Engineering and Technology Journal e-ISSN: 2456-3358

Volume 11 Issue 08 August-2026, Page No.-11066-11071

DOI: 10.47191/etj/v11i08.03, I.F. – 8.595

Detection Of AI-Generated Images Using Pretrained CNN Feature Extraction and Support Vector Machine Classification

Hersatoto Listiyono[1*], Yunus Anis[2], Sri Mulyani[3], Vici Tiara Anjarsari[4], Vanyariska Indriani[5]

[1,2,3,4]Faculty of Vocational Studies, Universitas Stikubank, Semarang, Indonesia

[5]Faculty of Information Technology and Indutry, Universitas Stikubank, Semarang, Indonesia

ABSTRACT: Because Generative AI, we now see highly realistic synthetic images that are difficult to distinguish from the real thing. This technology raises concerns about misinformation, the trustworthiness of digital content, and how people can misuse synthetic media. Therefore, finding an accurate and efficient way to detect AI-Generated images is a significant research challenge. In this research, we explore a hybrid image classification setup: we combine CNN-based feature extraction (using a pre-trained network) with SVM classification to determine which images are AI-Generated and which are real. For feature extraction, we use MobileNetV2, pre-trained on ImageNet, as the fixed extractor. Then, to classify these features, we use SVM with an RBF kernel. This experiment uses the Kaggle dataset AI Generated Images vs real Images, with a total of 975 images divided into training and testing sets with a stratified 80:20 split. Our model performs quite well: 84.1% accuracy, 81.8% precision, 91.7% recall, and an F1 score of 86.5%. The confusion matrix shows that the model is able to detect most of the AI-generated images, demonstrating its sensitivity to synthetic content. Overall, our CNN–SVM setup offers an effective and efficient alternative to full end-to-end deep learning methods for detecting these images. This holds promise for applications such as digital content verification, combating disinformation, and automated authenticity checking.

KEYWORDS: AI-Generated image, CNN, Real Image, MobileNetV2, SVM.

I. INTRODUCTION

The development of Artificial Intelligence (AI) technology has driven the emergence of various generative models capable of producing synthetic images with increasingly high realism (Raharjo et al., 2025). Models such as Stable Diffusion, DALLE, and Midjourney enable image creation based on text descriptions that are visually difficult to distinguish from original images (Maulana et al., 2025). These advances bring broad implications for various creative sectors, including art and digital design, but also raise new problems related to visual authenticity, the spread of false information, and the potential misuse of technology for digital crime and defamation (Aria et al., 2025).

In a broader context, the use of AI-Generated images also poses serious challenges to legal and social aspects (Huwaida et al., 2025). Synthetic images have the potential to reduce trust in visual evidence and diminish its probative value in legal proceedings (Octalia & Librianti, 2025). Several studies indicate that AI-Generated images can be used to disseminate misinformation, including in political contexts and public opinion, thereby underscoring the urgency of developing reliable automatic detection systems (Hakim et al., 2024). This problem is exacerbated by the fact that people often have difficulty distinguishing genuine images from AI-made images, even when given specific visual cues (Subkhi et al., 2023).

Previous research has examined the application of machine learning and deep learning to image classification tasks. Ibrahim et al., in their research titled "Nominal Currency Detection in Images Using Convolutional Neural Networks", show that end-to-end CNN approaches can recognize visual patterns in currency images, but classification performance remains variable and tends to be unstable across some classes, influenced by labeling process limitations and model efficiency (Ibrahim et al., 2023). In a more specific context concerning AI-Generated image detection, Erwin et al., in the research "Implementation of DeiT Model to Distinguish AI-Generated Images and Human-made Images in 2D Animated Illustrations," report that transformer-based models like DeiT Base can achieve high accuracy and stable convergence but require substantial computational resources, while lighter variants show a drop in performance (Erwin et al., 2025). Additionally, Arrosyid et al., through the study "Comparative Analysis of CNN Models for Classifying Electronic Component Images," emphasize that model complexity does not always correlate with accuracy, where lightweight CNN architectures such as MobileNet can still provide competitive performance with better computational efficiency compared to larger models (Arrosyid et al., 2025). These findings indicate that although pure deep learning and transformer approaches have advantages in accuracy, there remains a

need for image classification methods that balance performance and computational efficiency.

This research uses a hybrid method that pulls features from images with a pretrained CNN, then hands those over to a Support Vector Machine (SVM) for classification. By doing this, we get the powerful visual abilities of CNNs along with the reliable performance of SVMs, especially when working with medium-sized datasets (Fitri et al., 2024).

The research grab features from images using a pretrained CNN, train an SVM to tell AI-Generated images apart from real ones, and then see how well the model does by looking at accuracy, precision, recall, and F1-Score. For data, we use the "AI Generated Images vs Real Images" set from Kaggle, which has both real web images and ones made by AI. Overall, this approach offers a lighter, more efficient alternative to running full deep learning models every time we want to spot an AI-Generated image.

II. METHODOLOGY

A. Research Design

In this research, we set out to build and test a framework that can tell apart AI-Generated images from real ones. The method brings together a pretrained CNN for pulling features from images and a SVM for doing the actual classifying. With this combo, we wanted to strike a good balance between strong results and reasonable computing time.

Here's how we went about it: First, we gathered the dataset. Next came preprocessing the images. After that, we focused on feature extraction, then trained the model, and finally checked how well everything worked. Each step let us dig deeper into whether our CNN–SVM hybrid could really detect AI-made images.

B. Dataset

The experiments were conducted using the publicly available AI generated Images vs Real Images dataset obtained from the Kaggle platform. The dataset contains two image categories: AI generated images and authentic images collected through web scraping. The image collection includes a wide variety of visual content, such as human portraits, animals, landscapes, artistic illustrations, and everyday scenes. This diversity introduces additional classification challenges and provides a more realistic environment for evaluating model robustness.

The dataset was downloaded using the Kagglehub library and organized according to the original directory structure provided by the dataset source. To ensure an unbiased evaluation process, the dataset was partitioned into training and testing subsets using a stratified sampling strategy. An 80:20 split ratio was adopted, allowing the class distribution to remain consistent across both subsets.

Table 1. Dataset Distribution

Class	Training Set (80%)	Testing Set (20%)	Total
AI – Generated	431	108	539
Real Images	348	88	436
Total	779	196	975

The use of stratified sampling helps maintain balanced class representation during model development and reduces the risk of classification bias caused by uneven data distribution.

C. Image Processing

Several preprocessing procedures were applied to ensure data consistency and compatibility with the selected CNN architecture. First, all images were resized to 224 × 224 pixels, corresponding to the input dimensions required by MobileNetV2. Subsequently, pixel values were normalized using the preprocess_input function specifically designed for the MobileNetV2 architecture.

After that, each image was then assigned a label according to its category. Authentic images were assigned a label of 0, whereas AI-generated images were assigned a label of 1. These preprocessing operations standardize the input data and facilitate effective feature extraction by reducing variations unrelated to image content.

D. Feature Extraction Using Pretrained CNN

We used MobileNetV2, pretrained on ImageNet, to extract features from our images. MobileNetV2 stands out because it strikes a nice balance between accuracy and speed—it manages to keep its model size small while still pulling in strong feature representations. Bigger CNNs usually eat up more resources, but with MobileNetV2, we get solid performance without all that overhead.

In our setup, the pretrained CNN was utilized as a feature extractor. The original classification layers were removed by setting include-top = False, and Global Average Pooling (GAP) was applied to the final convolutional feature maps to generate fixed-length feature vectors. No fine-tuning process was performed, allowing the pretrained weights to remain unchanged throughout the experiment.

These feature vectors capture all sorts of high-level details about each image things like textures, shapes, how objects are arranged, and even deeper semantic stuff. By tapping into a model that is already learned a lot from huge image datasets, we get those rich visual representations while keeping our computation light.

E. Feature Image Classification Using SVM

First, we used the extracted feature vectors as input for a Support Vector Machine (SVM) classifier. We picked SVM because it works well with high-dimensional data and handles medium-sized datasets without much trouble. To deal with the fact that the data isn't always linearly separable, We chose

the Radial Basis Function (RBF) kernel. This way, the SVM could find nonlinear boundaries between AI-generated and real images. While training, the SVM looked for the best margin to separate the two classes. After training, we tested the model on a separate dataset to see how well it could predict image labels and measure its accuracy.

By combining pretrained CNNs for feature extraction with SVM for classification, I got the best of both its. The CNNs pull out strong, layered features, and SVM brings reliable decision boundaries along with efficient processing. Altogether, this setup is a solid fit for real-world image classification tasks

F. Model Evaluation

We tested how well the proposed framework worked using the testing dataset. To really see how good it was at classifying, we used at four main metrics: accuracy, precision, recall, and F1-Score. We also checked out a confusion matrix, which helped us see where the model got things right and where it slipped up for each class (Hardiyanto & Akbar, 2025).

The confusion matrix lays out everything in terms of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). So with these numbers, we could dig into the model's actual prediction habits—not just how often it got things right overall.

We figured out classification accuracy by dividing the number of correct predictions by the total number of test samples, as shown in Equation (1).

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision(2) measures the correctness of positive predictions.

$$\textbf{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall(3) represents the model's ability to correctly identify all AI-generated images.

$$\textbf{Recall} = \frac{TP}{TP+FN} \quad (3)$$

F1-Score (4) is the harmonic mean of precision and recall.

$$\textbf{F1 - Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

III. RESULTS AND DISCUSSION

A. Dataset Processing Results

We used the AI-Generated Images vs Real Images dataset from Kaggle for this research. It includes two types of images: ones created by AI and real ones scraped from the web. After checking the dataset and rearranging the folders, we ended up with 975 images ready for our tests.

We used the AI-Generated Images vs Real Images dataset from Kaggle for this research. It includes two types of images: ones created by AI and real ones scraped from the web. After checking the dataset and rearranging the folders, we ended up with 975 images ready for our tests.

To ensure balanced representation across classes, the dataset was divided into training and testing subsets using a stratified 80:20 split. This process resulted in 779 training images and 196 testing images, preserving the original class distribution and minimizing potential sampling bias during model evaluation

The complete research workflow is illustrated in Figure 1. The pipeline begins with dataset acquisition and folder identification, followed by image preprocessing, feature extraction using a pretrained CNN model, SVM training, testing, and final performance evaluation.

Figure 1 shows how the framework works: it first uses deep learning (the CNN) for feature extraction, then passes those features to a traditional SVM classifier, all in a simple step-by-step process. This approach lets us use the strong feature extraction of CNNs and the efficiency of SVMs at the same time.

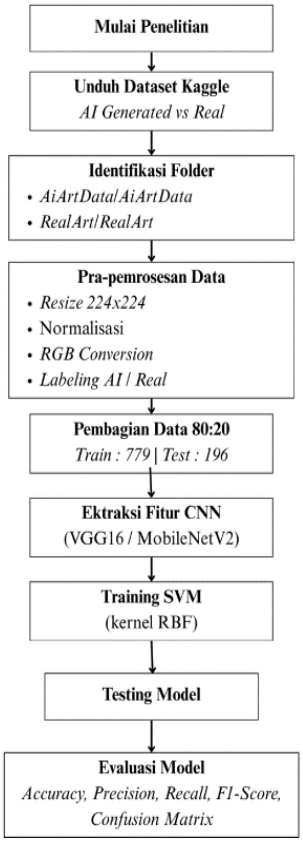

Figure 1. Research Workflow

B. Feature Extraction and Classification Results

We used a pretrained MobileNetV2 model to pull features from the images. We took MobileNetV2 as it is and let it do the heavy lifting—turning each image into a detailed set of features (Santosa et al., 2025).

Before feeding the images in, we made sure to resize them, convert them to RGB, and normalize the colors. This step was crucial so everything fit neatly into the MobileNetV2 pipeline and to smooth out differences in size or pixel values across images.

At the end of this process, we had feature vectors, each loaded with information about textures, edges, objects and some detailed meaning from each picture. These vectors became the input for a Support Vector Machine (SVM) with an RBF kernel. We picked SVM because it handles high-dimensional data well and draws flexible decision boundaries that aren't just straight lines.

Combining a pretrained CNN as a feature extractor and an SVM for classification lets us work with smaller datasets and keeps the process efficient. This way, we dodge the huge computational demands. That is an advantage when we do not have many resources or we need to get results quickly.

C. Model Performance Evaluation

The performance of the proposed model was evaluated using four standard classification metrics: accuracy, precision, recall, and F1-score. The performance evaluation results are presented in Table 2.

Tabel 2. Performance Evaluation Result

Metric	Score
Accuracy	0.841
Precision	0.818
Recall	0.917
F1-Score	0.865

The results show the model reached an overall accuracy of 84.1%, which means it can tell apart AI-generated images from real ones well.

One thing that stands out is the model's high recall score 91.7%. In plain terms, it caught most of the AI-generated images in the test set. That is very importantl, especially in areas like misinformation detection, digital forensics or content verification. Missing fake images in these cases can cause more trouble.

The precision score of 81.8% suggests that although the model occasionally classified real images as AI-generated, the majority of positive predictions remained correct. Meanwhile, the F1-score of 86.5% reflects a balanced trade-off between precision and recall, indicating stable overall classification performance.

All of this points to one thing, the hybrid approach does not just hold its own against bigger, more complex deep learning models, but also does so while using a lot less computational power.

D. Confusion Matrix Analysis

To gain deeper insight into the model's behavior, a confusion matrix was analyzed, as shown in Figure 2.

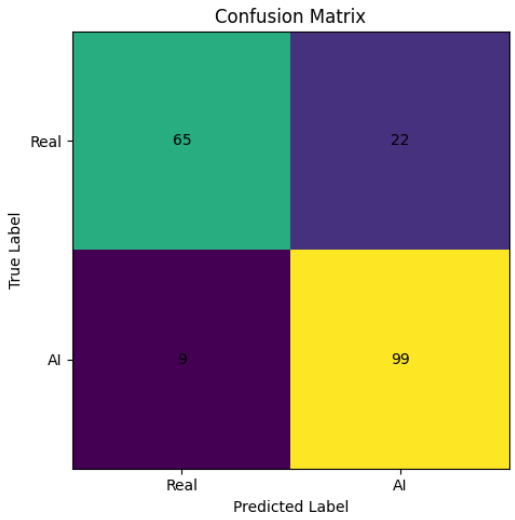

Figure 2. The Confusion Matrix

Based on the confusion matrix, the classifier correctly identified 99 out of 108 AI-generated images and 65 out of 87 authentic images. Meanwhile, 22 authentic images were incorrectly classified as AI-generated, and only 9 AI-generated images were misclassified as authentic.

Tabel 3. Confusion Matrix Interpretation

Actual Class	Predicted Real	Predicted AI
Real Image	65	22
AI-Generated	9	99

Table 3 reveals an important characteristic of the proposed model: it exhibits a stronger tendency to detect AI-generated images than authentic images. This behavior explains the high recall score observed during evaluation.

From a practical perspective, this tendency can be considered beneficial because the primary objective of AI-generated image detection systems is to minimize the number of synthetic images that remain undetected. The model successfully reduced false negatives to only nine instances, indicating strong sensitivity toward AI-generated content.

However, the presence of 22 false positives suggests that some real images share visual characteristics commonly found in AI-generated images. This phenomenon may be attributed to advanced image editing, stylized photography, compression artifacts, or visual patterns that resemble those produced by AI-generated.

The imbalance between false positives and false negatives suggests that the extracted MobileNetV2 features could effectively capture synthetic-image signatures.

So, with more false alarms than misses, we can see how MobileNetV2 recognize on the features typical of AI creations. The classification does not take chances; when it's not sure, it's safer to flag something as AI-generated than to let a fake slide through.In jobs like digital forensics or content moderation, that a few real images caught by mistake are a much smaller problem than missing a well-crafted fake.This

behavior is desirable in security-oriented applications, such as content moderation and digital forensic analysis, where failing to detect synthetic content may result in greater consequences than generating occasional false alarms.

E. Discussion

This research backs up what earlier work has shown: pretrained CNNs do a great job as feature extractors for image classification.Unlike end-to-end deep learning approaches that require extensive training data and computational resources, the proposed CNN–SVM framework achieves satisfactory performance using a relatively small dataset of 975 images.

The achieved accuracy of 84.1% indicates that discriminative visual information extracted by MobileNetV2 remains sufficiently informative for separating AI-generated and authentic images. Furthermore, the high recall value suggests that AI-generated images contain detectable visual patterns that can be captured by pretrained CNN representations even without fine-tuning.

This research also proves that lightweight hybrid models still matter, even as transformer models grab the spotlight. Sure, transformers often hit top scores, but they demand a lot more computing muscle. This approach gives you a much more efficient way to get strong results without all the hardware.

Therefore, the results suggest that combining pretrained CNN feature extraction with traditional machine learning classifiers represents a practical and scalable approach for AI-generated image detection, particularly in resource-constrained environments.

IV. CONCLUTION

This research proposed a hybrid image classification framework that combines pretrained CNN-based feature extraction with Support Vector Machine (SVM) classification for distinguishing AI-generated images from authentic images. By leveraging the representational power of MobileNetV2 and the classification capability of SVM, the proposed approach was designed to provide an effective yet computationally efficient alternative to fully end-to-end deep learning models.

Experimental results demonstrated that the proposed framework achieved an accuracy of 84.1%, a precision of 81.8%, a recall of 91.7%, and an F1-score of 86.5% on the AI Generated Images vs Real Images dataset. These results indicate that the model is capable of reliably identifying synthetic images while maintaining balanced overall classification performance. In particular, the high recall value highlights the model's strong ability to detect AI-generated content, which is crucial for applications involving digital content verification, misinformation prevention, and media authenticity assessment.

The confusion matrix analysis further revealed that the proposed model successfully identified the majority of AI-generated images, although several authentic images were incorrectly classified as synthetic. This finding suggests that the extracted CNN features effectively capture discriminative visual patterns associated with AI-generated content. Moreover, the results demonstrate that combining pretrained deep feature extraction with a traditional machine learning classifier can provide competitive performance while requiring considerably fewer computational resources than many transformer-based and end-to-end deep learning approaches.

From a practical perspective, the proposed framework offers a lightweight and scalable solution that can be integrated into automated systems for detecting AI-generated visual content on digital platforms, social media environments, and online information ecosystems. The research therefore contributes to ongoing efforts aimed at improving the reliability and trustworthiness of digital visual information in the era of generative artificial intelligence.

Despite these promising results, several limitations should be acknowledged. The experiments were conducted using a single publicly available dataset, and the model was evaluated using only one pretrained CNN architecture and one machine learning classifier. Future research may investigate larger and more diverse datasets, explore alternative feature extraction architectures, including EfficientNet and Vision Transformers, and examine fine-tuning strategies to further improve classification performance and generalization capability. In addition, comparative studies involving multiple machine learning classifiers and cross-dataset validation would provide a more comprehensive assessment of the robustness of AI-generated image detection systems.

REFERENCES

1. Aria, M. S. A., Slamet, C., Firdaus, M. D. (2025). Klasifikasi Fake dan Real Menggunakan Vision Transformer dan EfficientNet-B0 pada Gambar Asli dan Generatif AI. SMATIKA, 15(1), 179–192. https://doi.org/10.32664/smatika.v15i01.1531
2. Arrosyid, M. Z., Hermawan, A. & Sutarman, S. (2025). Analisis Perbandingan Model CNN Terhadap Klasifikasi Citra Komponen Elektronika. ALGORITME, 5(2), 152–163. https://doi.org/10.35957/algoritme.v5i2.10259
3. Santosa, D. B., Wahana, A., Uriawan, W. (2025). Implementation Of Convolutional Neural Network Using Mobilenetv2 To Distinguish Human And Artificial Intelligence Mobilenetv2 Untuk Membedakan Hasil Karya Seni Lukis Manusia. JUTIF , 6(1), 441–452. https://doi.org/10.52436/1.jutif.2025.6.1.3827
4. Subkhi, M. B., Setiawan, A. B., Chandra, M. Y. A. (2023). Klasifikasi Gambar : Membedakan Lukisan Buatan Manusia dan AI dengan CNN. PARADIGMA, 29(4), 149–155.

https://doi.org/10.33503/paradigma.v30i4.1284

5. Erwin, I. T., Arda, A. L., Taufik, I., rosyadi, S, M. E. R. & Erwin, H. A. (2025). Implementasi Model Deit Untuk Membedakan Gambar Buatan Ai Dan Manusia Pada Ilustrasi Animasi 2D. INTI, 19(2), 172–180. https://doi.org/10.33480/inti.v19i2.6306
6. Fitri, S. D., Lestari, D., Bintana, R. R., Aryani, R, Ilhami M. & Noverina Y. (2024). Implementasi Model Support Vector Machine Dalam Analisa Sentimen Masyarakat Mengenai Kebijakan Penerapan Aplikasi Mypertamina. BRIDGE, 2(2), 176–193. https://doi.org/10.62951/bridge.v2i3.180
7. Hakim, S. A., Ubaidillah, M., Ramadhan, A. R., Hawari, R. Z. A., Rizky, A. B., Lutfi, R., Hermanto, P. T. M. & Yudistira, N. (2024). Klasifikasi Citra Generasi Artificial Intelligence Menggunakan Ai Generated Image Classification Using Fine Tuning On Residual. JTIIK, 11(3), 655–666. https://doi.org/10.25126/jtiik.938118
8. Raharjo, T., Adristi, F. I., Romadhona, E. Y., Syahputra, R. S., Halim, M. Y., Rachman, M. A., Marjianto, R. N., Santycho, D., Kusuma, P. & Ramadhani, E. (2025). Analisis Forensik Deepfake Berbasis Convolutional Neural Network (Cnn) Untuk Deteksi Inkonsistensi. JATI, 9(2), 2731–2738. https://doi.org/10.36040/jati.v9i2.13058
9. Hardiyanto, A. & Akbar, M., (2025). Klasifikasi Citra Sintetis Hasil Model Difusi Menggunakan Gray Level Co-Occurrence Matrix (GLCM) Dan Convolutional Neural Network (CNN). JIKO, 9*(3)*,572–582. https://doi.org/10.26798/jiko.v9i3.2033
10. Huwaida, F. S. , Widodo, S., & Elviani, U. (2025). Penyalahgunaan Deepfake dan Tantangan Tata Kelola AI: Tinjauan Literatur Sistematis. EduTIK, 5(6), 2590–2604.
11. Ibrahim, M. M., Rahmadewi, R., Nurpulaela, L. (2023). Pendeteksian Nominal Uang Pada Gambar Menggunakan Convolutional Neural Network : Integrasi Metode Pra- Pemrosesan Citra Dan Klasifikasi Berbasis Cnn. JATI, 7(2), 1395–1400. https://doi.org/10.36040/jati.v7i2.6863
12. Maulana, D., Azzahrah, N. S., Syach, R. A., Syauri, S., Syafi'i, M. & Rozikin, C. (2025). Identifikasi Konten Visual Buatan Ai Dengan Resnet Dan Fine-Grained Feature Extraction. Jurnal Informatika dan Teknik Elektro Terapan, 13(3), 1248-1256. https://doi.org/10.23960/jitet.v13i3S1.8056
13. Octalia, E., & Librianti, I. (2025). Deepfake Dalam Komunikasi Politik : Tantangan Etika dan Aspek Hukum Dalam Era Artificial Intelligence. Siyasah 05(2), 222–238. https://doi.org/10.32332/siyasah.v4i1
14. Santosa, D. B., Wahana, A., Uriawan, W. (2025). Implementation Of Convolutional Neural Network Using Mobilenetv2 To Distinguish Human And Artificial Intelligence Mobilenetv2 Untuk Membedakan Hasil Karya Seni Lukis Manusia. JUTIF , 6(1), 441–452. https://doi.org/10.52436/1.jutif.2025.6.1.3827