Intelligent Methods in Engineering Sciences 4(3): 114-128, 2025

IMIENS

INTELLIGENT METHODS
IN ENGINEERING SCIENCES

https://www.imiens.org

International
Open Access

December, 2025

e-ISSN 2979-9236

Research Article

https://doi.org/10.58190/imiens.2025.158

Deep Learning Based Classification of Real and Synthetic Animal Images

Ibrahim AKBAS [a], ***Oya KILCI*** [b], ***Murat KOKLU*** [c,*]

[a] *Graduate School of Natural and Applied Sciences, Department of Computer Engineering, Selcuk University, Konya, Türkiye*
[b] *Graduate School of Natural and Applied Sciences, Department of Mechatronics Engineering, Selcuk University, Konya, Türkiye*
[c] *Technology Faculty, Department of Computer Engineering, Selcuk University, Konya, Türkiye*

ARTICLE INFO

Article history:
Received 18 November 2025
Accepted 29 December 2025

Keywords:
Stable Diffusion Turbo,
Real - Synthetic Image Classification,
MobileNetV2,
DenseNet121,
DenseNet169,
DenseNet201,
NASNetMobile

ABSTRACT

This study aims to develop a convolutional neural network based classification framework that can distinguish between synthetic and real animal images generated by the Stable Diffusion Turbo model. Additionally, this study will evaluate the performance of different network architectures for this task. The study employed a balanced dataset of 31,995 images, including 16,000 real images and 15,995 synthetic images generated by Stable Diffusion Turbo. The dataset includes eight animal categories: dogs, cats, cows, rabbits, horses, sheep, chickens, and elephants. All images were resized to 224 by 224 pixels, and standard preprocessing techniques were applied. During the classification stage, five pretrained convolutional neural network architectures were retrained using transfer learning, including MobileNetV2, DenseNet121, DenseNet169, DenseNet201, and NASNetMobile. Model performance was evaluated using accuracy, precision, recall, the F1 score, the area under the curve of the receiver operating characteristic, confusion matrices, and training time. The experimental results demonstrate that MobileNetV2 and DenseNet201 achieved the highest classification performance, with respective accuracy rates of 99.58% and 99.56%, and perfect area under the curve values. All DenseNet variants exhibited complete sensitivity in detecting synthetic images, whereas NASNetMobile showed substantially lower performance compared to the other models. These results suggest that synthetic images produced by diffusion-based generative models can be reliably identified when appropriately designed CNN architectures and balanced datasets are used. This provides a significant methodological contribution to the discrimination of synthetic versus real images, the detection of fake content, and the verification of visual authenticity.

1. INTRODUCTION

In recent years, advancements in technology have led to significant improvements in the efficiency and speed of visual content production. This complicates the discernment of the distinction between authentic photographs and those produced by artificial intelligence. This is particularly salient in the context of deep generative models, which have the capacity to generate visual content that exhibits a high degree of realism. In 2014, Goodfellow et al. developed Generative Adversarial Networks (GANs), which represented a significant milestone in the field of deep learning-based image production. GANs employ an adversarial learning paradigm, whereby two networks are trained concurrently. One network generates synthetic images, while the other network is tasked with differentiating between authentic and fabricated images. This adversarial training has contributed to the remarkable success and immediate popularity of GANs in the domain of computer vision [1]. In recent years, there has been a proliferation of variants of GANs developed with the objective of producing high-quality and photorealistic images [2].

The utilization of artificial image production can prove to be highly advantageous in a multitude of scenarios. However, it is imperative to acknowledge the potential risks associated with this practice, particularly in contexts pertaining to security, reliability, and ethical considerations. In the medical field, for instance, Generative Adversarial Networks (GANs) are employed to generate synthetic images, thereby enhancing the efficacy of Convolutional Neural Network (CNN) models. This approach is instrumental in addressing the challenge of insufficient data [3, 4]. In a similar vein, artificial image generation has been extensively utilized across a range of disciplines, including agriculture [5, 6], cybersecurity [7],

* **Corresponding Author:** mkoklu@selcuk.edu.tr

vehicle and intelligent transportation systems [8]. However, diffusion-based generative models have recently emerged as a compelling alternative to GANs. This is due to the fact that they are easier to train, produce superior images, and exhibit greater diversity. Denoising Diffusion Probabilistic Models (DDPMs) established the foundational framework for the field by conceptualizing image production as a progressive denoising process. Subsequently, Latent Diffusion Models (LDMs) emerged as a more affordable solution by operating within lower-dimensional latent spaces. A significant number of individuals employ Stable Diffusion, an open-source LDM, due to its user-friendly interface and its ability to generate high-quality images from text-based prompts [9].

The present study examines the effectiveness of convolutional neural network (CNN)–based models in discriminating authentic images from synthetic images generated by Stable Diffusion Turbo (SD Turbo), a diffusion-based generative model designed to accelerate the sampling process of conventional diffusion frameworks. To this end, a balanced dataset consisting of 16,000 real images and 15,995 synthetic images generated by SD Turbo was constructed and utilized for the classification task. The experimental work utilizes transfer learning to enhance numerous pre-trained CNN architectures, including NASNetMobile, MobileNetV2, DenseNet121, DenseNet169, and DenseNet201. Each design's capacity to differentiate between authentic images and those generated by diffusion was meticulously evaluated.

In this context, with the increasing prevalence of diffusion-based production models, current approaches in the literature aimed at reliably distinguishing between synthetic and real images have been examined.

Lokner Lađević et al. (2024), stressed that figuring out how to tell the difference between AI-generated photographs and actual ones has been an important research challenge in the last several years. They said that existing ways of finding false images, which mostly depend on hand-crafted features, are becoming less and less useful as GAN-based picture-generating algorithms evolve more complicated. In this situation, the authors suggested a simple CNN design with eight convolutional layers and two fully connected layers. The suggested model was thoroughly tested on two conventional benchmark datasets and a new dataset made from Sentinel-2 satellite photos. The suggested architecture attained an accuracy of 97.32% on the CIFAKE dataset, surpassing or matching the performance of four leading methodologies [10].

Y. Patel et al. (2023) did a study that showed many of the current deepfake spotting algorithms are not good at finding differences between frames in movies or pictures. The writers showed a better deep convolutional neural network (D-CNN) architecture to get around this problem. The trial test used a total of 15,000 photos, with 10,000 real photos and 5,000 deepfake photos from seven different datasets under the Reconstruction Challenge framework. It was found that the suggested D-CNN worked very well with a number of GAN-based creation methods. StarGAN got it right 99.17% of the time, AttGAN got it right 95.33% of the time, GDWCT got it right 94.67% of the time, and StyleGAN got it right 94.67% of the time. It turns out that the suggested way not only finds things well, but it also works well with many different GAN sources [11].

Bird and Lotfi (2024) created a new real–synthetic dataset called CIFAKE by using a latent diffusion method to make synthetic pictures that look like the CIFAR-10 structure. A total of 36 different CNN designs were tested on this dataset. The model that did the best got an accuracy rate of 92.98%. Grad-CAM analyses showed that the decisions about how to classify things were mostly based on small visual differences in the background, not on what the items meant. The study gives researchers working on synthetic picture detection a clear and easy-to-understand standard to use [12].

Mallet et al. (2023) emphasized the risks of misinformation stemming from the widespread dissemination of deepfake content on social media and devised a hybrid CNN–SVM approach for identifying deepfake photos. The study employed the publicly available 140k Real and Fake Faces dataset, and the proposed model achieved a classification accuracy of 88.33% in identifying deepfake images. The findings indicate that hybrid machine learning techniques offer a viable and effective approach for deepfake detection tasks. [13].

Raza et al. (2022) suggested that deepfake content poses a significant hacking threat, potentially resulting in identity theft, extortion, and misinformation. They proposed an innovative method for detecting deepfakes with deep learning techniques. They utilized a dataset comprising authentic and fabricated facial photos to develop a Deepfake Predictor (DFP) model that integrates VGG16 with a CNN architecture. Advanced transfer learning models such as Xception, NASNet, and MobileNet were employed to evaluate the proposed DFP. It outperformed these models, with an accuracy of 94% and a precision of 95%. The findings indicate that the proposed method is effective for detecting deepfakes in digital forensic applications [14].

Abir et al. (2023), talked about how deepfake content is having a bigger effect on society and looked at how well and how easy it is to find deepfake content. Four CNN designs were looked at in the study: InceptionV3, DenseNet201, ResNet152V2, and InceptionResNetV2. It used a large dataset from Kaggle that included 70,000 real pictures (Flickr–NVIDIA) and 70,000 256x256-pixel fake photos made by StyleGAN. The three models all did a

great job of putting facts in order. With a rate of 99.81%, DenseNet201 was the most correct. With a score of 99.68%, InceptionV3 came in second. Third place went to ResNet152V2, which got it right 99.19% of the time. There was one model that was right 99.87% of the time, and that was InceptionResNetV2. LIME-based explainability analysis showed that it could make smart decisions. The results show that combining deep learning and explainable AI (XAI) might be the best way to find deepfake pictures that can be trusted [15].

Chen et al. (2025) treat diffusion models as a probabilistic framework based on forward noise reduction and reverse denoising processes in image production. They systematically compare three major formulations from the literature: DDPM, score-based models, and the continuous-time approach based on SDEs that combines them. The authors also discuss guidance strategies for conditional production and acceleration methods to improve sampling efficiency and stability. Additionally, they address the socio-ethical impacts of diffusion-based production, including privacy, copyright, bias, and abuse [16].

Rahat et al. (2025) propose Automated Generative Augmentation (AGA), a framework that reduces data scarcity in fine-grained image classification. AGA produces richer training data by segmenting and preserving the subject (foreground), while increasing background diversity through large language model (LLM)-based prompt diversification and diffusion-based synthesis. It also applies controlled manipulation to the subject using affine transformations. Evaluations on ImageNet, CUB, and iWildCam showed that AGA improved in-distribution accuracy by 15.6%, out-of-distribution accuracy by 23.5%, and SIC score by 64.3% compared to basic approaches. Grad-CAM analyses showed that the model's attention was more consistently directed to the target subject regions. However, subject-background context mismatches led to visual inconsistencies in some examples, revealing room for improvement in terms of contextual suitability [17].

Siddiqui et al. (2025) propose a hybrid detection approach combining DenseNet121-based local feature extraction and global contextual modeling capabilities via Cross-ViT. This approach addresses the growing security risks posed by increasingly persuasive face-focused deepfakes. The study aims to achieve competitive performance without complex strategies like distillation or ensemble, utilizing a voting-based inference mechanism to improve decision stability when multiple faces are present in a single frame. Experiments showed that the method achieved an area under the curve (AUC) of 99.99% and an average F1 score of 99.0% in DeepForensics 1.0 and an AUC of 97.4% and an F1 score of 95.1% in CelebDF. However, generalization to unknown fake samples is noted as an area that needs improvement in the future with richer feature representation [18].

Banerjee (2025) proposes a neural architecture search (NAS)-based deep learning model to improve the accuracy of deepfake detection. The model enhances the quality of the input by performing face and related region segmentation with YOLOv8 during the preprocessing phase. It also expands the diversity of the sample through data augmentation. For the study, frames were extracted from 100 real and 100 deepfake videos selected from the CelebDF v2 dataset. Then, a training set consisting of 2,000 real and 2,000 deepfake images was created after augmentation. The proposed approach achieved 99.04% accuracy in testing. With high precision, recall, and F1 values, the model demonstrated its ability to reduce false positives while effectively detecting real deepfake samples. The comparative analysis reported that the scenario using YOLOv8-based face extraction and augmentation yielded significantly better results than the scenario without segmentation or augmentation. This conclusion suggests that these two components significantly strengthen deepfake classification [19].

A comprehensive assessment of these studies indicates that, although existing methods yield strong performance, they still exhibit notable limitations when confronted with diverse data sources, emerging generative models, and large-scale real–synthetic image distributions. Consequently, a systematic comparison of models from different architectural families such as DenseNet, MobileNet, and NASNet for the classification of real and AI-generated images is essential, both to better understand performance variations and to develop more reliable and lightweight detection systems suitable for practical applications. By addressing this gap in the literature, the present study aims to provide a thorough evaluation of modern CNN architectures on a large-scale dataset and to offer a more comprehensive, up-to-date, and practically applicable framework for synthetic image detection.

The remainder of this paper is organized as follows. As delineated in Section 2, the characteristics of the dataset are described, along with the CNN architectures employed and the data preprocessing and enhancement procedures. The third section of the text presents the experimental results and discusses them through analysis. The final section of the study presents the summary of the results and potential future research directions.

2. MATERIAL AND METHODS

In this section, the dataset forming the experimental design of the study, the preprocessing steps, the structural components of the deep learning models, the training protocol, and the performance evaluation metrics are presented in detail. Figure 1 presents the workflow diagram of the proposed methodology.

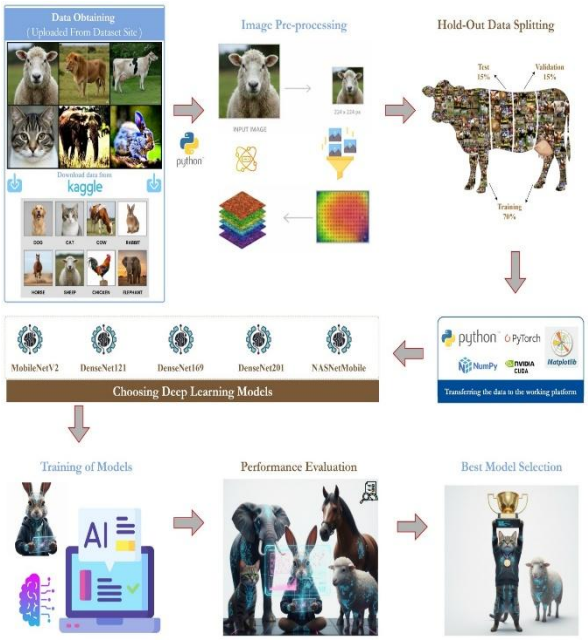

Figure 1. Workflow Diagram

2.1. Dataset and Preprocessing

The dataset used in this study consists of a total of 31,995 images and comprises two primary classes: real and artificially generated (AI) images. The real image class includes 16,000 natural animal photographs, while the artificial image class consists of 15,995 synthetic animal images generated using the Stable Diffusion Turbo (SD-Turbo) model. The animal images in the dataset cover eight distinct categories: dog, cat, cow, rabbit, horse, sheep, chicken, and elephant. The number of images in both classes was kept balanced to prevent the model from developing bias due to class imbalance. During the preprocessing stage, all images were resized to a resolution of 224×224 pixels to ensure compatibility with the selected model architectures, and pixel values were normalized accordingly.

Table 1. Class distribution of the dataset

Class	Training Set	Validation Set	Test Set	Total
AI	11196	2399	2400	15995
Real	11200	2400	2400	16000
Total	22396	4799	4800	31995

Table 1 presents the quantitative distribution of the dataset used in this study across the training, validation, and test sets for the artificial (AI) and real (Real) classes. The dataset consists of a total of 31,995 images, with the number of samples kept balanced between the two classes. Specifically, the AI class contains 15,995 images, while the Real class includes 16,000 images. The training set comprises 22,396 images, whereas the validation and test sets contain 4,799 and 4,800 images, respectively. This balanced distribution ensures that the model is trained and evaluated without developing class-related bias.

Figure 2. Representative examples of synthetic images belonging to the AI class in the dataset

Figure 2 shows representative examples selected from the AI-generated image dataset. The images illustrate the synthetic data distribution and the intra-class diversity used during model training.

Figure 3. Representative examples of real images belonging to the Real class in the dataset

Figure 3 shows representative examples selected from the real image dataset. These images reflect the natural data distribution and illustrate the intra-class diversity used during model training.

2.2. Experimental Setup and Training Parameters

In this section, the experimental environment in which the deep learning–based classification studies were conducted is described in detail. In addition to the hardware and software infrastructure used during model training and evaluation, the core libraries employed and the hyperparameter configurations of the network architectures are systematically presented.

Table 2. Hardware and software components used in the experimental study environment

Component	Description
Processor (CPU)	Intel Xeon processor with 12 cores
Memory (RAM)	167 GB
Graphics Processing Unit (GPU)	NVIDIA A100-SXM4-80GB
Software Frameworks	CUDA 12.5.82, PyTorch 2.8.0

Table 2 summarizes the hardware and software components used in the experimental studies. The training processes were conducted on a high-capacity system

equipped with a 12-core Intel Xeon processor and 167 GB of RAM. The NVIDIA A100-SXM4-80GB GPU, with its high parallel processing capability and large memory capacity, significantly reduced the computational cost of the deep learning models. The software environment was built on CUDA 12.5.82 and PyTorch 2.8.0, enabling efficient and stable training through GPU-accelerated computations.

Table 3. Library information in the experimental study environment

Library	Version
NumPy	2.0.2
Pandas	2.2.2
Scikit-learn	1.6.1
Matplotlib	3.10.0

Table 3 lists the experimental study's primary Python libraries and their versions. NumPy (v2.0.2) and Pandas (v2.2.2) offer basic data and math capabilities. Scikit-learn (v1.6.1) helped us calculate model evaluation measures and other machine learning tasks. We plotted training-related losses, accuracy, and performance indicators using Matplotlib (v3.10.0). Using up-to-date libraries helps keep the experimental setup stable and reproducible.

Table 4. Hyperparameter Configuration of the Employed Models

Hyperparameter	Value	Description
Batch Size	32	It represents the number of samples simultaneously fed into the model at each training iteration [20, 21]. This hyperparameter plays a critical role in determining memory requirements while also contributing to the stability of the optimization process [22].
Epochs Frozen	15	It refers to the number of epochs specified for the initial phase of transfer learning during which the pre-trained layers are frozen. This stage enables training exclusively of the newly added classification layers.
Epochs Ft	10	It denotes the number of additional training epochs performed after unfreezing the upper layers of the model during the fine-tuning stage. This process contributes to better adaptation of the model to the target dataset.
Unfreeze Last	30	It specifies the number of final layers that are unfrozen (made trainable) during the fine-tuning stage. This hyperparameter determines how much of the pre-trained network is re-optimized.
Learning Rate Frozen	0.0001	It refers to the learning rate used during the initial training phase with frozen layers. This value is selected to enable rapid yet stable adaptation of the classification layers.
Learning Rate Ft	0.00001	It denotes the low learning rate applied during the fine-tuning stage, allowing for delicate parameter updates without disrupting the model's previously learned representations.
Patıence Early Stopping	10	It represents the patience threshold of the early stopping mechanism used to terminate training when no improvement is observed in the validation loss [23]. This strategy is employed to prevent overfitting.
Patience Reduce LR	5	It specifies the number of epochs with no improvement in validation loss required to trigger the learning rate reduction mechanism. This parameter supports the optimization process when training reaches a plateau.
Random Seed	42	It represents the deterministic initialization value that ensures the reproducibility of data splitting, weight initialization, and other stochastic processes [24, 25]. This parameter is critical for maintaining experimental consistency.
Image Size	224	It defines the input image size of the model. Prior to training, all images were resized to a resolution of 224×224 pixels to ensure compatibility with the standard input format required by the network architectures.

Table 4 lists the basic hyperparameter parameters used during model training. The batch size was 32, and transfer learning was two-stage. First, the pre-trained layers were frozen and the models trained for 15 epochs. The final 30 layers were unfrozen and fine-tuned for 10 epochs in the second step. Using 1×10^{-5} learning rates for each stage allowed for more controlled optimization. Early stopping (10 patience) and learning rate lowering (5 patience) reduced overfitting risk. All trials were replicated using a random seed value of 42 and a fixed model input size of 224x224 pixels, as per CNN architecture specifications.

2.3. Deep Learning Architectures

Pre-trained CNN-based deep learning architectures were used to classify real photographs from fake ones in this study. Both lightweight designs for mobile and resource-limited contexts (MobileNetV2 and NASNetMobile) and more complicated architectures with densely connected frameworks (DenseNet121, DenseNet169, and DenseNet201) were retrained using transfer learning. The classification architectures' core structural and operational ideas are briefly covered in this section.

2.3.1. MobileNetV2

MobileNetV2 preserves the use of Depthwise Separable Convolutions (DSC) introduced in MobileNetV1, reducing the number of parameters and computational cost to approximately 18% compared to standard convolution operations [26, 27]. The model employs Linear Bottleneck layers and Inverted Residual blocks to mitigate the

information loss caused by ReLU activations in low-dimensional feature spaces [28, 29]. In these blocks, the input is first expanded into a higher-dimensional space, then filtered through depthwise convolution, and subsequently projected back to a low-dimensional representation via a linear convolution. [26, 30, 31].

2.3.2. DenseNet121

Huang et al. presented DenseNet, an architecture in which each layer in a dense block gets the feature maps from all the layers before it as input [29]. Thus, the ℓ-th layer operates on the concatenated feature maps from all earlier layers and processes them through a BN–ReLU–Conv sequence [32, 33]. There are four dense blocks in the design, each separated by transition layers that do 1×1 convolutions and 2×2 average pooling operations. After that, there is a fully connected layer with a softmax activation that gives the class probabilities [33-36].

2.3.3. DenseNet169

DenseNet-169 starts with a convolutional and pooling layer, then has four dense blocks, each separated by a transition layer. It ends with a classification layer that has a softmax activation function [37]. Batch normalization (BN), Rectified Linear Unit (ReLU) activation, and convolution processes are all part of each convolutional layer [38]. In each dense block, there are 1×1 convolutional layers followed by 3×3 convolutional layers. In DenseNet-169, there are four dense blocks, each with 6, 12, 32, and 32 pairs of 1×1–3×3 convolutional layers. This makes a total of 82 convolutional pairings [39].

2.3.4. DenseNet201

DenseNet201 facilitates feature map reutilization by transmitting the output of each layer to all future layers in a feed-forward fashion, thus diminishing the parameter number and overall model complexity while enhancing computational efficiency [40]. The architecture consists of Dense Block structures where the spatial dimensions of feature maps are invariant inside each block, and Transition Layers between blocks that execute downsampling via batch normalization, 1×1 convolution, and 2×2 pooling operations [41].

2.3.5. NASNetMobile

NASNetMobile is designed to be computationally efficient for mobile applications and autonomously generates a CNN architecture optimized for mobile devices using a neural architecture search (NAS) algorithm [42, 43]. The architecture consists of multiple convolutional layers with varying filter sizes, max-pooling layers that downsample feature maps, and skip connections that enhance information flow and mitigate the vanishing gradient problem. The core of NASNetMobile is formed by reusable and composable subnetworks known as Cells. These cells are categorized into two types: Normal Cells and Reduction Cells, and the network is constructed by stacking these cells sequentially [44]. This modular design provides flexibility to adapt to different task requirements and hardware constraints. Moreover, as NASNetMobile is pre-trained on large-scale datasets such as ImageNet, it is capable of effectively learning complex visual features [45].

2.4. Performance Evaluation Metrics

In this subsection, the primary evaluation metrics used to objectively compare the classification performance of the proposed models are defined. This approach enables a fair comparison of the algorithms without prioritizing any specific application domain [46].

Table 5. Performance Metrics

Metric	Formula	Description
Precision	$\frac{tp}{tp+fp}x100\%$	It is defined as the ratio of correctly classified positive samples to the total number of samples predicted as positive [47, 48]. This metric is also referred to as the Positive Predictive Value.
Accuracy	$\frac{(tp+tn)}{(tp+tn+fp+fn)}x100\%$	Accuracy is defined as the ratio of correctly classified samples to the total number of sample [47], reflecting the overall classification performance of the model [49, 50].
Recall / Sensivity	$\frac{tp}{tp+fn}x100\%$	It represents the true positive rate, indicating the proportion of correctly identified positive samples among all actual positive samples [51].
F1-Score	$\frac{2*Precision*Sensivity}{Precision+Sensivity}x100\%$	The F-measure is an evaluation metric defined as the harmonic mean of Precision and Recall [52].
Macro avg	$\frac{1}{N}\sum_{i=1}^{N}\frac{2*Precision_i*Recall_i}{Precision_i+Recall_i}$	This metric computes the F1-score independently for each label and then takes the arithmetic mean of these scores, assigning equal weight to all classes [53].
Weighted avg	$\sum_{i=1}^{N} w_i \frac{2*Precision_i*Recall_i}{Precision_i+Recall_i}$ $w_i=\frac{tp_i+fn_i}{\sum_{j=1}^{N}(tp_j+fn_j)}$	Similar to the macro F1-score, this metric computes the F1 value separately for each class; however, when averaging these values, it assigns weights based on the number of true samples (support) associated with each class. Here, $w_i$denotes the proportion of true samples belonging to class $i$relative to the total number of samples [53]. The weighted F1-score incorporates the support (weight) of each class through w_i, where $tp_i$and fn_irepresent the true positive and false negative samples of class i, respectively, and the denominator corresponds to the sum of true positive and false negative samples across all classes [53].

Table 5 summarizes the primary performance metrics used to evaluate the classification performance of the models, along with their mathematical formulations. Precision, Recall, Accuracy, and F1-score are derived from the components of the confusion matrix (TP, TN, FP, FN), enabling a comprehensive assessment of the model's ability to correctly identify positive classes, its overall accuracy, and its classification performance even in the presence of class imbalance.

Table 6. Confusion matrix representation for binary classes [54, 55].

		Predicted	
		Positive C^+	Negative C^-
Actual	Positive C^+	True positive **(tp)** *The number of positive predicted positive values.*	False negative **(fn)** *Number of false negative estimates.*
	Negative C^-	False positive **(fp)** *The number of false positive estimates.*	True negative **(tn)** *The number of correctly predicted negative values.*

Table 6 presents the structure and components of the confusion matrix used to evaluate model performance in binary classification problems. The matrix consists of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values, from which classification performance metrics are derived. With true class labels on one axis and predicted class labels on the other, the confusion matrix enables a detailed analysis of the types of errors made by the model in a binary classification setting [56].

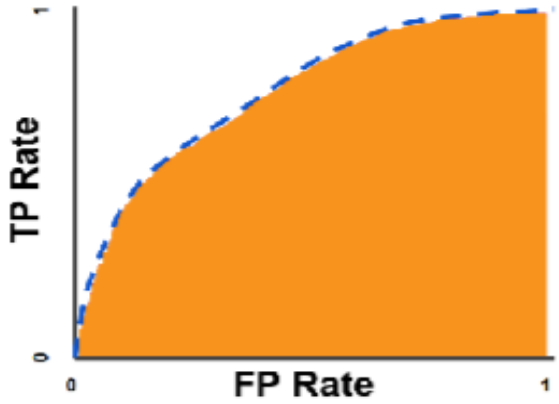

Figure 4. ROC curve (blue dotted line) and the AUC area (orange region) [57].

The ROC curve illustrates the variation of the model's sensitivity with respect to specificity across different decision thresholds. In this representation, sensitivity is plotted along the y-axis, while specificity is shown on the x-axis, with each point corresponding to the performance pair obtained at a particular threshold. The ROC curve is formed by connecting these points. The area under the ROC curve, referred to as AUC, provides a quantitative measure of the model's discriminative capability. As the AUC value approaches 1, the model's ability to distinguish between classes increases, whereas values approaching 0 indicate performance close to random classification. Figure 4 presents the ROC curve of the model along with the corresponding AUC area [57].

3. RESULTS AND DISCUSSION

In this section, the experimental results of the proposed deep learning–based classification approach are presented. An evaluation of the MobileNetV2, DenseNet121, DenseNet169, DenseNet201, and NASNetMobile architectures is conducted, and a comparison of the models is made in terms of accuracy, sensitivity, specificity, F1-score, and AUC. Furthermore, computational efficiency is analyzed by considering the training time of each model. The findings reveal discernible disparities in performance among the architectures, with certain models attaining a favorable equilibrium between high classification accuracy and low computational expense.

Table 7. Results of Performance Evaluation

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	ROC / AUC	Training Time
MobileNetV2	99.58	99.30	99.87	99.58	1.0000	19.7 min
DenseNet121	97.85	95.88	100.00	97.85	1.0000	24.4 min
DenseNet169	99.12	98.28	100.00	99.12	1.0000	27.0 min
DenseNet201	99.56	99.13	100.00	99.56	1.0000	29.5 min
NASNet Mobile	92.46	87.71	98.75	92.43	0.9935	28.5 min

Table 7 presents a comprehensive comparison of the classification performance of the five deep learning architectures evaluated in this study. The findings indicate substantial disparities in the performance metrics, including accuracy, precision, recall, F1-score, ROC/AUC, and training time, among the models under consideration. MobileNetV2 demonstrated the highest overall performance, with an accuracy of 99.58%, indicating an effective balance between computational efficiency and classification accuracy despite its lightweight architecture, as evidenced by its high precision (99.30%) and recall (99.87%). Among the DenseNet variants, DenseNet169 and DenseNet201 demonstrated robust performance, attaining high accuracy levels of 99.12% and 99.56%, respectively, along with perfect recall 100%. DenseNet121, while requiring a shorter training time compared to the other DenseNet models, yielded relatively lower accuracy and precision. NASNetMobile demonstrated the weakest performance, with an accuracy of 92.46% and an F1-score of 92.43%. However, its relatively high recall value of 98.75% indicates a strong tendency to correctly identify positive samples.

The analysis revealed that all models demonstrated ROC/AUC values greater than 99%, suggesting a high discriminative capability between the classes. With regard to the duration of training, MobileNetV2 demonstrated the most efficient use of time, underscoring its efficacy in terms of computational efficiency.

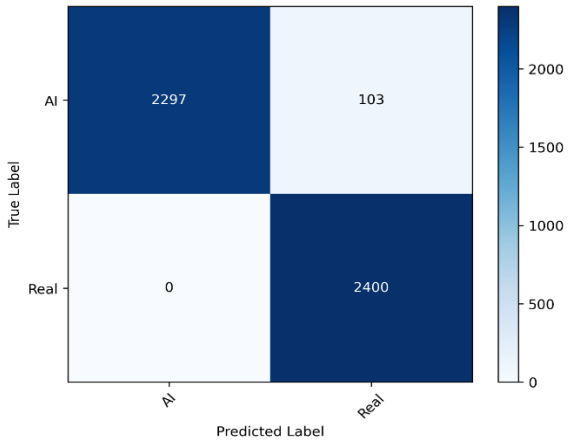

Figure 5. Confusion Matrix of the DenseNet121 model

Figure 5 shows the classification performance of the DenseNet121 model on the test dataset. The model accurately classified all 2400 samples in the Real category (2400/2400); however, it correctly identified only 2297 samples in the AI category, resulting in 103 misclassifications. The model performs very well in recognizing real images. However, it makes some errors when distinguishing synthetic images.

Table 8. Evaluation metrics of the DenseNet121 model

Class / Metric	Precision (%)	Recall (%)	F1-score (%)
AI	100.00	96.00	98.00
Real	96.00	100.00	98.00
Accuracy	-	-	98.00
Macro avg	98.00	98.00	98.00
Weighted avg	98.00	98.00	98.00

Table 8 shows the main classification success indicators for distinguishing between the Real and AI classes using the DenseNet121 model. The precision and recall scores are 100% and 96%, respectively, for the AI class, and 96% and 100%, respectively, for the Real class. Each class has an F1 score of 98%. Overall, the model is accurate 98% of the time, and both the macro-averaged and the weighted-averaged measures show the same result. This indicates that the model performs impressively across all classes.

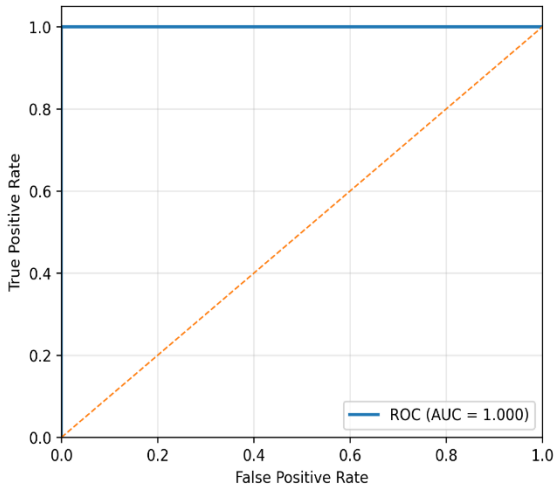

Figure 6. ROC curve and AUC value of the DenseNet121 model

Figure 6 shows that the DenseNet121 model can effectively distinguish between AI-generated and real samples in the test dataset. This is evident from its ROC AUC value of 1.0000.

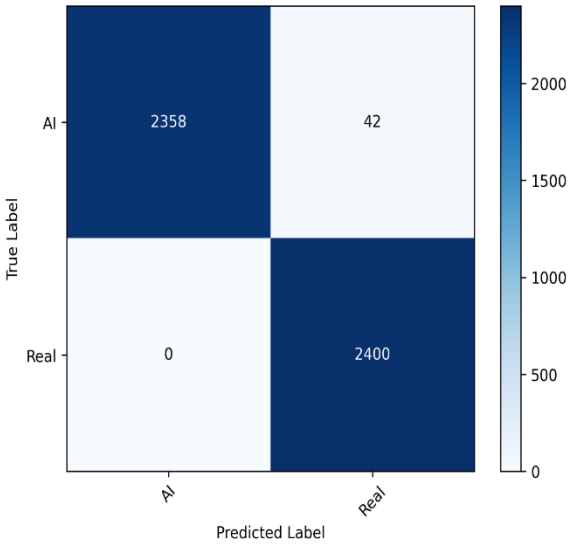

Figure 7. Confusion Matrix of the DenseNet169 model

As illustrated in Figure 7, the DenseNet169 model demonstrates a high degree of classification performance on the test dataset. The model exhibited an accuracy of 100% in its classification of all samples in the Real class, while generating 2.358 accurate predictions and 42 erroneous predictions for the AI class. The findings suggest that the model exhibits a high degree of accuracy in differentiating between authentic images and synthetic ones.

Table 9. Performance evaluation metrics of the DenseNet169

Class / Metric	Precision (%)	Recall (%)	F1-score (%)
AI	100.00	98.00	99.00
Real	98.00	100.00	99.00
Accuracy	-	-	99.00
Macro avg	99.00	99.00	99.00
Weighted avg	99.00	99.00	99.00

As illustrated in Table 9, the DenseNet169 model

demonstrates a high level of accuracy in its classification of data, with metrics indicating optimal performance for both the AI and Real classes. The precision and recall values for the AI class are 100% and 98%, respectively, while for the Real class, precision reaches 98% and recall reaches 100%. The F1 score was observed to be 99% for both classes. Both the macro-average and weighted-average metrics reached 99%, indicating a highly consistent and balanced performance between the two classes.

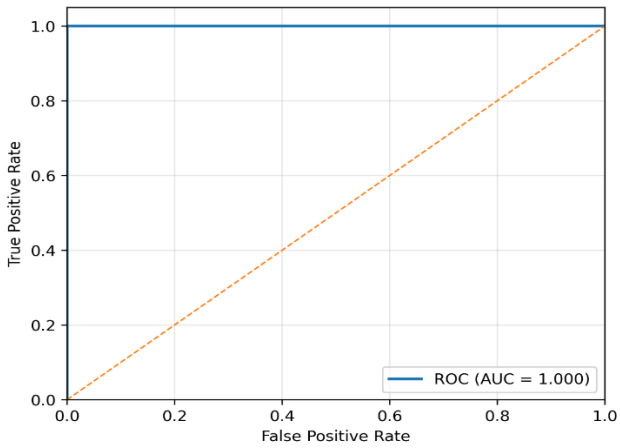

Figure 8. ROC curve and AUC value of the DenseNet169 model

As demonstrated in Figure 8, the ROC/AUC value of 1.0000 indicates that the DenseNet169 model exhibits an exceptionally high discriminatory capacity between artificial intelligence (AI)-generated and authentic samples in the test dataset.

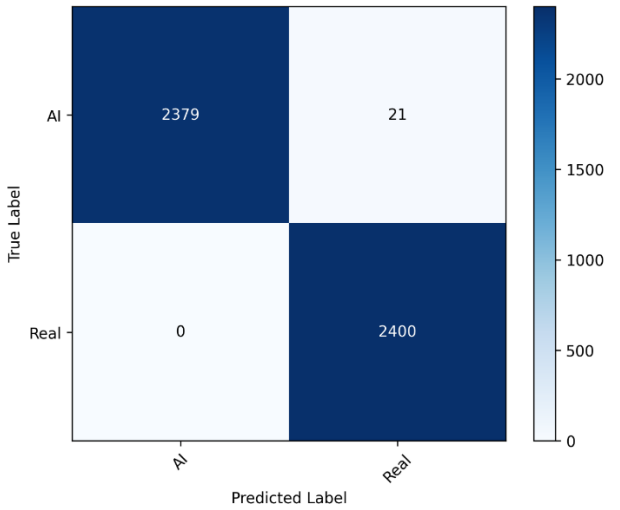

Figure 9. Confusion Matrix of the DenseNet201 model

As demonstrated in Figure 9, the DenseNet201 model yielded 2,379 accurate predictions and 21 erroneous ones for the AI class on the test dataset. These findings suggest that the model attains high accuracy in distinguishing real images and exhibits strong selectivity in detecting synthetic images.

Table 10. Performance evaluation metrics of the DenseNet201

Class / Metric	Precision (%)	Recall (%)	F1-score (%)
AI	100.00	99.00	100.00
Real	99.00	100.00	100.00
Accuracy	-	-	100.00
Macro avg	100.00	100.00	100.00
Weighted avg	100.00	100.00	100.00

Table 10 presents the classification performance metrics of the DenseNet201 model. For the AI class, the precision and recall values are 100% and 99%, respectively, while for the Real class, the precision is 99% and the recall is 100%. The F1-score is calculated as 100% for both classes. The model demonstrated an overall accuracy of 100%, with both macro-averaged and weighted-averaged metrics also exhibiting values of 100%.

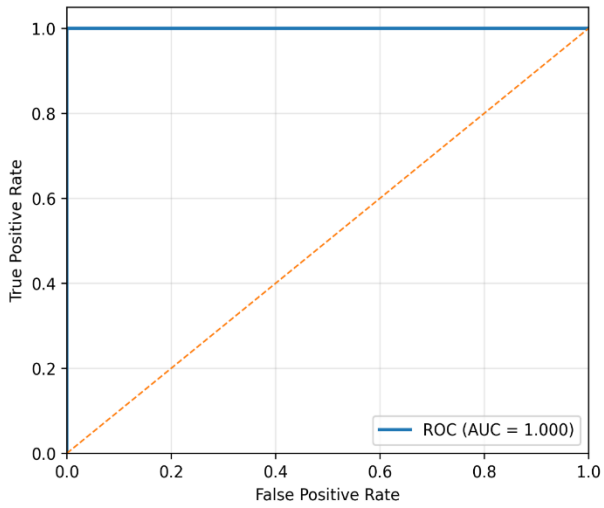

Figure 10. ROC curve and AUC value of the DenseNet201 model

As illustrated in Figure 10, the ROC curve of the DenseNet201 model on the test dataset demonstrates the model's performance. The area under the curve is calculated as 1.000 based on the relationship between the true positive rate and the false positive rate.

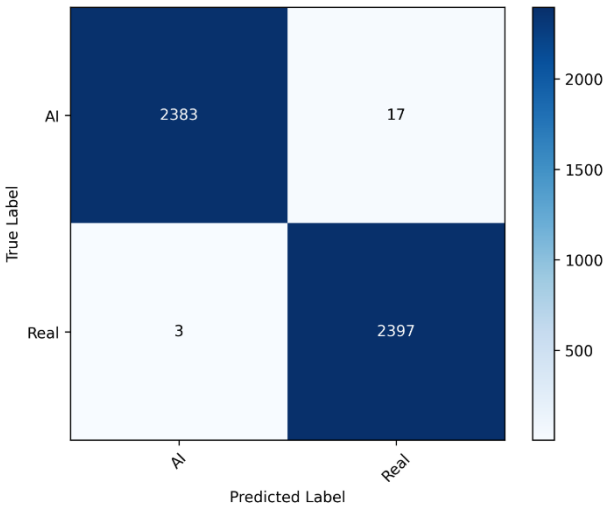

Figure 11. Confusion Matrix of the MobileNetV2 model

As illustrated in Figure 11, the MobileNetV2 model's

classification performance on the test dataset is evident. The model yielded 2,383 accurate predictions and 17 erroneous predictions for the AI class, and 2,397 accurate predictions and 3 erroneous predictions for the Real class.

Table 11. Performance evaluation metrics of the MobileNetV2

Class / Metric	Precision (%)	Recall (%)	F1-score (%)
AI	100.00	99.00	100.00
Real	99.00	100.00	100.00
Accuracy	-	-	100.00
Macro avg	100.00	100.00	100.00
Weighted avg	100.00	100.00	100.00

Table 11 presents the classification performance metrics of artificial intelligence (AI) and real images using the MobileNetV2 model. For both classes, sensitivity, recall, and F1-score values 100%, while overall accuracy was calculated as 100%. Both the macro-average and weighted-average metrics reached 100%, indicating a highly consistent and balanced performance between the two classes.

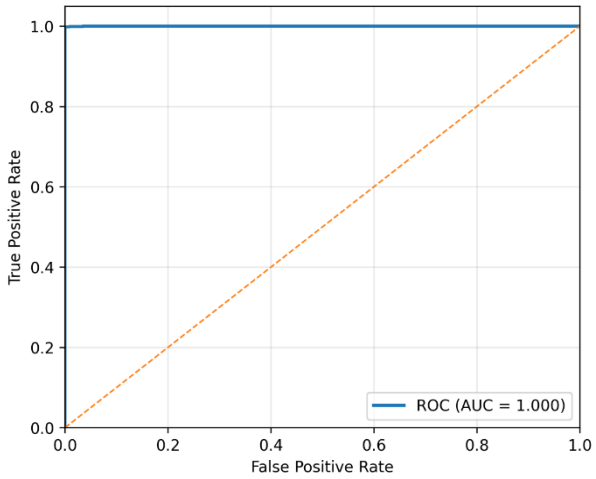

Figure 12. ROC curve and AUC value of the MobileNetV2 model

Figure 12 presents the ROC curve of the MobileNetV2 model on the test dataset. The area under the curve is calculated as 1.000, based on the relationship between the true positive rate and the false positive rate.

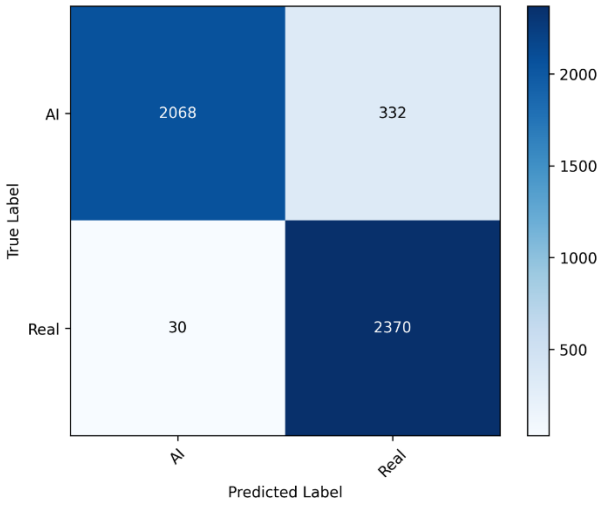

Figure 13. Confusion Matrix of the NASNetMobile model

Figure 13 shows the classification results of the NASNetMobile model on the test dataset. The model produced 2,068 correct and 332 incorrect predictions for the AI class, and 2,370 correct and 30 incorrect predictions for the Real class.

Table 12. Performance evaluation metrics of the NASNetMobile model

Class / Metric	Precision (%)	Recall (%)	F1-score (%)
AI	99.00	86.00	92.00
Real	88.00	99.00	93.00
Accuracy	-	-	92.00
Macro avg	92.00	92.00	92.00
Weighted avg	92.00	92.00	92.00

Table 12 presents the classification performance metrics of the NASNetMobile model for the AI and Real classes. For the AI class, the precision and recall values are 99% and 86%, respectively, while for the Real class, the precision is 88% and the recall is 99%. The corresponding F1-scores are 92% for the AI class and 93% for the Real class. Both macro-averaged and weighted - averaged metrics are 92%.

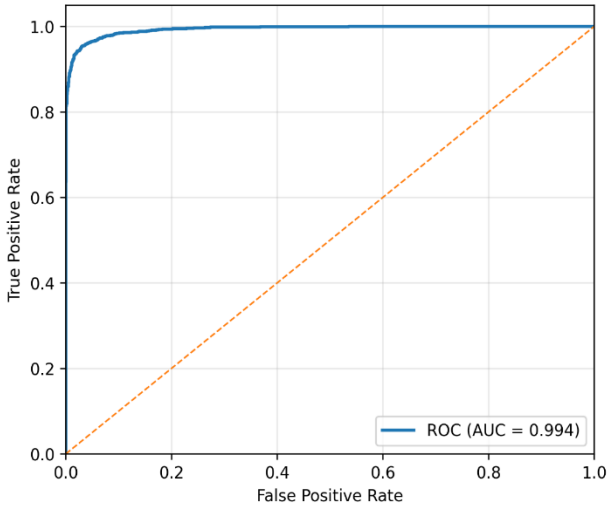

Figure 14. ROC curve and AUC value of the NASNetMobile model

Figure 14 presents the ROC curve of the NASNetMobile model on the test dataset. The area under the curve is calculated as 0.994 based on the relationship between the true positive rate and the false positive rate.

According to Table 7, MobileNetV2 (99.58%) and DenseNet201 (99.56%) achieved the highest success rates; DenseNet169 also produced a similarly strong performance. These results show that success in tasks such as real - synthetic discrimination is influenced not only by overall accuracy but also by representational power, particularly the ability to capture subtle texture/artifact clues. In DenseNet architectures, dense inter - layer connections enable more effective transfer of low-level texture and edge cues learned in early layers to deeper

layers, potentially providing an advantage in distinguishing diffusion-induced micro-inconsistencies. MobileNetV2's ability to achieve similar accuracy with shorter training time demonstrates that lightweight models can also be highly competitive in this task with the right architectural design.

In contrast, NASNetMobile's lower performance 92.46% can be attributed more to task suitability than to the architecture's representational capacity. NASNetMobile is designed with architecture search to optimize semantic discrimination, mostly for general-purpose classification. However, synthetic–real discrimination requires sensitivity to high-frequency texture cues and small artifacts, independent of class semantics. The cell-based and more selective feature summarization structure in NASNetMobile may have led to the suppression of these weak signals in some cases and resulted in more errors, particularly in the synthetic class. Indeed, despite NASNetMobile's relatively high recall, its lower accuracy / F1 score suggests that the model has an increased tendency for false positives/misclassification in some synthetic examples and that the decision boundary remains more unstable.

In conclusion, the findings show that the DenseNet family and MobileNetV2 provide more appropriate representation in real–synthetic discrimination, while NASNetMobile cannot provide the same level of fine-grained artifact sensitivity required for this task.

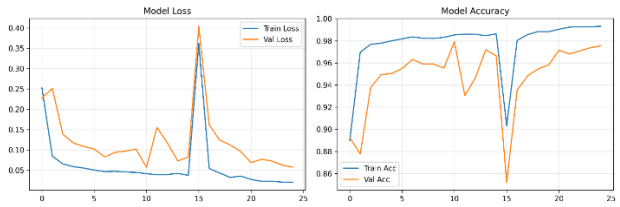

Figure 15. Training loss and accuracy curves of the DenseNet121 model

As illustrated in Figure 15, the training and validation loss and accuracy curves of the DenseNet121 model are presented. As illustrated by the loss curves, the convergence behavior across epochs is evident. Concurrently, the accuracy curves demonstrate the progression of training and validation accuracy during the training process.

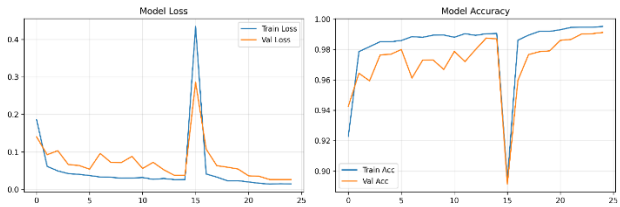

Figure 16. Training loss and accuracy curves of the DenseNet169 model

As illustrated in Figure 16, the training and validation loss and accuracy curves of the DenseNet169 model are presented. The loss curves demonstrate the convergence behavior over epochs, while the accuracy curves illustrate the evolution of training and validation accuracy throughout the training process.

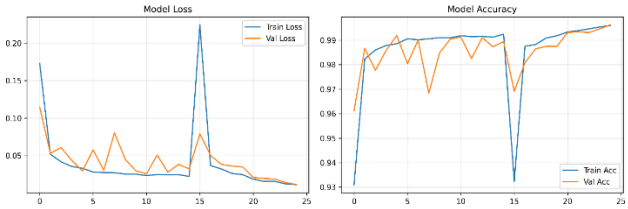

Figure 17. Training loss and accuracy curves of the DenseNet201 model

As illustrated in Figure 17, the training and validation loss and accuracy curves of the DenseNet201 model are presented. The loss curves demonstrate the convergence pattern across epochs, while the accuracy curves illustrate the progression of training and validation accuracy during the training process.

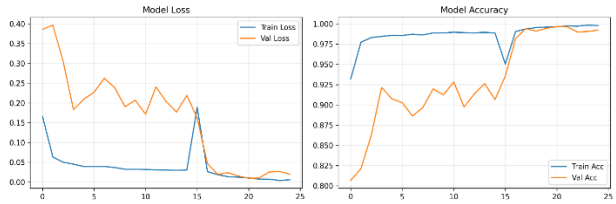

Figure 18. Training loss and accuracy curves of the MobileNetV2 model

As illustrated in Figure 18, the training and validation loss and accuracy curves of the MobileNetV2 model are presented. The loss curves demonstrate the convergence behavior over the periods, while the accuracy curves illustrate the enhancement of training and validation accuracy throughout the training process. In contrast to other DenseNet models, the MobileNetV2 model exhibited a more consistent loss and validation curve. The fluctuations in loss experienced during training periods are conspicuously diminished.

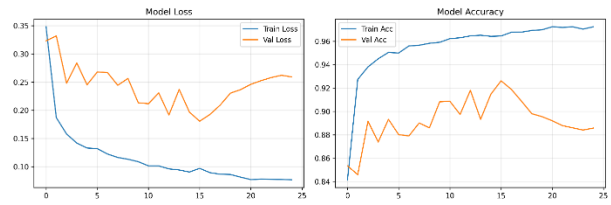

Figure 19. Training loss and accuracy curves of the NASNetMobile model

Figure 19 shows the training and validation loss and accuracy curves of the NASNetMobile model. The curves show the behavior of loss and accuracy values for both training and validation sets throughout the training process. The NASNetMobile model exhibited a lower validation accuracy curve during training while showing higher validation loss compared to the other four models.

This can be explained as follows: NASNetMobile's cell-based architecture and relatively complex search-space design may not be able to represent high-frequency cues critical for discrimination in binary classification problems focused on “fine texture/artifacts,” such as real–synthetic discrimination, in a sufficiently stable manner. Therefore, even if the model predicts the correct class in some examples, it may produce more “uncertain” predictions by generating low probability margins in its outputs. This behavior is consistent with the high loss value in the validation set, as cross-entropy loss penalizes not only correct/incorrect decisions but also the confidence level (calibration) of the decision. Furthermore, the relatively low F1/accuracy values reported for NASNetMobile in the previous section support this weak generalization behavior observed in the curves.

When other architectures are examined, a more stable and consistent learning dynamic is noticeable. The steady decrease in training and validation loss in the DenseNet121/169/201 models and the close tracking of accuracy curves indicate that the learned representations better align with the data distribution. The dense connectivity structure of the DenseNet family enables the effective transfer of low-level features, such as edge-texture, learned in early layers to deep layers, potentially providing an advantage in distinguishing micro-inconsistencies arising from diffusion-based generation. In particular, DenseNet201's performance, which approaches the lowest error rate, indicates that higher representation capacity can be beneficial in this task; however, an increase in training time is an expected result.

On the MobileNetV2 side, it is seen that the curves converge faster and the validation performance remains high despite the shorter training time. This finding shows that MobileNetV2, thanks to inverted residuals and depth-separable convolutions, is both computationally efficient and able to learn sufficient discriminative features in this binary classification problem. In conclusion, training curve analysis reveals that the DenseNet family stands out with its high representational power, MobileNetV2 with its efficiency-performance balance, and NASNetMobile with its inability to produce the fine-grained artifact sensitivity and stable generalization behavior required for this problem to the same extent.

4. CONCLUSIONS

The present study developed a deep learning-based classification framework to distinguish AI-generated images from real images. The study also comprehensively evaluated the performance of various CNN architectures. MobileNetV2, DenseNet121, DenseNet169, DenseNet201, and NASNetMobile were subjected to a rigorous evaluation process that utilized key classification metrics, including accuracy, sensitivity, recall, F1 score, and ROC/AUC. This evaluation also incorporated the analysis of confusion matrices and training curves. As illustrated in Table 7, MobileNetV2 99.58% and DenseNet201 99.56% demonstrated the highest overall accuracy, while NASNetMobile 92.46% exhibited a substantial decline in performance compared to the other models. The attainment of a 100% recall rate by all DenseNet models serves to substantiate their remarkable sensitivity in identifying AI-generated images.

An analysis of the confusion matrices indicates that the DenseNet family attains perfect classification for the Real class and generates a minimal number of errors in the AI class. Among the evaluated models, DenseNet201 showed low error rates compared to other DenseNet models, misclassifying only 21 examples, while MobileNetV2 showed the lowest error rates, misclassifying 17 examples in the AI class and 3examples in the Real class. The high rate of false positives observed for NASNetMobile, in which AI-generated images were erroneously classified as authentic, signifies challenges in accurately representing the variability of synthetic image patterns.

Training curves demonstrate that DenseNet models demonstrate rapid convergence; However, transient variations in validation loss are observed around the 15th epoch. This phenomenon suggests a transient instability in the system's capacity to adapt to complex data patterns. MobileNetV2 demonstrates notable variations in validation loss; Nevertheless, stability in subsequent epochs signifies the emergence of robust generalization capabilities. The observed rising validation loss trend for NASNetMobile points to a precursor trend toward overfitting and limited generalization capabilities in complex patterns.

A comparison of the models utilization in this research with those documented in the extant literature reveals their superior performance in comparable classification tasks. This enhancement can be attributed to the efficacy of hyperparameter optimization and the utilization of a high-quality, task specific dataset. Moreover, by incorporating training times alongside classification accuracy, the proposed methodology addresses a substantial gap in the extended literature with respect to both performance and computational efficiency.

This work corroborates earlier findings that conventional detection methods are inadequate when confronted with more realistic images generated by contemporary generative models, including GANs and diffusion-based techniques. The paper makes a substantial

contribution to the field by offering a comprehensive evaluation of various CNN architectures for AI-generated image detection. Future research endeavors should focus on incorporating multimodal features, investigating Vision Transformer-based architectures, and examining generative model fingerprints to enhance detection efficiency.

Declaration of Ethical Standards

This study does not involve any human participants or animal testing. Therefore, ethical approval was not required. All data used in this research were obtained from publicly available sources or generated in the laboratory under standard conditions.

Credit Authorship Contribution Statement

Author Contributions: Conceptualization: IA and OK; Methodology: IA and OK; Software: IA; Validation: OK and MK Formal analysis: IA and MK; Investigation: IA and OK; Resources: IA; Data curation: IA and OK; Writing original draft preparation: IA, OK and MK; Writing review and editing: OK and MK; Visualization: IA; Supervision: MK; Project administration: MK. All authors have read and agreed to the published version of the manuscript.

Declaration of Competing Interest

The authors declare that there are no conflicts of interest related to this work.

Funding / Acknowledgements

This research received no external funding. The authors would like to thank the Department of Computer Engineering, Faculty of Technology, Selçuk University, for providing the laboratory facilities used in this study.

Data Availability

The dataset used in this study was obtained from the Kaggle platform and is available at the following link: https://www.kaggle.com/datasets/lshgrandvoyage/animal-images-with-real-and-ai-generatedsd-turbo

References

[1] G. Celik and M. F. Talu, "Investigation of generative adversarial network models' image generation performance," *Balıkesir University Journal of Science and Engineering,* vol. 22, no. 1, pp. 181-192, 2020, doi: https://doi.org/10.25092/baunfbed.679608.

[2] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," *Neurocomputing,* vol. 321, pp. 321-331, 2018, doi: https://doi.org/10.1016/j.neucom.2018.09.013.

[3] P. Sedigh, R. Sadeghian, and M. T. Masouleh, "Generating synthetic medical images by using GAN to improve CNN performance in skin cancer classification," presented at the 2019 7th international conference on robotics and mechatronics, 2019. doi: 10.1109/ICRoM48714.2019.9071823.

[4] C. Han *et al.*, "GAN-based synthetic brain MR image generation," presented at the 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018), 2018. doi: 10.1109/ISBI.2018.8363678.

[5] H. Nazki, J. Lee, S. Yoon, and D. S. Park, "Image-to-image translation with GAN for synthetic data augmentation in plant disease datasets," *Smart Media Journal,* vol. 8, no. 2, pp. 46-57, 2019.

[6] J. J. Bird, C. M. Barnes, L. J. Manso, A. Ekárt, and D. R. Faria, "Fruit quality and defect image classification with conditional GAN data augmentation," *Scientia Horticulturae,* vol. 293, p. 110684, 2022, doi: https://doi.org/10.1016/j.scienta.2021.110684.

[7] F. Wang, H. Al Hamadi, and E. Damiani, "A visualized malware detection framework with CNN and conditional GAN," presented at the 2022 IEEE International Conference on Big Data (Big Data), 2022. doi: 10.1109/BigData55660.2022.10020534.

[8] C. Dewi, R.-C. Chen, Y.-T. Liu, and S.-K. Tai, "Synthetic Data generation using DCGAN for improved traffic sign recognition," *Neural Computing and Applications,* vol. 34, no. 24, pp. 21465-21480, 2022, doi: https://doi.org/10.1007/s00521-021-05982-z.

[9] S. M. Sristi and M. S. Sharma, "Culturally-Aware Multiclass Bangla Text-to-Image Generation via Fine-Tuned Stable Diffusion Turbo," presented at the 2025 IEEE 11th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService), 2025. doi: 10.1109/BigDataService65758.2025.00025.

[10] A. Lokner Lađević, T. Kramberger, R. Kramberger, and D. Vlahek, "Detection of AI-generated synthetic images with a lightweight CNN," *AI,* vol. 5, no. 3, pp. 1575-1593, 2024, doi: https://doi.org/10.3390/ai5030076.

[11] Y. Patel *et al.*, "An improved dense CNN architecture for deepfake image detection," *IEEE Access,* vol. 11, pp. 22081-22095, 2023, doi: 10.1109/ACCESS.2023.3251417.

[12] J. J. Bird and A. Lotfi, "Cifake: Image classification and explainable identification of ai-generated synthetic images," *IEEE Access,* vol. 12, pp. 15642-15650, 2024, doi: 10.1109/ACCESS.2024.3356122.

[13] J. Mallet, L. Pryor, R. Dave, and M. Vanamala, "Deepfake detection analyzing hybrid dataset utilizing cnn and svm," presented at the Proceedings of the 2023 7th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, 2023. doi: https://doi.org/10.1145/3596947.359695.

[14] A. Raza, K. Munir, and M. Almutairi, "A novel deep learning approach for deepfake image detection," *Applied Sciences,* vol. 12, no. 19, p. 9820, 2022, doi: https://doi.org/10.3390/app12199820.

[15] W. H. Abir *et al.*, "Detecting deepfake images using deep learning techniques and explainable AI methods," *Intelligent Automation & Soft Computing,* vol. 35, no. 2, pp. 2151-2169, 2023, doi: https://doi.org/10.32604/iasc.2023.029653.

[16] H. Chen *et al.*, "Comprehensive exploration of diffusion models in image generation: a survey," *Artificial Intelligence Review,* vol. 58, no. 4, p. 99, 2025, doi: https://doi.org/10.1007/s10462-025-11110-3.

[17] F. Rahat, M. S. Hossain, M. R. Ahmed, S. K. Jha, and R. Ewetz, "Data augmentation for image classification using generative ai," presented at the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2025. doi: 10.1109/WACV61041.2025.00410.

[18] F. Siddiqui, J. Yang, S. Xiao, and M. Fahad, "Enhanced deepfake detection with DenseNet and Cross-ViT," *Expert Systems with Applications,* vol. 267, p. 126150, 2025, doi: https://doi.org/10.1016/j.eswa.2024.126150.

[19] S. Banerjee, "Neural Architecture Search Based Deepfake Detection Model using YOLO," *International Journal of Advanced Research in Science, Communication and Technology,* pp. 375-383, 2025, doi: 10.48175/IJARSCT-22938.

[20] Y. Wu and K. He, "Group normalization," presented at the Proceedings of the European conference on computer vision (ECCV), 2018. [Online]. Available: https://link.springer.com/conference/eccv.

[21] F. He, T. Liu, and D. Tao, "Control batch size and learning rate

to generalize well: Theoretical and empirical evidence," *Advances in neural information processing systems,* vol. 32, 2019.

[22] P. M. Radiuk, "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets," 2017, doi: https://doi.org/10.1515/itms-2017-0003.

[23] M. J. Awan *et al.*, "Image-based malware classification using VGG19 network and spatial convolutional attention," *Electronics,* vol. 10, no. 19, p. 2444, 2021, doi: https://doi.org/10.3390/electronics10192444.

[24] P. S. Madhyastha and R. Jain, "On model stability as a function of random seed," presented at the Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), 2019. doi: 10.18653/v1/K19-1087.

[25] J. Åkesson, J. Töger, and E. Heiberg, "Random effects during training: Implications for deep learning-based medical image segmentation," *Computers in Biology and Medicine,* vol. 180, p. 108944, 2024, doi: https://doi.org/10.1016/j.compbiomed.2024.108944.

[26] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 model for image classification," presented at the 2020 2nd International Conference on Information Technology and Computer Application (ITCA), 2020. doi: 10.1109/ITCA52113.2020.00106.

[27] T. A. Cengel, B. Gencturk, E. T. Yasin, M. B. Yildiz, I. Cinar, and M. Koklu, "Automating egg damage detection for improved quality control in the food industry using deep learning," *Journal of Food Science,* vol. 90, no. 1, p. e17553, 2025, doi: https://doi.org/10.1111/1750-3841.17553.

[28] L. Yong, L. Ma, D. Sun, and L. Du, "Application of MobileNetV2 to waste classification," *Plos one,* vol. 18, no. 3, p. e0282336, 2023, doi: https://doi.org/10.1371/journal.pone.0282336.

[29] M. Eser, M. Bilgin, E. T. Yasin, and M. Koklu, "Using pretrained models in ensemble learning for date fruits multiclass classification," *Journal of Food Science,* vol. 90, no. 3, p. e70136, 2025, doi: https://doi.org/10.1111/1750-3841.70136.

[30] R. Indraswari, R. Rokhana, and W. Herulambang, "Melanoma image classification based on MobileNetV2 network," *Procedia computer science,* vol. 197, pp. 198-207, 2022, doi: https://doi.org/10.1016/j.procs.2021.12.132.

[31] Y. Unal and M. Turkoğlu, "Mango leaf disease detection using deep feature extraction and machine learning methods: A comparative survey," *El-Cezeri,* vol. 12, no. 1, pp. 35-43, 2025.

[32] [32] A. Bello, S.-C. Ng, and M.-F. Leung, "Skin cancer classification using fine-tuned transfer learning of DENSENET-121," *Applied Sciences,* vol. 14, no. 17, p. 7707, 2024, doi: https://doi.org/10.3390/app14177707.

[33] S. A. Albelwi, "Deep architecture based on DenseNet-121 model for weather image recognition," *International Journal of Advanced Computer Science and Applications,* vol. 13, no. 10, 2022, doi: 10.14569/IJACSA.2022.0131065.

[34] N. A. Zebari, A. A. Alkurdi, R. B. Marqas, and M. S. Salih, "Enhancing brain tumor classification with data augmentation and densenet121," *Academic Journal of Nawroz University,* doi: https://doi.org/10.25007/ajnu.v12n4a1985.

[35] S. Ogrekci, Y. Unal, and M. N. Dudak, "A comparative study of vision transformers and convolutional neural networks: sugarcane leaf diseases identification," *European Food Research and Technology,* vol. 249, no. 7, pp. 1833-1843, 2023.

[36] S. Harmanci, U. Yavuz, and A. Baris, "Classification of hazelnut species with pre-trained deep learning models," *Intelligent Methods In Engineering Sciences,* vol. 2, no. 2, pp. 58-66, 2023.

[37] B. A. S. Al-Rimy, F. Saeed, M. Al-Sarem, A. M. Albarrak, and S. N. Qasem, "An adaptive early stopping technique for densenet169-based knee osteoarthritis detection model," *Dgnostics,* vol. 13, no. 11, p. 1903, 202, doi: https://doi.org/10.3390/diagnostics13111903.

[38] D. Bhowmik, M. Abdullah, and M. T. Islam, "A deep face-mask detection model using DenseNet169 and image processing techniques," Brac University, 2022.

[39] P. P. Dalvi, D. R. Edla, and B. Purushothama, "Diagnosis of coronavirus disease from chest X-ray images using DenseNet-169 architecture," *SN Computer Science,* vol. 4, no. 3, p. 214, 2023, doi: https://doi.org/10.1007/s42979-022-01627-7.

[40] M. K. Awang *et al.*, "Classification of Alzheimer disease using DenseNet-201 based on deep transfer learning technique," *Plos one,* vol. 19, no. 9, p. e0304995, 2024, doi: https://doi.org/10.1371/journal.pone.0304995.

[41] A. Dash, P. K. Sethy, and S. K. Behera, "Maize disease identification based on optimized support vector machine using deep feature of DenseNet201," *Journal of Agriculture and Food Research,* vol. 14, p. 100824, 2023, doi: https://doi.org/10.1016/j.jafr.2023.100824.

[42] A. Ghodekar and A. Kumar, "LightLeafNet: Lightweight and efficient NASNetmobile architecture for tomato leaf disease classification," *Available at SSRN 4570347,* 2023, doi: http://dx.doi.org/10.2139/ssrn.4570347.

[43] A. E. B. Alawi and A. M. Qasem, "Lightweight CNN-based models for masked face recognition," presented at the 2021 International Congress of Advanced Technology and Engineering (ICOTEN), 2021. doi: 10.1109/ICOTEN52080.2021.9493424.

[44] H. Guven and A. Saygili, "Monkeypox Diagnosis Using MRMR-Based Feature Selection and Hybrid Deep Learning Models: ResNet50V2, NASNetMobile, and InceptionV3," *International Scientific and Vocational Studies Journal,* vol. 9, no. 1, pp. 173-182, 2025, doi: https://doi.org/10.47897/bilmes.1706322.

[45] A. W. A. Ameer, P. Salehpour, and M. Asadpour, "Deep transfer learning for lip reading based on NASNetMobile pretrained model in wild dataset," *IEEE Access,* 2024, doi: 10.1109/ACCESS.2024.3470521.

[46] I. A. Ozkan and M. Koklu, "Skin lesion classification using machine learning algorithms," *International Journal of Intelligent Systems and Applications in Engineering,* vol. 5, no. 4, pp. 285-289, 2017, doi: 10.18201/ijisae.2017534420.

[47] Y. Karatas *et al.*, "Identification of Leaf Diseases from Figs Using Deep Learning Methods," *Selcuk Journal of Agriculture and Food Sciences,* vol. 38, no. 3, pp. 414-426, 2024, doi: 10.15316/SJAFS.2024.037.

[48] E. Yasin and M. Koklu, "A comparative analysis of machine learning algorithms for waste classification: inceptionv3 and chi-square features," *International Journal of Environmental Science and Technology,* vol. 22, no. 10, pp. 9415-9428, 2025, doi: https://doi.org/10.1007/s13762-024-06233-z.

[49] R. Kursun and M. Koklu, "Classification of Eggplant Diseases Using Feature Extraction with AlexNet and Random Forest," presented at the 4th International Conference on ContemporaryAcademic Research, Konya, Turkey, February 22-23, 2025, 2025.

[50] H. Incekara, I. H. Cizmeci, M. M. Saritas, and M. Koklu, "Classification of almond kernels with optuna hyper-parameter optimization using machine learning methods," *Journal of Food Science and Technology,* pp. 1-17, 2025, doi: https://doi.org/10.1007/s13197-025-06494-7.

[51] T. A. Cengel, B. Gencturk, E. T. Yasin, M. B. Yildiz, I. Cinar, and M. Koklu, "Apple (malus domestica) quality evaluation based on analysis of features using machine learning techniques," *Applied Fruit Science,* vol. 66, no. 6, pp. 2123-2133, 2024, doi: https://doi.org/10.1007/

[52] M. Koklu, R. Kursun, E. T. Yasin, and Y. S. Taspinar, "Detection of Defects in Soybean Seeds by Extracting Deep Features with SqueezeNet," presented at the 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications doi: 10.1109/IDAACS58523.2023.10348939.

[53] M. C. Hinojosa Lee, J. Braet, and J. Springael, "Performance metrics for multilabel emotion classification: comparing micro, macro, and weighted f1-scores," *Applied Sciences,* vol. 14, no. 21, p. 9863, 2024, doi: https://doi.org/10.3390/app14219863.

[54] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," *Computers and Electronics in Agriculture,* vol. 174, p. 105507,

2020, doi: https://doi.org/10.1016/j.compag.2020.105507.
[55] M. Koklu, I. Cinar, and Y. S. Taspinar, "Classification of rice varieties with deep learning methods," *Computers and electronics in agriculture,* vol. 187, p. 106285, 2021, doi: https://doi.org/10.1016/j.compag.2021.106285.
[56] Y. S. Taspinar, I. Cinar, R. Kursun, and M. Koklu, "Monkeypox Skin Lesion Detection with Deep Learning Models and Development of Its Mobile Application," presented at the International Journal of Research in Engineering and Science (IJRES), January 2024, 2024.
[57] Y. S. Taspinar, I. Cinar, and M. Koklu, "Prediction of computer type using benchmark scores of hardware units," presented at the Selcuk University Journal of Engineering Sciences April 2021, 2021.