Received: xxxxxxxx | Revised: xxxxxxxx | Accepted: xxxxxxx

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]

[1]Department of Computer Science and Engineering, International Standard University

Nusrat Jahan[2]

[2]Department of Computer Science and Engineering, International Standard University

Ovishek Mohanta[3]

[3]Department of Electrical & Computer Engineering, Rajshahi University of Engineering & Technology

Md. Ismiel Hossen Abir[4]

[4]Department of Computer Science and Engineering, International Standard University

Abstract

This research focuses on distinguishing between real images and AI-generated fake images—a task of growing importance in the age of synthetic media. Accurate identification of such images holds value for a broad range of users, including digital content creators, consumers, and platforms concerned with authenticity, misinformation, and intellectual property. Multiple deep learning models were evaluated in this study. The performance results are as follows: ResNet50 achieved an accuracy of 73.46%, VGG16 scored 68.36%, MobileNetV2 reached 62.24%, DenseNet121 obtained 64.28%, and InceptionV3 achieved 75.26%. Notably, EfficientNetB0 outperformed all others with an accuracy of 97.59%. Additionally, a custom CNN model developed specifically for this task achieved an accuracy of 70.31%. These findings demonstrate that EfficientNetB0 is highly effective for image authenticity classification, making it a strong candidate for real-world applications. Furthermore, the research includes a functional website implementation, providing a user-friendly interface that allows users to upload and test images—bridging the gap between academic research and practical deployment.

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

1 | INTRODUCTION

The evolution of artificial intelligence has transformed numerous fields, and image generation is no exception. Generating images through different models like Stable Diffusion has become quite straightforward, and these models are becoming more sophisticated each day, making it very challenging to distinguish between real images and those generated by AI. Different models are being developed to detect AI-generated images. Due to factors like authenticity, intellectual property rights, transparency, and ethical concerns, being able to detect AI-generated images is crucial for both artists and consumers.

Several techniques have been used in the past. Thakre et al. employed pixel-level feature extraction methods like Photo Response Non-Uniformity (PRNU) and Error Level Analysis (ELA) to detect anomalies in AI-generated images. Using CNNs trained on these features, they achieved over 95% accuracy, highlighting the challenges in differentiating realistic AI-generated visuals. This approach depends heavily on pixel-level inconsistencies, which can be computationally expensive and may struggle against newer generative models that minimize such artifacts [1]. Hossain et al. utilized CNNs and Vision Transformers, achieving 96.31% accuracy on the CIFAKE dataset. Grad-CAM provided insights into feature attribution, showcasing advancements in combating misinformation through image authenticity detection. Since this study was conducted solely on the CIFAKE dataset, its performance on other datasets and more advanced AI generation techniques remains uncertain [2]. Chinta et al. used the AI-ArtBench dataset to evaluate models like CNN, VGG-19, and ResNet-50, with CNN achieving the highest accuracy of 92.69%. Their research underscores the importance of detecting synthetic art in maintaining digital authenticity. Given the limited size and diversity of the AI-ArtBench dataset, the model's scalability and real-world applicability may not hold across a broader range of AI-generated images [3]. Bird and Lofti proposed a CNN model to classify AI-generated images from the CIFAKE dataset, achieving 92.98% accuracy. Using Grad-CAM, they identified background imperfections as key discriminators, emphasizing the necessity of synthetic image detection for data authenticity. Relying on background flaws may limit the model's success against more advanced image generators that produce seamless, realistic backgrounds [4].

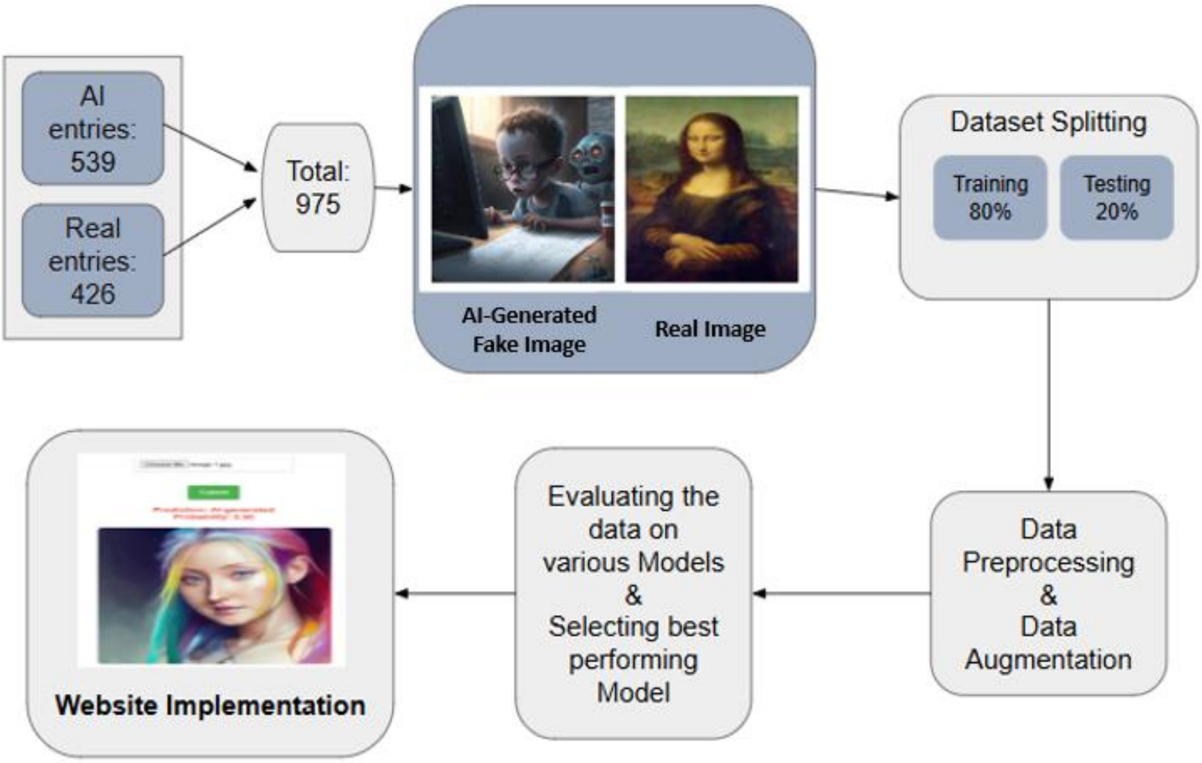


Fig. 1. Working Procedure

Purohit et al. trained CNN models on datasets of AI-generated and real images, achieving accuracies of 88% and 81%, respectively. This research addresses deepfake detection as a critical tool for countering cybersecurity risks and misinformation. The relatively modest accuracy rates suggest the need for further refinement, especially for handling newer deepfake techniques and more diverse datasets [5].
In this study, we aim to do an evaluation of various Convolutional Neural Network (CNN) architectures for image classification. Several popular CNN models were implemented, including ResNet50, VGG16, MobileNetV2, DenseNet121, InceptionV3, and EfficientNetB0, along with a custom CNN model. Each model was trained and tested on the same dataset to ensure consistency in comparison.

Previous works on this topic and related research gaps worked as a motivation to dive into this promising field of work. The key contributions of this work include-

(I) Improvement of result, i.e., accuracy of the pre-trained model through customization.

(II) A comparison between popular models used at present and how they interact with this specific dataset.

(III) A real-life implementation of the model for better interface and an opportunity for easy access to mass people.

The remainder of this paper is organized as follows. Section 2 describes the methodologies adopted for the experiments, including the data collection, splitting, and augmentation process, followed by a detailed overview of the CNN architectures and the proposed EfficientNetB0 model. Section 3 discusses the implementation of the best-performing model as a web interface using the Flask framework. Section 4 presents the experimental results and provides a comparative analysis of the different models' performance. Section 5 evaluates our model against previous research in AI-generated image detection and highlights its comparative advantages. Finally, Section 6 concludes the paper and outlines future directions for further research and enhancements.

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 two classes: AI-generated fake images and real images, with 10,800 images in each class, resulting in a total of 21,600 images [6]. The images were collected to represent diverse variations within each class, ensuring the robustness and generalizability of the trained model. For training and evaluation, the dataset is split into 80% for training and 20% for testing, maintaining class balance in both splits. This results in 8,640 images per class for training and 2,160 images per class for testing. The large volume of images in each class allows the model to learn intricate features effectively while the test set provides a reliable measure of its generalization performance. AI-generated fake and real images are shown in Figure 2.

2.2 | Data Augmentation

Data augmentation was done to increase the size and diversity of the training dataset so it can perform better to the new and unseen data. Several augmentation techniques like rotation, width shift, height shift, shear, zoom, and horizontal flip were applied. The augmentation was only done to the training dataset and the testing dataset wasn't modified. To achieve consistent performance, the images were resized to 150x150 in size.

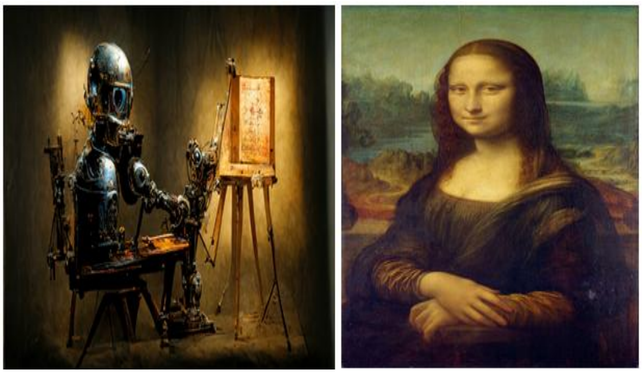


Fig. 2. AI- Generated Fake and Real Image

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 uses shortcut connections to solve the problem of vanishing gradients. It helps train very deep networks and is widely used for tasks like image classification and object detection.

VGG16 is a 16-layer neural network known for its simple design, using small 3x3 filters. It is good at learning features but needs a lot of computation and is commonly used for transfer learning in image tasks.

MobileNetV2 is a lightweight network designed for mobile devices. It uses special convolutions to work faster while using less power and is great for real-time image processing.

DenseNet121 connects each layer to all previous layers, making it very efficient at reusing features. It achieves high accuracy with fewer parameters and is used in tasks like medical imaging.

InceptionV3 uses modules that process images at different scales, capturing more features while keeping the network efficient. It works well for image recognition and reduces computation needs.

EfficientNetB0 balances depth, width, and resolution to create a network that is both fast and accurate. It is often used for tasks requiring good performance with limited resources. A custom CNN is built for a specific task, allowing flexibility in the design. It helps create efficient solutions for unique problems like object detection or image segmentation.

Based on the described models, EfficientNetB0 was proposed as the model for the task due to its balanced approach to depth, width, and resolution. An optimal combination of speed and accuracy is provided by this model, making it ideal for tasks requiring high performance with limited resources. Unlike other models, EfficientNetB0 is recognized for its high efficiency, with superior classification accuracy achieved while maintaining relatively low computational demands. Its capability to capture complex features and process images effectively is considered well-suited for distinguishing between real and AI-generated fake images in this study.

2.4 | Website Implementation

Various Python language-based frameworks have been developed to deploy machine learning and deep learning models for years. Frameworks like Fast API, and Flask have made it really convenient to work with the trained model with a better interface and easy access even for beginners. In this work, flask was selected due to its lightweight nature and flexible work environment. Having created the interface to take input images from the user, then the image was run by the loaded model EfficientNetB0 in this case. The output was two probabilistic fraction numbers representing the probability of the image belonging to either of the classes. These numbers were compared and judged and sent as JSON data to the server and finally, the result was displayed on the same page of the website. In the figure 3 that shows the website working procedure.

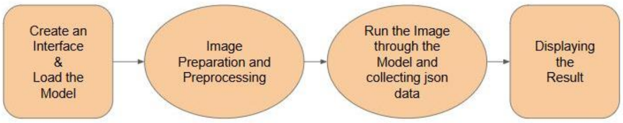

Create an Interface & Load the Model
Image Preparation and Preprocessing
Run the Image through the Model and collecting json data
Displaying the Result

Fig. 3. Website Work Procedure

3 | EXPERIMENTAL RESULTS

3.1 | Accuracy Curve Analysis

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.

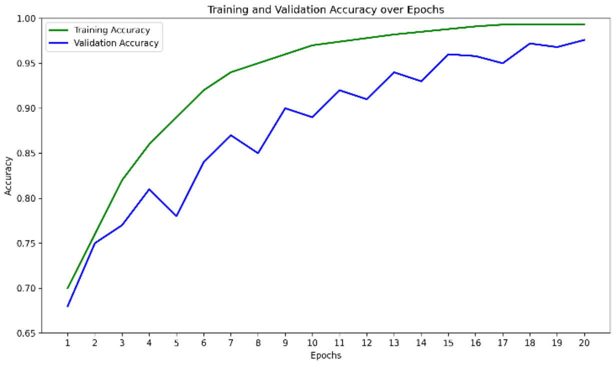
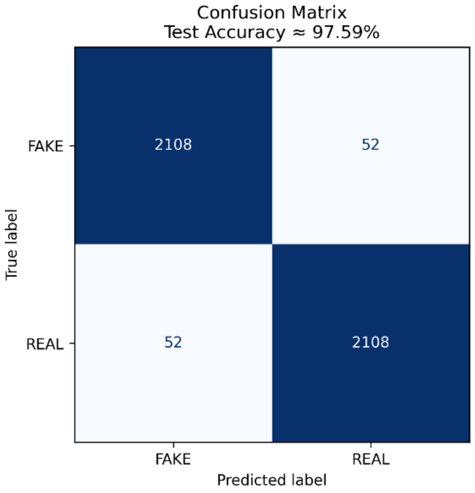


Fig. 4. Accuracy Curve

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 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.



True label \ Predicted label	FAKE	REAL
FAKE	2108	52
REAL	52	2108

Fig. 5. Confusion Matrix

3.3 | Classification Report Analysis

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.

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

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

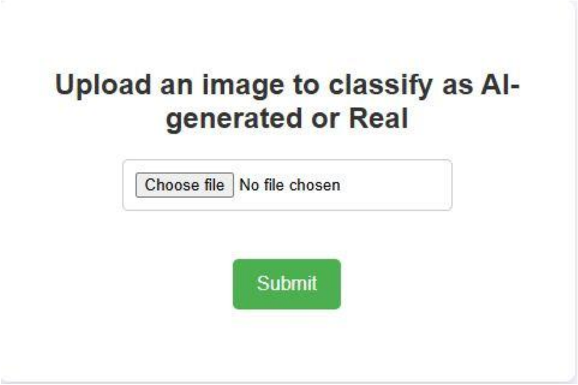
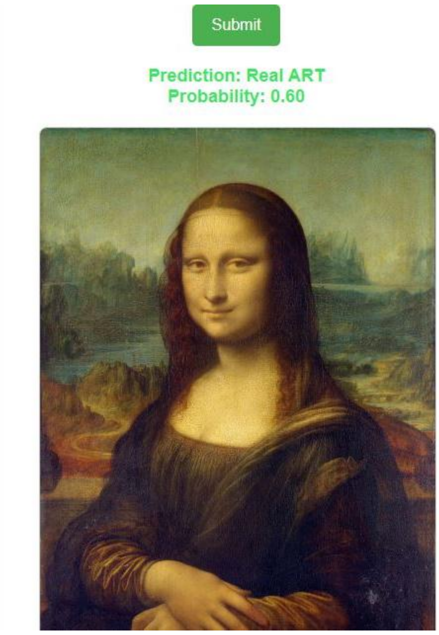

Upload an image to classify as AI-generated or Real

Fig. 6. Website User Interface

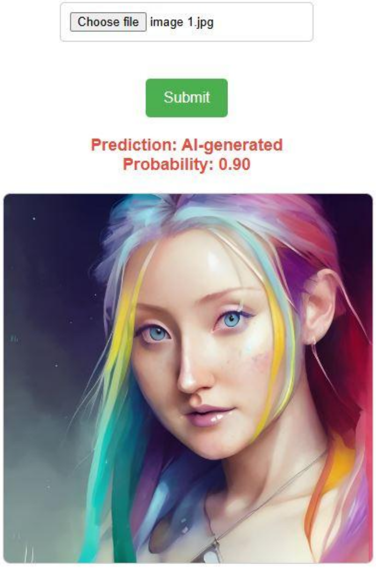
Figure 6 illustrates the user interface design of a locally hosted website that facilitates interaction with the system. Through this graphical interface, users can upload an image, which is then displayed alongside a prediction indicating whether it is AI-generated fake or a real image.

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.



Prediction: Real ART
Probability: 0.60

Fig. 7. Identified Real Image



Prediction: AI-generated
Probability: 0.90

Fig. 8. Identified AI-Generated Fake Image

4 | COMPARATIVE ANALYSIS

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
EfficietNetB0	**97.59**

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.

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.

4 | 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.

5 | CONCLUSION

This research demonstrated the effectiveness of the EfficientNetB0 model in distinguishing AI-generated fake image from real image, achieving an accuracy of **97.59%**, the highest among the models tested. While other works have achieved higher accuracy scores, our research bridges the gap between theoretical research and real-world implementation by integrating the model into a functional website. This practical application ensures accessibility for a broader audience, providing an effective tool for artists and consumers to verify the authenticity of images.

Despite its strong performance, there are areas for improvement. The dataset size was relatively small, which may limit the model's generalization ability. Additionally, the real-world performance of the model indicates promising potential, though further testing with larger, more diverse datasets would strengthen its reliability.

Future research could explore the use of ensemble learning to combine the strengths of multiple models or investigate the application of vision transformers for this task. Enhancements in the website's user interface and extending its capabilities, such as explaining predictions through visualization, would further elevate the practical utility of this work.

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).