Received: 15/04/2025 | Revised: 12/03/2026 | Accepted: 04/04/2026

DOI: xxxxx/xxxxx

ORIGINAL RESEARCH

Deep Learning Meets Deployment: EfficientNetB0-Based Real vs. AI-Generated Fake Image Classification via a Web Interface

Hasibul Islam Peyal *[1], Nusrat Jahan[2], Ovishek Mohanta[3], Md. Ismiel Hossen Abir[4]

1 Senior Lecturer, Department of Computer Science and Engineering, International Standard University, Dhaka, Bangladesh.
Email: peyal@isu.ac.bd
2 Senior Lecturer, Department of Computer Science and Engineering, International Standard University, Dhaka, Bangladesh.
3 Department of Electrical & Computer Engineering, Rajshahi University of Engineering & Technology.
4 Department of Computer Science and Engineering, International Standard University

* Corresponding author

Abstract

The rapid advancement of artificial intelligence has led to the widespread generation of highly realistic AI-generated images, raising concerns about digital authenticity and misinformation. Detecting such synthetic images has become an important challenge for researchers and digital platforms. This research presents a deep learning–based approach for distinguishing between real and AI-generated images using the EfficientNetB0 model. The dataset used in this study consists of real and AI-generated images collected from publicly available sources and is divided into training and testing sets. To improve generalization, several data augmentation techniques were applied, including rotation, flipping, and scaling. In addition to EfficientNetB0, several popular convolutional neural network models, including ResNet-50, VGG-16, MobileNetV2, DenseNet-121, and InceptionV3, were evaluated for performance comparison. Experimental results show that EfficientNetB0 achieved the highest classification accuracy of 97.59%, outperforming the other models on the dataset. To demonstrate practical applicability, the trained model was deployed through a locally hosted web interface that allows users to upload images and receive instant predictions regarding their authenticity. The proposed system provides an effective solution for detecting AI-generated images and highlights the potential of deep learning models to address challenges in synthetic media and digital content verification.

Keywords: AI-generated image detection, Real vs. Fake Image Classification, EfficientNetB0, Image Authenticity, Convolutional Neural Networks (CNN), Flask Framework, Deep Learning.

1 | INTRODUCTION

The rapid evolution of artificial intelligence has significantly transformed the field of digital media creation. In particular, recent advances in generative models, such as diffusion-based systems and generative adversarial networks, have made it possible to produce highly realistic synthetic images. As these technologies continue to improve, distinguishing between real photographs and AI-generated images has become increasingly difficult. This capability raises important concerns regarding digital authenticity, misinformation, intellectual property protection, and ethical use of visual content. Consequently, the development of reliable methods for detecting AI-generated images has become an important research challenge for both researchers and digital content platforms.

Several approaches have been proposed to address this problem. Earlier studies focused on identifying pixel-level inconsistencies and statistical artifacts present in synthetic images. For example, Thakre et al. applied feature extraction techniques such as Photo Response Non-Uniformity (PRNU) and Error Level Analysis (ELA) to detect anomalies in AI-generated images. These features were then used as inputs for convolutional neural networks (CNNs), achieving an accuracy above 95%. While this approach demonstrates the usefulness of pixel-level forensic analysis, it relies heavily on detectable artifacts, which may become less reliable as modern generative models increasingly minimize such inconsistencies [1].

More recent studies have explored deep learning–based feature learning approaches. Hossain et al. combined CNN architectures with Vision Transformers to detect synthetic images and achieved an accuracy of 96.31% on the CIFAKE dataset. Their work also employed Grad-CAM to visualize the regions influencing model predictions, providing insight into the decision-making process. However, the model was evaluated only on a single dataset, leaving questions about its ability to generalize to other datasets or newer generative techniques [2]. Similarly, Chinta et al. investigated several deep learning architectures, including CNN, VGG-19, and ResNet-50, using the AI-ArtBench dataset.

Their results indicated that CNN achieved the highest accuracy of 92.69%. Although the study highlights the role of deep learning in detecting synthetic artwork, the relatively limited dataset size may restrict the scalability and robustness of the model in real-world scenarios [3].

Bird and Lofti proposed a CNN-based classifier trained on the CIFAKE dataset and reported an accuracy of 92.98%. Their Grad-CAM analysis suggested that background imperfections often serve as discriminative features when distinguishing AI-generated images. While such observations are valuable, reliance on background artifacts may reduce model reliability as modern generative models produce increasingly coherent and artifact-free backgrounds [4]. In another study, Purohit et al. trained CNN models on mixed datasets of real and AI-generated images, achieving classification accuracies between 81% and 88%. Their work highlights the relevance of AI-generated image detection for addressing cybersecurity threats and misinformation; however, the relatively modest accuracy values indicate the need for more effective architectures and improved training strategies [5].

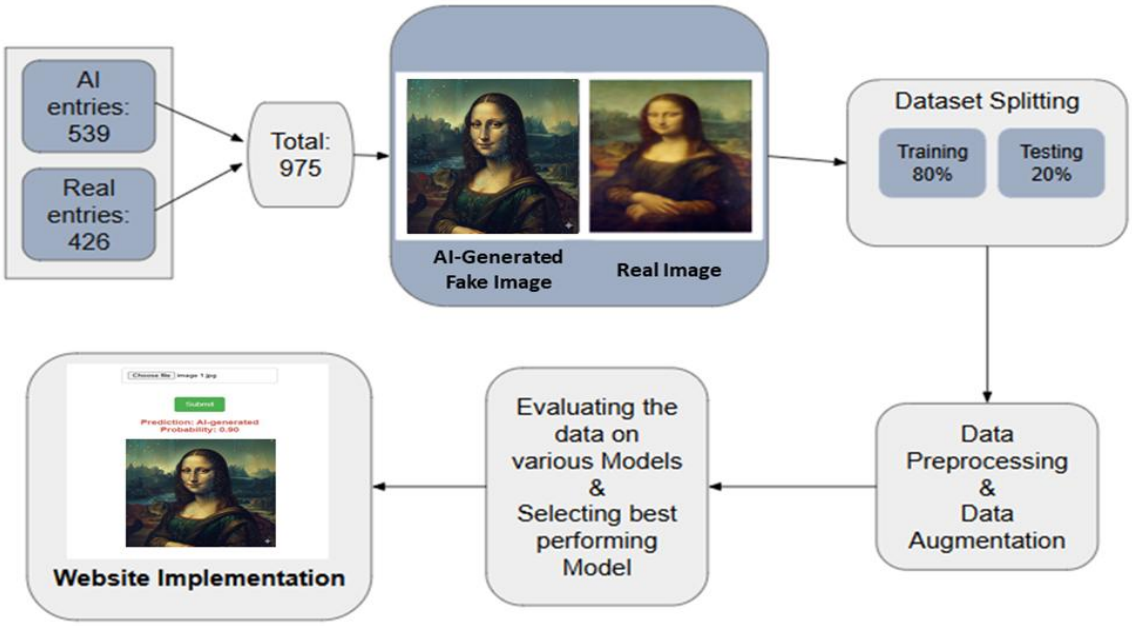

Fig. 1. Working Procedure

Although these studies demonstrate the potential of deep learning techniques for detecting AI-generated images, several research gaps remain. First, many studies focus primarily on achieving high accuracy on specific datasets without providing comprehensive comparisons across multiple deep learning architectures under consistent experimental settings. Second, methodological discussions in prior works often emphasize performance metrics rather than analyzing architectural behavior and suitability for detecting subtle synthetic patterns. Finally, most existing studies remain limited to experimental environments and rarely demonstrate practical deployment of the detection models in real-world applications.

To address these limitations, this research performs a comprehensive evaluation of several widely used convolutional neural network architectures for AI-generated image detection. The models considered in this study include ResNet50, VGG16, MobileNetV2, DenseNet121, InceptionV3, EfficientNetB0, and a custom CNN model. Each model is trained and evaluated using the same dataset and experimental settings to ensure a fair comparison of their classification performance and behavior.

The main contributions of this study are summarized as follows:

(I) Comparative evaluation of multiple CNN architectures under consistent experimental conditions for detecting AI-generated images.

(II) Customization and optimization of the EfficientNetB0 architecture to improve classification performance for this specific task.

(III) Analysis of architectural performance differences to better understand how different CNN models handle AI-generated image detection.

(IV) Practical deployment of the best-performing model through a web-based interface, enabling users to upload images and receive real-time authenticity predictions.

The remainder of this paper is organized as follows. Section 2 describes the methodology adopted in this research, including dataset preparation, data preprocessing, augmentation strategies, and the deep learning architectures used for the experiments. Section 3 presents the experimental results and performance analysis of the models. Section 4 provides a comparative analysis with previous studies in AI- generated image detection. Finally, Section 5 concludes the paper and outlines possible directions for future research

2 | METHODOLOGIES

In this study, the EfficientNetB0 model was proposed to identify whether the image is AI-generated or real. The working procedure of our work is shown in Figure 1.

2.1 | Dataset collection and splitting

The dataset used in this study consists of 21,600 images, including 10,800 AI-generated images and 10,800 real images. The dataset was obtained from a publicly available repository on Kaggle [6], where images are labeled according to their source class.

The AI-generated images were created using modern generative models such as diffusion-based generators and other AI image synthesis techniques available in the dataset. The real images were collected from publicly available image datasets containing natural photographs. The dataset includes im ages from diverse categories, ensuring variability in objects, scenes, and visual styles. This diversity helps improve the robustness and generalization ability of the trained model.

Before training, all images were preprocessed and resized to a fixed resolution of 150 × 150 pixels to maintain consistency across different architectures.
To evaluate model performance, the dataset was divided into training and testing sets using an 80:20 split, while maintaining equal class distribution in both subsets. As a result:

Training set: 17,280 images (8,640 per class)
Testing set: 4,320 images (2,160 per class)

To further improve reliability, stratified sampling was applied during the split to ensure that both classes were evenly represented in each subset. Samples of AI-generated and real images are shown in Figure 2.

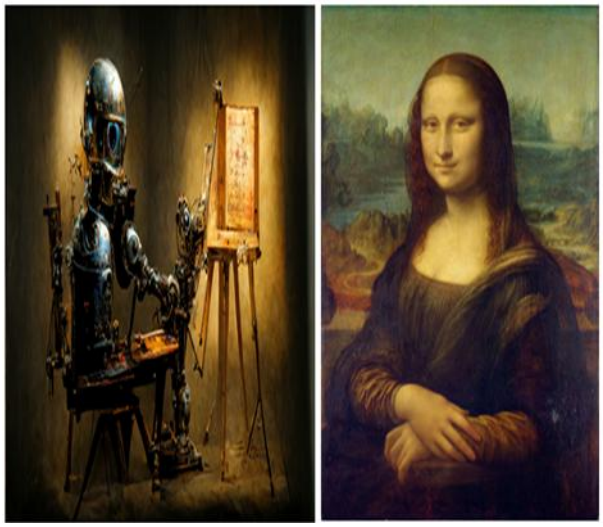

Fig. 2. AI-Generated Fake and Real Image

2.2 | Data Augmentation

Data augmentation was performed to increase the size and diversity of the training dataset, enabling the model to perform better on new and unseen data. Several augmentation techniques, including rotation, width shift, height shift, shear, zoom, and horizontal flipping, were applied. Augmentation was applied only to the training dataset, while the testing dataset remained unmodified. To ensure consistent performance, all images were resized to 150 × 150 pixels.

2.3 | Performance Comparison & Proposed Model

The following models were used for the performance comparison.

ResNet50 is a deep neural network with 50 layers that employs shortcut connections to address the vanishing gradient problem. This architecture facilitates the training of very deep networks and is widely used for tasks such as image classification and object detection.

VGG16 is a 16-layer neural network known for its simple and uniform design, utilizing small 3 × 3 convolutional filters. It is effective at learning hierarchical features but requires substantial computational resources and is commonly used for transfer learning in image-related tasks.

MobileNetV2 is a lightweight network designed for mobile and embedded devices. It employs depth-wise separable convolutions to achieve faster computation while consuming less power, making it suitable for real-time image processing applications.

DenseNet121 connects each layer to all preceding layers, enabling efficient feature reuse throughout the

network. This architecture achieves high accuracy with fewer parameters and is widely applied in tasks such as medical image analysis.

InceptionV3 uses specialized modules that process images at multiple scales, enabling the capture of rich feature representations while maintaining network efficiency. It performs well in image recognition tasks and helps reduce computational requirements.

EfficientNetB0 balances network depth, width, and input resolution to create a model that is both fast and accurate. It is commonly used in applications that require strong performance under limited computational resources.

A custom CNN is designed for a specific task, offering flexibility in architectural design. This approach enables the development of efficient solutions tailored to unique problems such as object detection or image segmentation.

Based on the described models, EfficientNetB0 was selected as the proposed model for this task due to its balanced scaling of depth, width, and resolution. This model provides an optimal trade-off between speed and accuracy, making it well-suited for applications requiring high performance with limited resources. Compared to other models, EfficientNetB0 is recognized for its high efficiency, achieving superior classification accuracy while maintaining relatively low computational complexity. Its ability to capture complex features and process images effectively makes it particularly suitable for distinguishing between real and AI-generated fake images in this study.

2.4 | Website Implementation

Various Python-based frameworks have been developed over the years to deploy machine learning and deep learning models. Frameworks such as FastAPI and Flask have made it convenient to work with trained models by providing user-friendly interfaces and easy accessibility, even for beginners. In this work, Flask was selected due to its lightweight nature and flexible development environment.

An interface was created to accept input images from users, which were then processed by the loaded EfficientNetB0 model. The model produced two probability values representing the likelihood of the image belonging to each class. These probabilities were compared to determine the final classification result, which was sent to the server as JSON data. Finally, the result was displayed on the same webpage. Figure 3 illustrates the website's working procedure.

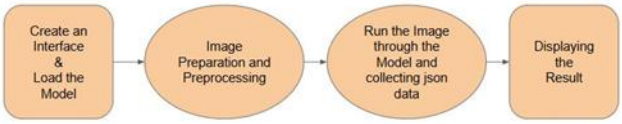

Fig. 3. Website Work Procedure

3 | EXPERIMENTAL RESULTS

3.1 | Accuracy Curve Analysis

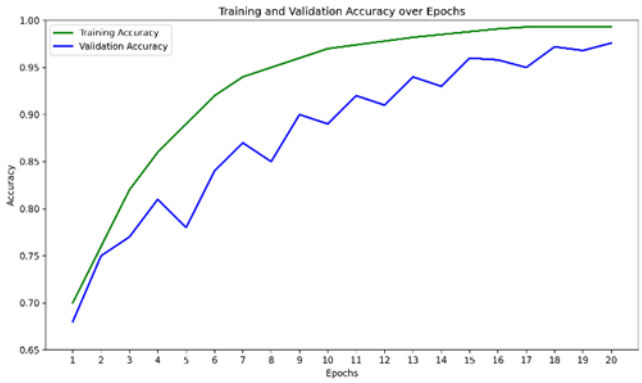

Fig. 4. Accuracy Curve

The training and validation accuracy curves in Figure 4 demonstrate the learning behavior of the model over the course of the training process. The training accuracy steadily increased and stabilized around 99.32%, indicating that the model effectively learned the underlying patterns within the training dataset. The validation accuracy exhibited some fluctuations, which is expected due to the model's performance being evaluated on unseen data at each epoch. Despite these fluctuations, the validation accuracy consistently remained high, ultimately reaching approximately 97.59%. This indicates good generalization capability of the model, with minimal overfitting. The gap between training and validation accuracy is small, further confirming that the model is neither underfitting nor severely overfitting. Overall, the accuracy curves validate the robustness and reliability of the trained model for the binary classification task between AI-Generated fake and real images.

3.2 | Confusion Matrix Analysis

The confusion matrix in Figure 5 was computed on the test dataset, which comprised 20% of the total images from each class—2,160 images for both the AI-generated fake and real categories, totaling 4,320 test samples. The matrix illustrates the classification performance of the model by showing the counts of true positives, true negatives, false positives, and false negatives. The

model achieved an overall accuracy of approximately 97.59% on the test set, correctly classifying the majority of samples. The high true positive and true negative counts indicate the model's strong capability in correctly identifying both real and AI-generated fake images. The relatively low number of false positives and false

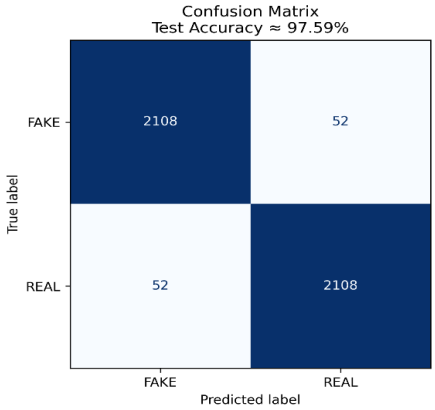

Fig. 5. Confusion Matrix

negatives suggests minimal misclassification errors. These results demonstrate that the proposed model generalizes well to unseen data, maintaining balanced and robust performance across both classes. The confusion matrix validates the effectiveness of the model in accurately distinguishing between AI-generated fake and real images, supporting the reported accuracy metrics.

3.3 | Classification Report Analysis

The performance of the proposed model was evaluated using standard classification metrics, namely Accuracy, Precision, Recall, F1-score, and Support, which are defined as follows:

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

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

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

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

$$\text{Support} = \text{TP} + \text{FN} \quad (5)$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively. Support indicates the total number of actual instances of a given class in the test dataset.

Table 1 presents the precision, recall, and F1-score metrics for the AI-generated fake and real classes evaluated on the test dataset, which comprises 20% of the original dataset (2,160 images per class). The model achieved a precision of 0.98 and recall of 0.97 for the fake class, indicating that 98% of predicted fake images were correct and 97% of actual fake images were successfully identified. For the real class, the precision and recall were 0.97 and 0.98 respectively, reflecting similarly strong performance. The F1-score, which balances precision and recall, was 0.98 for both classes, demonstrating a consistent and robust classification capability. The support values confirm an equal number of test samples per class, ensuring a balanced evaluation.

These results indicate that the proposed model effectively discriminates between AI- Generated fake and real images with minimal classification errors, supporting its suitability for practical deployment in binary image classification tasks.

3.4 | Implementation of Locally Hosted Website

Figure 6 illustrates the user interface design of a locally hosted website that facilitates interaction with the system. Through this graphical interface, users can upload an image, which is then displayed alongside a prediction indicating whether it is an AI-generated fake or a real image.

Figure 7 presents an example of a correctly identified real image, while Figure 8 illustrates the identification of an AI-generated fake image. This implementation enhances user accessibility and makes the research findings practical by providing a functional tool for real-world use. The website serves as a bridge between research and application, allowing users to easily test and explore the model's predictions. According to Figure 7 and 8, it shows that the EffieceintB0 model correctly identified the images.

Fig. 6. Website User Interface

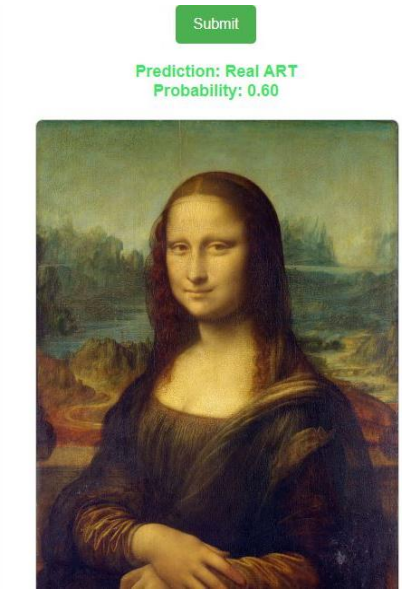

Fig. 7. Identified Real Image

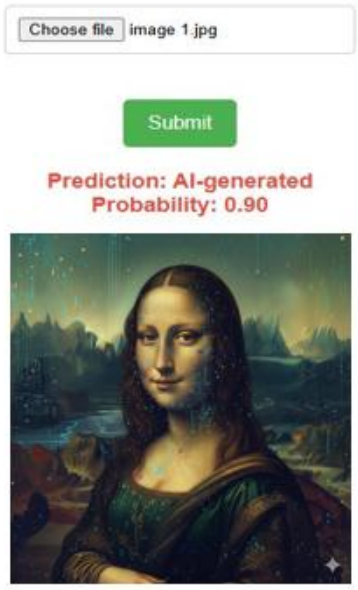

Fig. 8. Identified AI-Generated Fake Image

4 | COMPARATIVE ANALYSIS

Table 2 compares the performance of different models on our dataset based on their accuracy in classifying images. Among the models tested, EfficientNetB0 achieved the highest accuracy of 97.59%, showcasing its effectiveness for this task. InceptionV3 followed with 75.26%, while ResNet50 demonstrated a decent performance at 73.46%. In contrast, MobileNetV2 and DenseNet121 had relatively lower accuracies of 62.24% and 64.28%, respectively, indicating the varying capabilities of these models in handling the dataset.

Table 1. Classification Report of the Proposed Model

Class	Precision	Recall	F1-Score	Support
AI-generated fake	0.98	0.97	0.98	2160
Real	0.97	0.98	0.98	2160
Average / Total	**0.98**	**0.98**	**0.98**	4320

After the literature review in the introduction section, it was identified that all previous works achieved good accuracy scores. However, a key gap was found in real-world implementation. In this research, an accuracy score of 97.59% was achieved by the proposed EfficientNetB0 model, which is slightly lower than that of other works. Nevertheless, in the real-world scenario through the website implementation, it was observed that nearly all images were correctly identified.

Table 2. Performance comparison of different models in our dataset

Models	Accuracy (%)
ResNet50	73.46
VGG16	68.36
MobileNetV2	62.24
DenseNet121	64.28
InceptionV3	75.26
Custom CNN Model	70.31

5 | COMPARATIVE ANALYSIS WITH OTHER MODELS

To better understand the performance of our model, Table 3 compares our results with existing approaches from previous studies in the field of AI-generated image detection.

Table 3. Performance comparison of different models

Source	Methods	Accuracy (%)
[1]	CNN + PRNU & ELA	>95.00
[2]	CNNs & Vision Transformers	96.31
[3]	CNN, VGG-19, ResNet-50	92.69
[4]	CNN + Grad-CAM	92.98
[5]	CNNs trained on mixed AI/real images	81–88
	EfficientNetB0	**97.59**

Table 3 shows that while previous studies achieved strong results, such as 96.31% with Vision Transformers [2] and over 95% using PRNU & ELA [1], our proposed EfficientNetB0 model outperformed all with an accuracy of 97.59%. Unlike earlier works, which often lacked real-world application, our model was also deployed on a functional web platform, making it both highly accurate and accessible for practical use.

6 | CONCLUSION

This research demonstrated the effectiveness of the EfficientNetB0 model in distinguishing AI-generated fake images from real images, achieving an accuracy of 97.59%, the highest among all the models tested in this study. A key contribution of this work is the improvement of the pre-trained EfficientNetB0 model through customization, which significantly enhanced its classification performance on the chosen dataset. Additionally, a detailed comparative analysis was conducted across several popular deep learning models, including ResNet50, VGG16, MobileNetV2, DenseNet121, InceptionV3, and a custom CNN, providing valuable insights into their behavior and suitability for AI-generated image detection.

Another major contribution of this research is the successful real-world deployment of the best-performing model through a functional website. This implementation bridges the gap between theoretical research and practical application by allowing users to upload images and receive instant predictions, thereby making the system accessible to a wider audience such as artists, consumers, and digital content platforms concerned with image authenticity.

Despite its strong performance, the model has certain limitations. The dataset was relatively small, which may affect generalization to newer or more diverse AI-generated image styles. Although real-world testing showed promising results, evaluating the model on larger and more varied datasets would further improve its reliability.

Future work may explore ensemble approaches, transformer-based architectures, and explainability methods to further enhance both accuracy and interpretability. Improving the website interface and expanding its features could also increase the practical usability of the system. Overall, this research not only achieves high classification performance but also contributes a deployable tool that brings AI-generated image detection closer to everyday real-world use.

References

[1] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed., Pearson, 2021.

[2] M. Holmes, H. Bialik, and C. Fadel, Artificial Intelligence in Education: Promises and Implications for Teaching and Learning, Boston: Center for Curriculum Redesign, 2019.

[3] UNESCO, "Artificial Intelligence in Education: Challenges and Opportunities for Sustainable Development," 2019. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000366994

[4] B. Woolf et al., "AI Grand Challenges for Education," AI Magazine, vol. 34, no. 4, pp. 66–84, 2013.

[5] R. Purohit, Y. Sane, Devashree Vaishampayan, Sowmya Vedantam, and M. Singh, “AI vs. Human Vision: A Comparative Analysis for Distinguishing AI-Generated and Natural Images,” Jan. 2024, doi: https://doi.org/10.1109/icaect60202.2024.10469620.

[6] jeevans13, “ai image classifier,” Kaggle.com, May 17, 2025. https://www.kaggle.com/code/jeevans13/ai-image-classifier/input (accessed Jul. 05, 2025).