as the edge ratio, was also calculated. This value is used as a feature to describe the complexity of the object structure in the image.

3.8. Edge Features

The distribution of pixel intensities is obtained using a grayscale histogram. This histogram shows the distribution of pixel intensity values in an image after it has been converted to grayscale. This intensity distribution can describe the light and dark characteristics of an image, which is one indicator of the difference between an AI-generated image and a real image. The

grayscale histogram is then normalized so that it can be used as part of the feature representation in the classification process in figure 8.

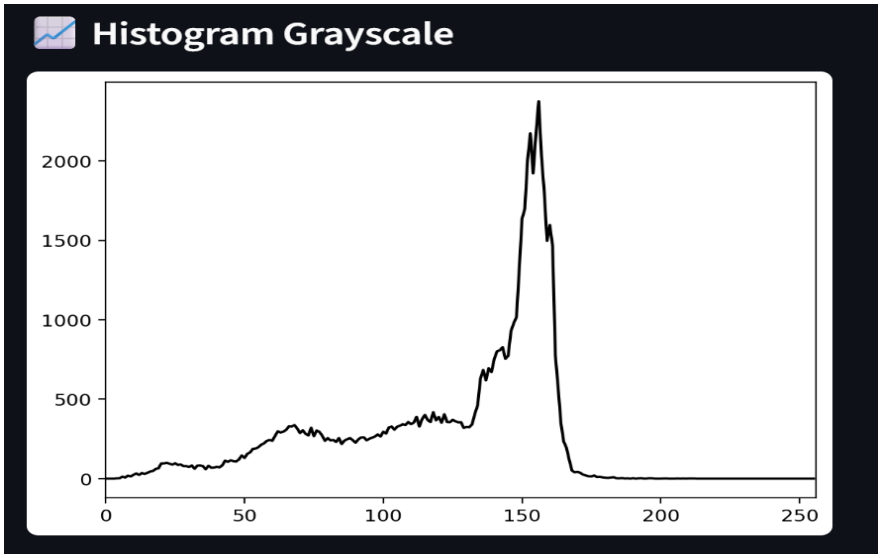

Figure 8. Grayscale Intensity Distribution Histogram Results

3.9. Model Training Results

After the visual feature extraction process is complete, the next step is training a classification model using the features obtained from the images. This study used several machine learning algorithms, namely Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Random Forest. Furthermore, several models were combined using Ensemble Learning with Hard Voting to improve classification performance. Model training was performed using a feature dataset that had been separated into training and testing data. The training data was used to train the model to learn patterns from visual image features, while the testing data was used to test the model's ability to classify previously unseen data. The use of training and testing data aimed to measure the model's generalization ability in image classification [22].

3.9.1 Support Vector Machine (SVM) Model Training

A Support Vector Machine is a classification algorithm that works by finding the optimal hyperplane that maximally separates two data classes in a feature space. This method is widely used in image processing and pattern recognition due to its ability to handle high-dimensional data well [23][24]. The training results of the Support Vector Machine (SVM) model in this study are shown in Figure 4.8. These results show the model's evaluation values, including accuracy, precision, recall, and F1-score, obtained from the testing process using the test data.

```
=== SVM ===
Accuracy : 0.5786
Precision: 0.5470
Recall   : 0.9143
F1-Score : 0.6845
```

Figure 9. SVM Evaluation Results

Based on the evaluation results in Figure 8, the Support Vector Machine model produced an accuracy value of 0.5786, a precision of 0.5470, a recall of 0.9143, and an F1-score of 0.6845 in Figure 9. The relatively high recall value indicates that the model is able to recognize most of the images in the target class, but the relatively lower precision value indicates that there are still some classification errors in the prediction process. These results indicate that the SVM model has a fairly good ability to detect images.

3.9.2 K-Nearest Neighbor (KNN) Model Training

The K-Nearest Neighbor algorithm is a distance-based classification method that determines the class of data based on the number of nearest neighbors in a feature space. This method is often used in pattern recognition due to its simple yet effective concept in classifying data based on feature similarity [25].
In this study, the KNN model was trained using visual features extracted from images. The training process was carried out using training data, while model evaluation was conducted using testing data to determine the model's performance in classifying AI images and real images.

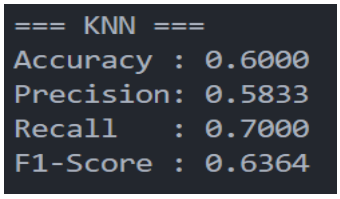

Figure 10. KNN Model Evaluation Results

Based on the evaluation results in Figure 4.9, the K-Nearest Neighbor model produced an accuracy value of 0.6000, a precision of 0.5833, a recall of 0.7000, and an F1-score of 0.6364 in Figure 10. A fairly good recall value indicates that the model is able to recognize most images in the target class, while the precision value indicates the model's level of accuracy in making predictions. These results indicate that the KNN model has a fairly stable performance in classifying images compared to other models.

3.9.3 Random Forest Training

Random Forest is an ensemble decision tree-based classification algorithm that works by randomly constructing multiple decision trees. Each decision tree performs the classification process independently, and then the predictions from all trees are combined to produce a final decision. This method is known to be good at handling data with a large number of feature dimensions and can reduce the risk of overfitting during model training [26]. In this study, the Random Forest model was trained using visual features obtained from the image feature extraction process. The model was then tested using test data to determine its classification performance in distinguishing between AI-generated and real images.

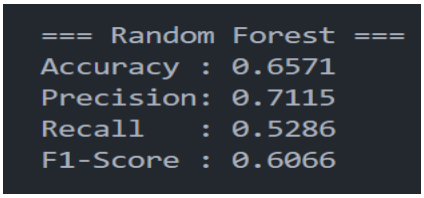

Figure 11. Random Forest evaluation results

Based on the evaluation results in Figure 10, the Random Forest model produced an accuracy value of 0.6571, a precision of 0.7115, a recall of 0.5286, and an F1-score of 0.6066 in Figure 11. The relatively high precision value indicates that the model has a good level of accuracy in predicting image classes. Furthermore, the higher accuracy value compared to other models indicates that Random Forest has better performance in classifying the dataset used in this study.

3.9.4 Ensemble Learning Training (Hard Voting)

Ensemble learning is a machine learning method that combines multiple classification models to produce more stable and accurate predictions. This approach aims to leverage the strengths of each model to improve overall system performance. One frequently used ensemble technique is hard voting, which determines classification results based on the majority vote of several models used [27].

Based on the evaluation results in Figure 11, the ensemble learning model produced an accuracy value of 0.6500, a precision of 0.6180, a recall of 0.7857, and an F1-score of 0.6918 in Figure 12. The relatively high recall value indicates that the ensemble method is able to recognize most images in the target class. In addition, the higher F1-score value compared to several individual models indicates that the ensemble method can provide a better balance between precision and recall in the image classification process.

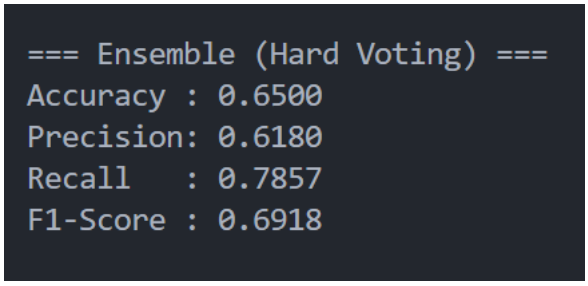

Figure 12. Ensemble learning evaluation results

3.10. Model Evaluation Using Confusion Matrix

After the model training process is complete, the next step is to evaluate the classification model's performance. Model evaluation aims to determine the model's ability to correctly classify images based on the test data used. One evaluation method commonly used in classification systems is the Confusion Matrix. The Confusion Matrix is an evaluation method used to analyze classification model performance by comparing the model's predictions with the actual labels of the test data to determine the number of correct and incorrect predictions [28].

The confusion matrix has four main components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). True Positive indicates the number of data items predicted as positive but actually included in that class. True Negative indicates the number of data items predicted as negative but actually included in that class. False Positives are data items that are actually in the negative class but predicted as positive by the model, while False Negatives are data items that are actually in the positive class but actually included in the negative class [29].

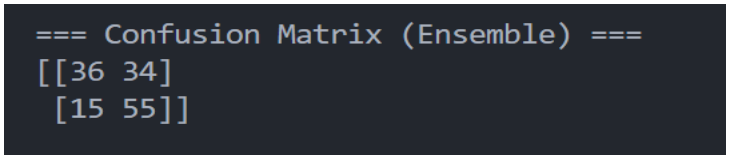

Figure 13. Confusion Matrix Results of Ensemble Learning Model

Based on the confusion matrix in Figure 12, the model successfully classified 36 real images correctly (True Negatives) and 55 AI images correctly (True Positives). However, there were still some misclassifications, with 34 real images predicted as AI (False Positives) and 15 AI images predicted as real (False Negatives) in Figure 13. These results indicate that the model has a fairly good ability to recognize AI-generated images, as evidenced by the relatively high number of True Positives. However, there were still some misclassifications of real images predicted as AI images. This could be due to the similarity in visual characteristics between real images and AI-generated images under certain conditions.

3.11. Model Performance Analysis

Based on the results of the model training and evaluation, an analysis of the performance of each classification algorithm used in this study can be performed. The models used included Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest, and Ensemble Learning with Hard Voting. Test results showed that each model had varying levels of performance in image classification. The Support Vector Machine (SVM) model produced an accuracy of 0.5786, a precision of 0.5470, a recall of 0.9143, and an F1-score of 0.6845. The relatively high recall value indicates that the model was able to recognize most images belonging to the target class, although some misclassification errors resulted in a relatively lower precision value.

The K-Nearest Neighbor (KNN) model produced an accuracy of 0.6000, a precision of 0.5833, a recall of 0.7000, and an F1-score of 0.6364. These results indicate that the KNN model has fairly stable performance in classifying images based on feature proximity between data points. Meanwhile, the Random Forest model performed better than the previous two models, with an accuracy of 0.6571, a precision of 0.7115, a recall of 0.5286, and an F1-score of 0.6066. The relatively high precision value indicates that the model has a good level of accuracy in predicting image classes.

Furthermore, the Ensemble Learning method using the Hard Voting technique, which combines the SVM, KNN, and Random Forest models, produced an accuracy of 0.6500, a precision of 0.6180, a recall of 0.7857, and an F1-score of 0.6918. This higher F1-score than the other models indicates that the ensemble method is able to provide a better balance between precision and recall in the image classification process. Overall, the Random Forest and Ensemble Learning models performed better than the other models in image classification. This shows that the ensemble-based method is able to improve the system's ability to recognize visual feature patterns in images.

Table 1. Comparison of Classification Model Performance

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.5786	0.5470	0.9143	0.6845
KNN	0.6000	0.5833	0.7000	0.6364
Random Forest	0.6571	0.7115	0.5286	0.6066
Ensemble	0.6500	0.6180	0.7857	0.6918

Based on Table 1, the Random Forest model has the highest accuracy value of 0.6571, while the Ensemble Learning method produces the highest F1-score value of 0.6918. This indicates that the ensemble method is able to provide a better performance balance in the image classification process

3.12. System Implementation

In the system implementation phase, the trained classification model is integrated into a web-based application using the Streamlit framework. This system implementation aims to simplify the user detection process for uploaded images, allowing the system to automatically predict whether the image is AI-generated or real. This application was developed using the Python programming language, utilizing several libraries such as OpenCV for image processing, Scikit-learn for classification, and Streamlit as the user interface

3.12.1 System Main Page View

The system's main page is the initial display of the application, which users use to upload images to be analyzed by the system. This page offers an image upload feature that allows users to select images from their device. The Figure 14 shows the initial view of a web-based image detection application developed using Streamlit. On this page, users can upload images for the system to analyze.

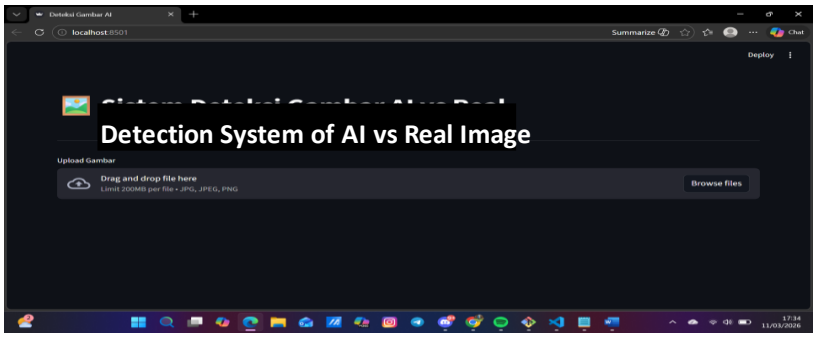

Figure 14. System Main Page View

3.12.2 Image Analysis Process Display

After the image is successfully uploaded, the system will process it by extracting visual features. At this stage, the system displays several image feature visualizations, such as color histograms and edge detection results, as a form of transparency in the image analysis process. The figure 15 shows the image analysis process performed by the system after an image is uploaded by a user. The system displays visualizations of image features such as color histograms and edge detection as part of the feature extraction process.

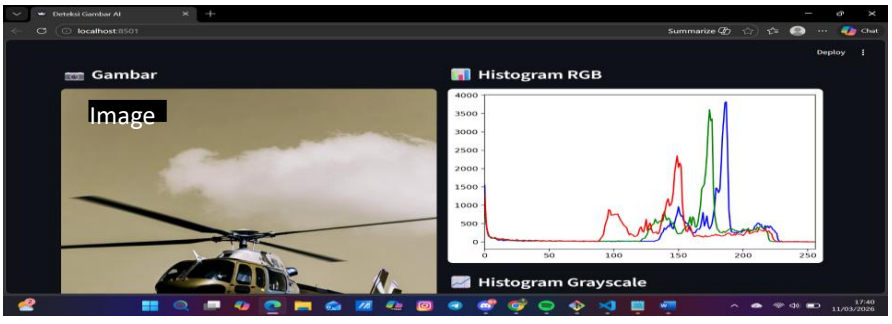

Figure 15. Image Analysis Process Display

3.12.3 System Prediction Results Display

After the image analysis process is complete, the system will display the prediction results in the form of an image category and the model's confidence level in the prediction. This category indicates whether the image belongs to the AI-generated image group or is a genuine image. The system also displays the classification model's confidence level in the prediction. The Figure 16 shows the system's prediction results, indicating whether the image is AI-generated or a real image, as well as the confidence level of the classification model. This confidence value indicates how confident the model is in determining the class of the tested image. The higher the confidence value, the greater the model's confidence in its prediction results. Conversely, a low confidence value indicates that the model is still uncertain about determining the image's category.

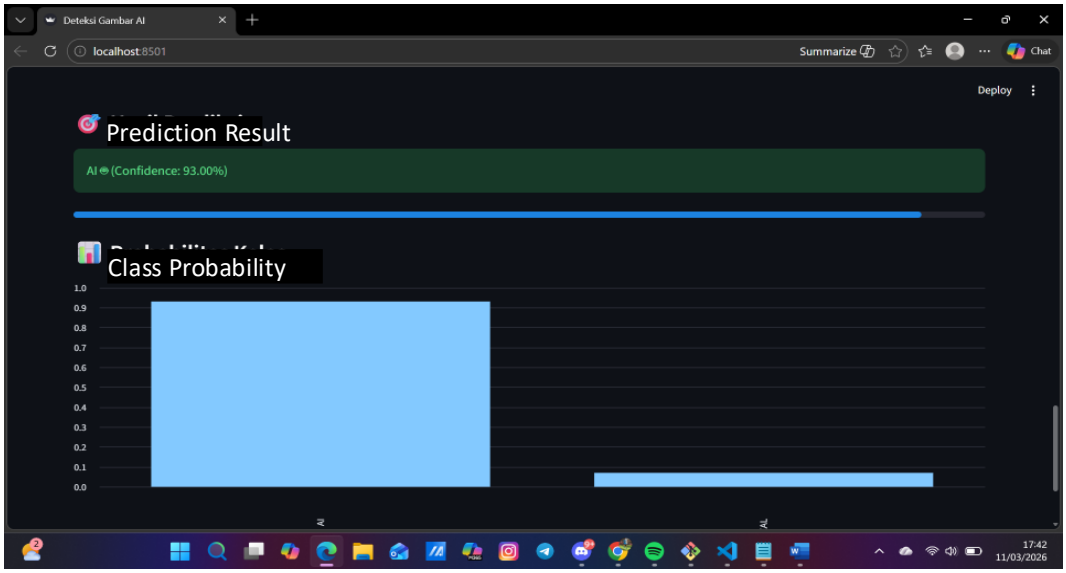

Figure 16. System Prediction Results

4. CONCULUSSION

Based on the research results on a machine learning-based image detection system to distinguish between images generated by artificial intelligence (AI) and real images, the following conclusions can be drawn.

1. This research successfully developed an AI image detection system using a machine learning approach that leverages the visual features of digital images. The research process began with a preprocessing stage, which standardized the image size to 256 x 256 pixels, followed by visual feature extraction, including RGB color histograms, grayscale intensity distributions, texture features using the Gray Level Co-occurrence Matrix (GLCM) method, and edge detection using Canny Edge Detection.
2. The extracted features were then used as input for several classification algorithms, namely Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Random Forest. Furthermore, this research applied ensemble learning with hard voting to combine the predictions from several classification models to improve system performance.
3. Based on the test results using test data, the classification models demonstrated varying performance. The Random Forest model achieved an accuracy of 65.71%, while the ensemble learning (hard voting) method achieved an accuracy of 65.00% with an F1-score of 0.6918. These results indicate that the ensemble method is capable of providing quite good performance in the classification process of AI and real images.
4. Based on the research results, the visual feature extraction and machine learning classification approaches can be used as methods to help identify whether an image is AI-generated or real.

Based on the results of this research, several suggestions can be made for further research:

1. Future research can use a larger and more diverse dataset so that the classification model can better learn visual patterns and achieve a higher level of accuracy.
2. Future research can also add other visual features, such as shape features or image frequency features, to improve the model's ability to distinguish between AI and real images.
3. The developed system can be further developed by adding features such as batch image analysis, allowing the system to process multiple images simultaneously.

REFRENCES

[1] A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," *International Journal of Information Management Data Insights*, vol. 1, no. 1, p. 100004, 2021.

[2] S. A. Alajaji *et al.*, "Generative adversarial networks in digital histopathology: current applications, limitations, ethical considerations, and future directions," *Modern Pathology*, vol. 37, no. 1, p. 100369, 2024.

[3] C. Kurniawan, "Implementasi Artificial Intelligence Dalam Penyelesaian Masalah," *Sinteks: Jurnal Teknik*, vol. 4, no. 1, 2015.

[4] Developers, "Scikit-Learn: Machine Learning in Python." Accessed: Jun. 12, 2026. [Online]. Available: https://scikit-learn.org/stable/

[5] P. Borugadda and H. K. Kalluri, "A comprehensive analysis of artificial intelligence, machine learning, deep learning and computer vision in food science," *Journal of Future Foods*, 2025.

[6] Raisa Shofia and W. Aulia, "Perancangan Perangkat Makan Bersama Bertolak dari Tradisi Bajamba Minangkabau Menggunakan Metode Ekstraksi Unsur Visual," *Keluwih: Jurnal Sains dan Teknologi*, vol. 4, no. 1, pp. 21–33, May 2023, doi: 10.24123/saintek.v4i1.5588.

[7] K. D. T. M. Milanez and M. J. C. Pontes, "Classification of edible vegetable oil using digital image and pattern recognition techniques," *Microchemical Journal*, vol. 113, pp. 10–16, 2014.

[8] F. F. Maulana and N. Rochmawati, "Klasifikasi citra buah menggunakan convolutional neural network," *Journal of Informatics and Computer Science (JINACS)*, vol. 1, no. 02, pp. 104–108, 2019.

[9] A. Alotaibi, "Ensemble Deep Learning Approaches in Health Care: A Review.," *Computers, Materials & Continua*, vol. 82, no. 3, 2025.

[10] U. Indahyanti, N. L. Azizah, and H. Setiawan, "Pendekatan ensemble learning untuk meningkatkan akurasi prediksi kinerja akademik mahasiswa," *Jurnal Sains dan Informatika*, vol. 8, no. 2, 2022.

[11] S. Tyagi *et al.*, "An Enhanced Diabetic Retinopathy Blindness Detection Using Deep Learning and Image Preprocessing Techniques," *Array*, p. 100716, 2026.

[12] A. M. Nababan, U. R. P. Nasution, T. D. Pandiangan, F. Nadi, R. Budiarto, and R. F. Rahmat, "Extreme learning machine approach on heart abnormalities identification in ECG images," *International Journal of Electronics and Telecommunications*, vol. 70, 2024.

[13] S. Hallur and A. Gavade, "Image feature extraction techniques: A comprehensive review," *Franklin Open*, p. 100366, 2025.

[14] R. Bala, N. Duhan, and K. K. Bhatia, "Exploring Image Processing Techniques for Skin Lesion Detection focusing on Segmentation and Classification: A Systematic Review," *Biomed. Signal Process. Control*, vol. 116, p. 109510, 2026.

[15] OpenCV Org, "OpenCV Documentation." Accessed: Jun. 12, 2026. [Online]. Available: https://opencv.org/

[16] Team Pandas, "Pandas documentation." Accessed: Jun. 12, 2026. [Online]. Available: https://pandas.pydata.org/

[17] R. Sharma, R. Gupta, and A. Sinha, "From image processing to artificial intelligence-driven tools: A comprehensive survey on the evolution of feature extraction methods in paintings," *Eng. Appl. Artif. Intell.*, vol. 161, p. 111794, 2025.

[18] Coorporation, "Visual studio code documentation. Retrieved from visual studio code." Accessed: Jun. 12, 2026. [Online]. Available: https://code.visualstudio.com/

[19] J. Ezenarro, Á. García-Pizarro, O. Busto, A. de Juan, and R. Boqué, "Analysing olive ripening with digital image RGB histograms," *Anal. Chim. Acta*, vol. 1280, p. 341884, 2023.

[20] Python Org, "Python Documentation. Retrieved from Python." Accessed: Jun. 12, 2026. [Online]. Available: https://www.python.org/

[21] A. I. Awodeyi, O. A. Ibok, I. Omokaro, J. U. Ekwemuka, and M. O. Ighofiomoni, "Effective preprocessing techniques for improved facial recognition under variable conditions," *Franklin Open*, vol. 10, p. 100225, 2025.

[22] Y. Liu, X. Wang, Z. Zhang, and F. Deng, "Deep learning based data augmentation for large-scale mineral image recognition and classification," *Miner. Eng.*, vol. 204, p. 108411, 2023.

[23] M. Z. Li, G. Liu, Z. Mao, Q. S. Yang, and J. W. Gu, "Two-dimensional motion estimation using phase-based image processing with Riesz transform," *Mech. Syst. Signal Process.*, vol. 188, p. 110044, Apr. 2023, doi: 10.1016/j.ymssp.2022.110044.

[24] K. Veena, K. Meena, Y. Teekaraman, R. Kuppusamy, and A. Radhakrishnan, "C SVM classification and KNN techniques for cyber crime detection," *Wirel. Commun. Mob. Comput.*, vol. 2022, no. 1, p. 3640017, 2022.

[25] O. U. Lenz, H. Bollaert, and C. Cornelis, "A unified weighting framework for evaluating nearest neighbour classification," *Fuzzy Sets Syst.*, vol. 519, p. 109516, 2025.

[26] M. Osenberg *et al.*, "Classification of FIB/SEM-tomography images for highly porous multiphase materials using random forest classifiers," *J. Power Sources*, vol. 570, p. 233030, 2023.

[27] S. Viswanathan, N. V. Sridharan, J. Rakkiyannan, and S. Vaithiyanathan, "Brake fault diagnosis using a voting ensemble of machine learning classifiers," *Results in Engineering*, vol. 23, p. 102857, 2024.

[28] J. Boulos, V. Eglin, B. Kerautret, E. Larue, and J.-M. Côme, "Soil image classification and segmentation: A survey from deep learning, multimodal data and hybrid models," *Geodata and AI*, p. 100053, 2026.

[29] Y. Wu and J. Wan, "A survey of text classification based on pre-trained language model," *Neurocomputing*, vol. 616, p. 128921, 2025.