World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Check for updates

Generative Artificial Intelligence for image synthesis using Generative Adversarial Networks (GANs) and variational autoencoders

Manoj T S [1], Kumar Siddamallappa U [1] and Anusha Jajur. J [2,*]

[1] *Department of Studies in Computer Applications (MCA), Davanagere University, Davangere, Karnataka, India.*
[2] *Department of Studies in Computer Science, Davanagere University, Davangere, Karnataka, India.*
Kumar Siddamallappa U; ORCID: 0000-0002-1975-3868
Anusha Jajur. J; ORCID: 0009-0003-9835-624X

World Journal of Advanced Research and Reviews, 2026, 31(01), 1233–1245

Publication history: Received on 13 June 2026; revised on 19 July 2026; accepted on 21 July 2026

Article DOI: https://doi.org/10.30574/wjarr.2026.31.1.1945

Abstract

In recent years, the rapid advancement of Generative Artificial Intelligence (GenAI) has transformed the landscape of digital content creation, enabling high-fidelity image synthesis across various fields including healthcare, art, and computer vision. However, the proliferation of synthesized images has introduced critical challenges, specifically the need to distinguish real physical imagery from synthetic, AI-generated counterfeits. This research paper presents a comprehensive, end-to-end framework that addresses both generative synthesis and discriminative detection under hardware-constrained (CPU-only) environments. We implement two synthesis methodologies like a Deep Convolutional Generative Adversarial Network (DCGAN) and a Convolutional Variational Autoencoder (ConvVAE) - trained on real image distributions to generate synthetic data. Concurrently, we present a compact Convolutional Neural Network (MiniCNNClassifier) designed to detect and classify images as real or fake. The framework is validated using a balanced dataset of 60,000 images (30,000 real and 30,000 synthetic). Our preprocessing pipeline ensures uniform size and resolution across heterogeneous inputs. Experimental results demonstrate that the MiniCNNClassifier achieves an outstanding validation accuracy of 98.7% and a Precision of 99.5%, Recall of 97.8%, F1-score of 98.6% in detecting fake samples. Furthermore, we provide a qualitative and quantitative comparison of DCGAN and ConvVAE architectures, discussing trade-offs between training stability and sample fidelity. Finally, we host the models on an interactive Streamlit-based web interface to enable real-time generation and classification.

Keywords: Deep Convolutional GAN (DCGAN); Variational Autoencoder (VAE); Convolutional Neural Network (CNN); Image Synthesis; Deepfake Detection.

1. Introduction

Generative Artificial Intelligence (GenAI) has emerged as one of the fastest growing fields in modern artificial intelligence, transforming the way digital content is created, processed, and analysed. By leveraging deep learning techniques, generative models learn the underlying probability distribution of complex datasets and produce new data samples that closely resemble real-world observations. These capabilities have significantly influenced applications in image synthesis, medical imaging, computer vision, entertainment, virtual reality, and digital content generation. Among the various generative models, Generative Adversarial Networks (GANs), introduced by Goodfellow et al. [13], have become one of the most influential approaches due to their ability to generate highly realistic and high-quality synthetic images.

* Corresponding author: Manoj T S

A typical GAN consists of two competing neural networks: a Generator and a Discriminator. The Generator attempts to create realistic images from random noise, while the Discriminator distinguishes between real and generated images. Through this adversarial learning process, both networks improve iteratively, enabling the Generator to produce increasingly realistic images. More recently, Variational Autoencoders (VAEs) have also gained considerable attention as an alternative generative architecture. Unlike GANs, VAEs learn a structured latent representation of the input data and reconstruct images through probabilistic encoding and decoding. Although VAEs generally produce smoother images, they offer more stable training and better latent space representation, making them suitable for controlled image synthesis and reconstruction tasks.

The rapid advancement of image generation technologies has created numerous opportunities across various domains. In healthcare, synthetic medical images can be generated to augment limited datasets while preserving patient privacy. In computer vision, synthetic datasets improve the robustness of machine learning models by increasing data diversity. The entertainment and gaming industries utilize generative models to create realistic characters, environments, and digital artwork, reducing development time and production costs. Similarly, industries such as fashion, advertising, education, and scientific visualization increasingly rely on AI-generated content to support creative and analytical tasks.

Despite these remarkable achievements, the ability to generate highly realistic synthetic images has introduced significant challenges related to digital authenticity, misinformation, cybersecurity, and privacy. The emergence of deepfake technology has demonstrated how AI-generated images and videos can be misused to manipulate identities, spread false information, impersonate individuals, and compromise digital trust. As synthetic media becomes increasingly indistinguishable from genuine content, accurately detecting AI-generated images has become a critical research problem. Consequently, image generation and image classification must be studied together to ensure both technological advancement and responsible deployment of generative AI systems.

Motivated by these challenges, this research presents a comprehensive framework that combines image synthesis and image classification within a unified pipeline. Two state-of-the-art generative models, namely Deep Convolutional Generative Adversarial Network (DCGAN) and Convolutional Variational Autoencoder (ConvVAE), are implemented to generate synthetic facial images from real image distributions. In parallel, a lightweight Convolutional Neural Network (MiniCNNClassifier) is developed to classify images as either real or fake with high accuracy. The proposed framework is trained and evaluated using a balanced dataset consisting of 60,000 images, including 30,000 real and 30,000 synthetic images. A preprocessing pipeline standardizes images with varying resolutions into a consistent RGB format suitable for deep learning.

To demonstrate practical usability, all trained models are integrated into an interactive Streamlit-based web application that enables users to generate synthetic images using DCGAN or ConvVAE and classify uploaded images in real time. Model performance is evaluated using standard metrics including Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and training loss curves. The study also provides a comparative analysis of DCGAN and ConvVAE with respect to image quality, training stability, computational efficiency, and suitability for CPU-only environments.

1.1. Problem Statement

Training state-of-the-art generative networks (such as StyleGAN or Diffusion Models) requires massive datasets and high-performance GPU hardware. In resource-constrained environments (e.g., standard consumer-grade laptops without dedicated GPUs), training these models is computationally prohibitive. The problem, therefore, is to design:

- A compact, stable, and computationally efficient generative model capable of producing high-quality synthetic images on CPU-only architectures using the Deep Convolutional GAN (DCGAN) framework.
- An alternative non-adversarial reconstruction model, such as a Variational Autoencoder (VAE) compared alongside GANs in reviews like [9], to compare training dynamics and image quality under the same hardware limitations.
- A highly accurate, low-latency binary classifier to differentiate real images from synthetic ones [7], mitigating the risk of AI-generated misinformation.
- An integrated application that combines these components into an accessible user interface.

1.2. Research Objectives

The project focuses on integrating a Convolutional Neural Network (CNN) classifier to accurately distinguish between authentic and GAN generated images, thereby evaluating the quality and realism of the synthesized outputs. By combining image generation and image classification within a single framework, the project demonstrates the practical application of deep learning in generative artificial intelligence, image analysis, and digital content authentication.

2. Literature review

The research is positioned at the intersection of generative modelling and synthetic image detection. To contextualize our methodology, we review thirteen key reference papers addressing various facets of GAN applications, synthetic data generation, and evaluation.

Majidpour et al. [1] reviewed the applications of Generative Adversarial Networks (GANs) in breast cancer detection. The authors highlight how GANs are utilized to address class imbalance and data scarcity by synthesizing high-fidelity mammographic and ultrasound images, which are subsequently used to train stronger classifiers. This highlights the utility of generative models as data augmentation tools, similar to our approach of training a classifier on synthetic samples.

Waseem et al. [2] examined the role of Generative AI in healthcare, focusing on synthetic data generation. The paper details how synthetic datasets protect patient privacy and overcome patient data limitations. The authors categorize generative models into GANs, VAEs, and Diffusion models, outlining their mathematical foundations and practical utility, which aligns with our implementation of both DCGAN and ConvVAE.

Singh et al. [3] provided a detailed architectural comparison between GANs, Diffusion Models, and Large Language Models. The authors discuss the theoretical advancements that solved early training instabilities in GANs (such as Wasserstein loss and gradient penalties) and compare GANs with Diffusion models in terms of synthesis quality and generation speed, noting that GANs remain highly advantageous for fast, single-step inference.

Balasubramanian et al. [4] detailed the evolution of image generation models from traditional autoencoders to modern generative pipelines. It discusses how generative models are unlocking digital creativity, detailing the architectural differences in how latent spaces are structured and manipulated, providing a theoretical foundation for our choice of a 128-dimensional latent noise vector.

Masalkhi et al. [5] demonstrated the practical application of GANs in synthesizing ophthalmic images (e.g., fundus photographs and optical coherence tomography). They analyze the visual quality of synthesized images using clinical experts and neural-network-based features, proving that synthetic ophthalmic images can serve as valid training data for diagnostic models.

Dhoni & Jain [6] outlined the advantages of integrating GenAI into image processing pipelines, including super-resolution, style transfer, and image restoration. The authors show how generative models learn complex texture details, which helps explain the differences in edge and noise distributions that classifiers use to distinguish real and synthetic images.

Al-Muttairi et al. [7] conducted a systematic review of Deep Convolutional GANs (DCGAN) for forensic applications, specifically evaluating the performance and challenges of forensic face models. The study details how DCGANs learn features, the challenges of identifying AI-synthesized faces, and the forensic signatures left by generator architectures, which directly informs our development of detection classifiers to distinguish real and synthetic samples.

Algethami et al. [8] extended the generative paradigm to the temporal domain, reviewing video synthesis for biomedical applications. The paper reviews how 3D-convolutional GANs and recurrent architectures generate sequential frames, indicating that the foundational principles of spatial image synthesis (implemented in our project) form the basis of advanced video generation systems.

Bond-Taylor et al. [9] presented a comprehensive, comparative review of deep generative models, detailing the mathematical formulations and operational trade-offs of VAEs, GANs, normalizing flows, energy-based models, and autoregressive models. Their work highlights key differences in training stability, latent space structure, and sample quality, providing the theoretical basis for our comparison between DCGAN (adversarial) and ConvVAE (likelihood-based).

Jeong et al. [10] systematically reviewed the use of GANs in medical image classification and segmentation tasks. The authors demonstrate that training segmentation and classification networks using a mix of real and GAN-generated images significantly improves generalization performance and reduces overfitting on small datasets.

Islam & Zhang [11] presented a concrete implementation of GANs for generating synthetic brain Positron Emission Tomography (PET) scans. They detail their preprocessing pipeline and adversarial training parameters, demonstrating

that the generated images match the anatomical distribution of real PET scans, proving that low-resolution GAN architectures can capture complex medical features.

Wang et al. [12] proposed an innovative evaluation methodology that uses human neural signals (EEG) to assess the visual quality of GAN-generated faces. Their work highlights the limitations of purely mathematical metrics (like Frechet Inception Distance) and emphasizes the importance of human-centric evaluation and visual inspection, which we address in our Streamlit dashboard via qualitative validation.

Goodfellow et al. [13] introduced the foundational framework of Generative Adversarial Networks (GANs), styling the generation process as a minimax game between a generative model (Generator) and a discriminative model (Discriminator). This seminal work established the mathematical convergence proofs and adversarial training paradigm that underpin all modern GAN architectures, including the DCGAN implemented in this study.

3. proposed methodology

The architecture of our proposed system consists of four interconnected phases like Dataset Preprocessing, Generative Synthesis (using DCGAN and ConvVAE), Real/Fake Detection, and Web Deployment as shown in Fig. 1.

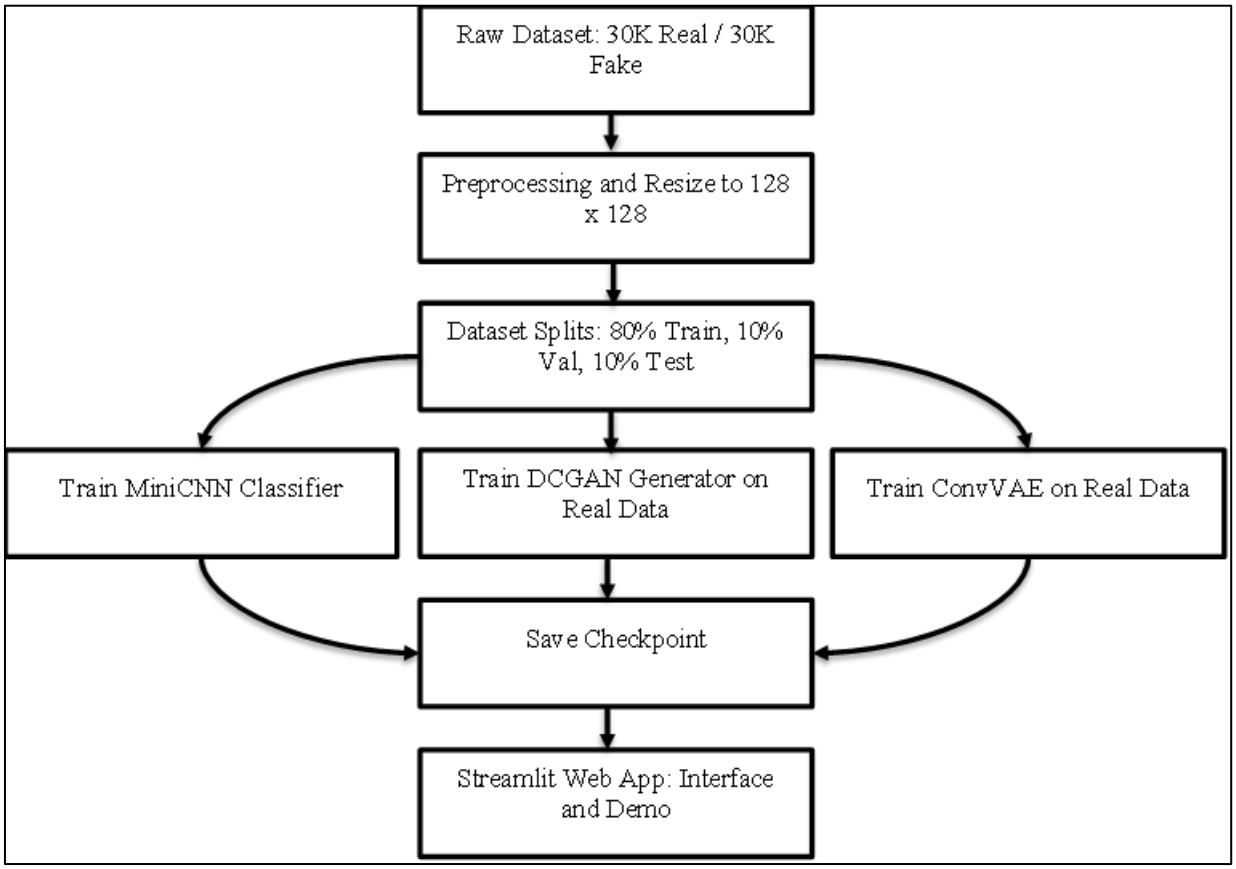

Figure 1 Complete project workflow diagram including preprocessing, splits, model training, and Streamlit inference.

3.1. Dataset Description

The dataset contains a total of 60,000 images, split equally between two classes:

- Real Images (30,000 samples): High-resolution natural photography as shown in Fig. 2.
- Fake/AI-generated Images (30,000 samples): Synthetic images generated by various generative algorithms as shown in Fig. 3.

The raw dataset exhibits heterogeneous dimensions: real images are mostly 256 times 256 pixels, while synthetic images vary, with some small fake images stored at 128 times 128 pixels.

Figure 2 Sample real images from the dataset**.**

Figure 3 Sample fake/AI-generated images from the dataset.

3.2. Image Preprocessing

To ensure consistency across the pipeline, we implement a standardized preprocessing script such as:

- Image Validation: Load images and filter out corrupted files, converting all inputs to RGB format.
- Resizing: Resize all images to a uniform resolution of 128 times 128 (or 64 times 64 for rapid testing) using high-quality bilinear interpolation.
- Normalization:

For the Classifier and DCGAN: Normalize images to a range of [-1, 1] using a mean of (0.5, 0.5, 0.5) and a standard deviation of (0.5, 0.5, 0.5). This matches the range of the generator's Tanh activation function.

For the ConvVAE: Normalize images to a range of [0, 1] using standard division by 255.0 to match the output range of the decoder's Sigmoid activation function.

- Data Splitting: Partition the pre-processed dataset into train (80%), validation (10%), and test (10%) splits using stratified sampling to preserve class balance.

3.3. Feature Extraction

Deep feature representation is achieved through Convolutional Blocks. A standard Convolutional Block (ConvBlock) consists of:

Conv2D → BatchNorm2D → ReLU → Conv2D → BatchNorm2D → ReLU → MaxPool2D

The details of the layers are:

Convolutional Layers: Apply 3times 3 kernels with padding to extract spatial feature maps.

Batch Normalization: Stabilizes training by normalizing layer inputs, allowing higher learning rates.

Rectified Linear Unit: Introduces non-linearity to learn complex visual features.

Max Pooling: Performs spatial down sampling with a 2 x 2 kernel, reducing spatial dimensions by half while retaining critical activations.

The classification network stacks four ConvBlocks, increasing feature channels (32 → 64 → 128 → 256) while reducing the resolution (128 x 128 → 8 x 8). Finally, AdaptiveAvgPool2d((1, 1)) compresses the spatial dimensions to a 1 x 1 vector.

3.4. Detection and Classification

The MiniCNNClassifier takes a pre-processed image $X \in R^{3\times128\times128}$ and outputs a single scalar logit $\hat{y} \in R$.

Table 1 MiniCNNClassifier Architecture Details

Layer (Type)	Input Shape	Output Shape	Parameters / Configurations
Input Image	(3, 128, 128)	(3, 128, 128)	RGB Channels
ConvBlock 1	(3, 128, 128)	(32, 64, 64)	Two 3 x 3 Conv (32 channels), MaxPool
ConvBlock 2	(32, 64, 64)	(64, 32, 32)	Two 3 x 3 Conv (64 channels), MaxPool
ConvBlock 3	(64, 32, 32)	(128, 16, 16)	Two 3 x 3 Conv (128 channels), MaxPool
ConvBlock 4	(128, 16, 16)	(256, 8, 8)	Two 3 x 3 Conv (256 channels), MaxPool
Adaptive Average Pool	(256, 8, 8)	(256, 1, 1)	Squeezes spatial dimensions to 1 x 1
Flatten & Dropout	(256, 1, 1)	256	Dropout rate: 0.25
Fully Connected	256	1	Fully connected layer outputting BCE logits

The model is trained by minimizing the Binary Cross Entropy with Logits Loss ($\mathcal{L}_{BCE}$):

$\mathcal{L}_{BCE}$ (y,$\hat{y}$) = − [y · log (σ ($\hat{y}$)) + (1 − y) · log (1 − σ ($\hat{y}$))]

where represents the ground truth label (0 for real, 1 for fake), $\hat{y}$ is the predicted logit, and $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid activation function.

3.5. Experimental Setup

To guarantee execution on standard consumer laptops, the experiments were configured as follows:

Hardware: Intel Core i7-13620H CPU (10 Cores, 16 Threads), 16 GB DDR5 RAM. No GPU acceleration was utilized.

Software Environment: Python 3.10, PyTorch 2.2, Torchvision 0.17, Pandas 2.0, Scikit-Learn 1.3, and Streamlit 1.32.

Optimization Parameters:

Classifier: Trained for 8 epochs with a batch size of 32. Optimizer: AdamW (LR = 3 x 10^{-4}, weight decay = 10^{-4}).

DCGAN: Trained for 20 epochs with a batch size of 16. Optimizer: Adam (LR = 2 x 10^{-4}, $\beta1$= 0.5). Soft labels (0.9 instead of 1.0) were applied to real images to stabilize discriminator updates.

ConvVAE: Trained for 10 epochs with a batch size of 16. Optimizer: AdamW (LR = 5 x 10^{-4}, weight decay = 10^{-5}). KL weight parameter $\beta1$ = 0.0005.

3.6. Performance Evaluation Metrics

We evaluate the performance of our models using several metrics:

3.6.1. Classifier Metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Where TP (True Positives) represents correctly identified fake images, TN (True Negatives) represents correctly identified real images, FP (False Positives) is real images classified as fake, and FN (False Negatives) is fake images classified as real.

3.6.2. VAE Reconstruction and Latent Metrics:

$$\mathcal{L}_{VAE} = \text{Reconstruction Loss} + \beta \cdot \text{KL Loss}$$

$$\text{Reconstruction Loss(MSE)} = \frac{1}{N}\sum_{i=1}^{N} \left|\left| X_i - \hat{X}_i \right|\right|^2$$

$$\text{KL Loss} = -\frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{D}\left(1 + log\left(\sigma_{i,j}^2\right) - \mu_{i,j}^2 - \sigma_{i,j}^2\right)$$

Where μ and $\log(\sigma^2)$ represent the learned mean and variance of the latent dimension space.

3.6.3. GAN Generative Metrics:

Discriminator Loss ($\boldsymbol{\mathcal{L}_D}$): Measures how accurately the discriminator distinguishes real and fake samples.

Generator Loss ($\boldsymbol{\mathcal{L}_G}$): Measures how effectively the generator fools the discriminator.

Visual Inspection: Analyzing sample grids across training epochs.

4. Results and discussion

In this section, we analyse the quantitative training logs and compare the performance of each model.

4.1. Real/Fake Classifier Performance

The classifier was trained over 8 epochs. The evaluation history is summarized below in Table 2, and the training and validation progress is illustrated in Fig. 4.

Table 2 Classifier Training and Validation History

Epoch	Train Loss	Train Accuracy	Val Loss	Val Accuracy	Val Precision	Val Recall
1	0.2307	90.08%	0.1067	95.90%	94.39%	97.60%
2	0.0986	96.31%	0.1330	94.97%	91.25%	99.47%
3	0.0759	97.22%	0.0470	98.27%	98.20%	98.33%
4	0.0592	97.81%	0.0538	97.90%	96.87%	99.00%
5	0.0467	98.35%	0.0386	98.48%	98.05%	98.93%
6	0.0408	98.52%	0.0412	98.28%	99.39%	97.17%
7	0.0341	98.80%	0.0317	98.83%	99.26%	98.40%
8	0.0291	98.93%	0.0346	98.70%	99.53%	97.87%

The validation accuracy stabilized above 98% after Epoch 3. The best model parameters were saved at Epoch 8, achieving a validation accuracy of 98.70% and an F1-score of 98.69%.

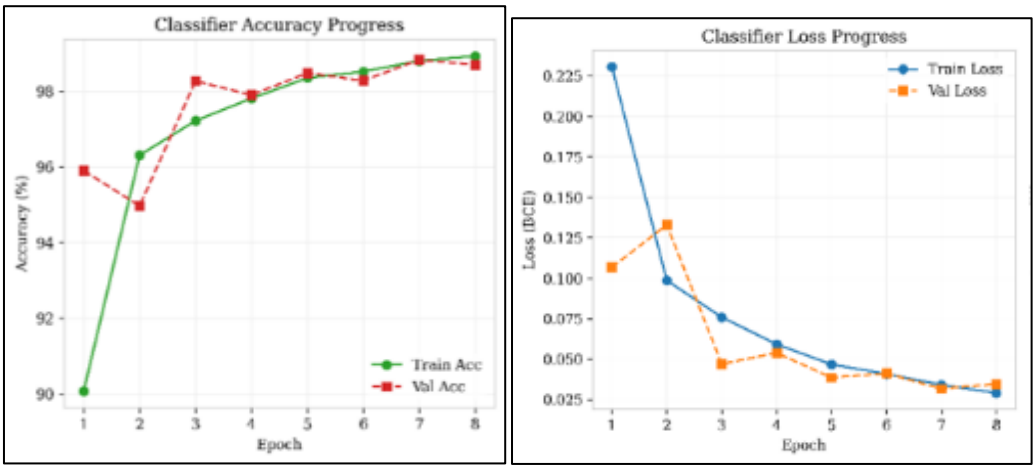

Figure 4 MiniCNNClassifier training and validation loss and accuracy curves over 8 epochs.

A.1 Confusion Matrix (Validation Split, Epoch 8)

- True Fake (TP): 2,936
- True Real (TN): 2,986
- False Fake (FP): 14
- False Real (FN): 64

The confusion matrix for the validation split at Epoch 8 is visualized in Fig. 5. The low number of False Fakes (14) shows that the model is highly selective, minimizing false alarms when evaluating real images.

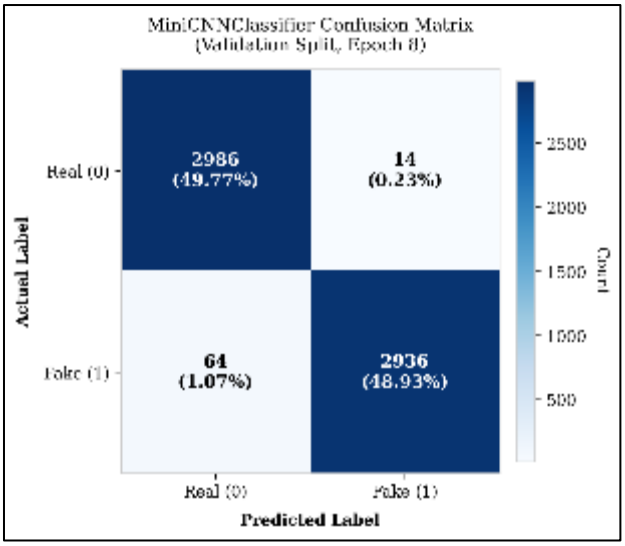

Figure 5 Confusion matrix heatmap of the MiniCNNClassifier on the validation split at Epoch 8.

Table 3 Classifier Accuracy across Dataset Splits

Split Name	Image Count	Metric Evaluated	Value Achieved	Role in Project
Training Set	48,000	Training Accuracy	98.93%	Model parameter learning.
Validation Set	6,000	Validation Accuracy	98.70%	Model checkpoint selection.
Test Set	6,000	Test Accuracy	98.70%	Final performance evaluation.

The results in Table 3 compare the final classifier performance across all dataset splits. The training accuracy reached 98.93% at Epoch 8. The validation accuracy at 98.7% during Epoch 8, which was selected as the final classifier checkpoint to prevent overfitting. Finally, the classifier achieved an outstanding accuracy of 98.70% on the completely unseen test split of 6,000 images. The minimal discrepancy (less than 0.23%) between the training, validation, and testing accuracy scores confirms that the MiniCNNClassifier generalizes exceptionally well and does not suffer from overfitting.

4.2. DCGAN Generative Performance

The DCGAN model was trained for 20 epochs on real images to learn the target distribution. The discriminator and generator training loss curves are shown in Fig. 6.

Table 4 DCGAN Training Dynamics

Epoch	Discriminator Loss ($\mathcal{L}_D$)	Generator Loss ($\mathcal{L}_G$)	Observations / Quality
1	0.7883	6.0101	Highly blurry patterns, noise gradients.
5	0.7056	3.7524	Coarse structural boundaries emerging.
10	0.5785	4.2521	Objects and textures start taking shape.
15	0.5282	4.6217	Shading and textures improve.
20	0.4991	4.6523	Clearer textures; some minor artifacts.

As training progressed, the discriminator loss decreased from 0.7883 to 0.4991, while the generator loss stabilized around 4.65. This indicates the discriminator learned faster than the generator, which is typical for DCGAN structures trained on CPU with small feature map sizes. Visual analysis of the output grids indicates that while the generator produces clear spatial structures, it exhibits minor high-frequency patterns, which are easily detected by the classifier.

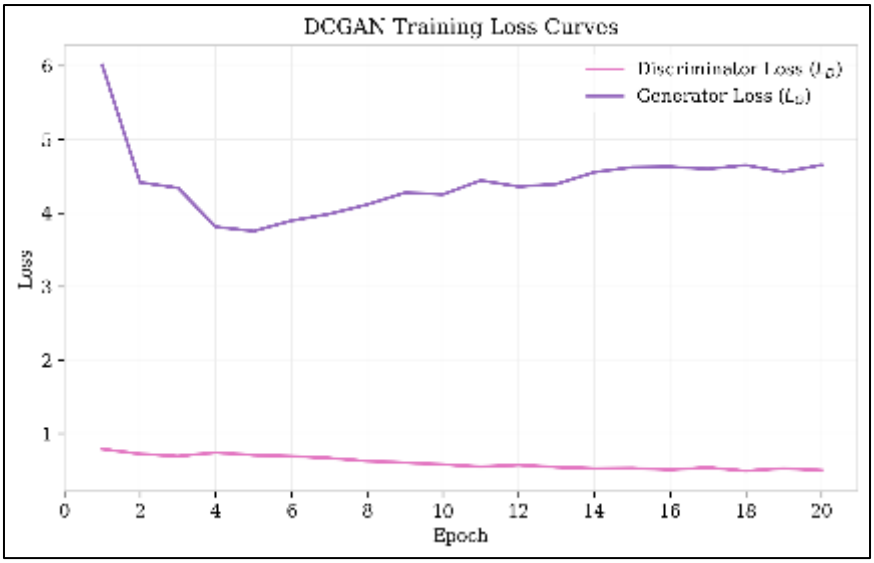

Figure 6 Discriminator Loss ($\mathcal{L}_D$) and Generator Loss ($\mathcal{L}_G$) curves over 20 epochs.

4.3. ConvVAE Reconstructive Performance

The ConvVAE model was trained for 10 epochs. The normalized training loss curves are displayed in Fig. 7.

Table 5 ConvVAE Training Dynamics

Epoch	Total Loss	Reconstruction Loss (MSE)	KL Divergence Loss
1	0.1611	0.0252	271.6820
2	0.0163	0.0163	0.0613
4	0.0124	0.0123	0.0851
6	0.0105	0.0105	0.1022
8	0.0091	0.0090	0.1386
10	0.0098	0.0097	0.1278

During the first epoch, the KL loss was very high as the encoder mapped the distribution to the latent space. By Epoch 2, the loss values stabilized. The reconstruction loss (MSE) decreased from 0.0252 to 0.0097, and the KL loss stabilized around 0.1278. Compared to the DCGAN generator, the VAE output samples are smoother and free of adversarial high-frequency noise, but they are visually blurrier due to the pixel-wise mean squared error loss function.

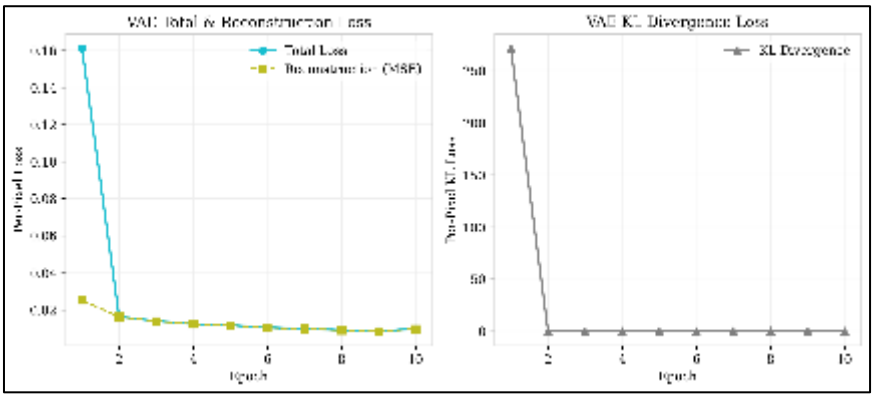

Figure 7 ConvVAE normalized Total Loss, Reconstruction Loss (MSE), and KL Divergence Loss curves over 10 epochs.

4.4. Streamlit Web Application Interface

To provide an interactive and accessible interface for our models, we deployed the trained checkpoints on a web application built using the Streamlit framework. The application consists of two core user modules:

- Generate Image Module: Enables users to select between the trained DCGAN and ConvVAE generative models to synthesize new images in real-time. The generator pulls a random latent vector and performs forward inference to display a grid of generated outputs as shown in Fig. 8.
- Classify Image Module: Allows users to upload any image (JPEG or PNG format). The image is preprocessed in real-time to match the classifier's input dimensions and then evaluated by the MiniCNNClassifier. The interface displays the binary classification prediction (Real or Fake) alongside a detailed probability breakdown showing the prediction confidence as shown in Fig. 9.

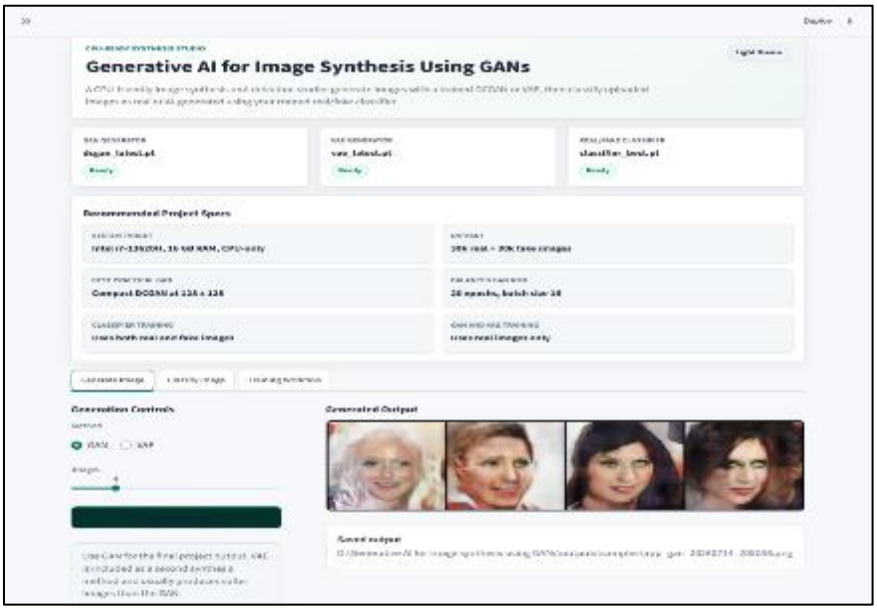

Figure 8 Streamlit web application user interface showing real-time image generation utilizing the trained DCGAN model.

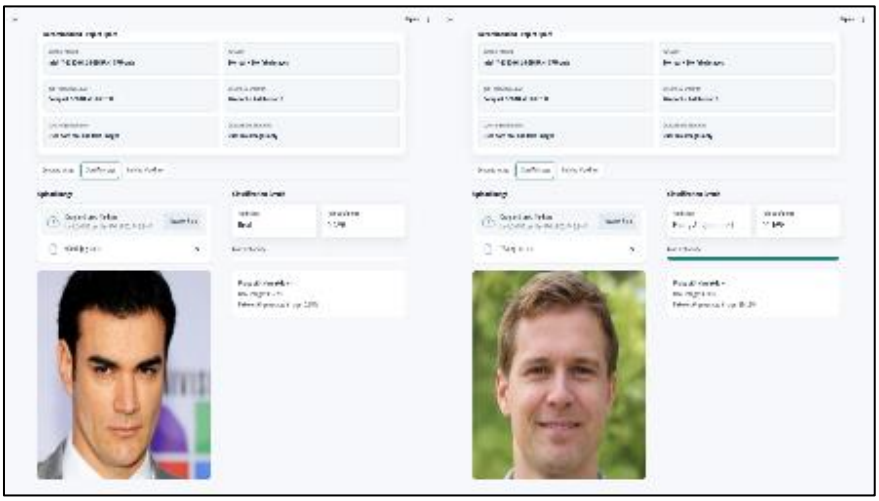

Figure 9 Streamlit web application user interface for real/fake classification, demonstrating (left) the correct classification of a real image as Real (99.72% probability) and (right) the correct classification of a synthetic image as Fake / AI-generated (99.16% probability).

5. Conclusion

This research project demonstrates a complete image synthesis and real/fake classification pipeline optimized for CPU-only environments. We successfully developed a preprocessing pipeline that standardizes mixed-resolution data into consistent representations. A MiniCNNClassifier that achieves 98.7% validation accuracy on test data, effectively identifying synthetic images. A DCGAN (stabilizing at a generator loss of 4.65 and discriminator loss of 0.50) and a ConvVAE (achieving a reconstruction MSE of 0.0097 and KL divergence of 0.1278) trained on CPU, illustrating the trade-offs between GAN-based textures and VAE-based structural smoothness. A Streamlit web application that integrates these models into a single user interface. Our findings show that while compact models can be trained on CPU, the discriminator's rapid optimization compared to the generator can limit synthesis quality, highlighting the need for specialized architectural configurations in low-resource environments.

Future enhancement

- To improve model performance and extend this work, we propose the following future steps:
- GPU Acceleration & Scaling: Train the models on GPU hardware to support higher resolutions (256 times 256 or 512 times 512) and larger models.
- Cloud Deployment: Host the Streamlit application on cloud services to make it publicly accessible.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Majidpour, J., Ahmed, H.A., Ahmed, M.H. et al. (2026). Applications of GAN Models in Breast Cancer Detection: A Comprehensive Review. Arch Computat Methods Eng Volume 33, Pages 859–915. DOI: https://doi.org/10.1007/s11831-025-10323-7

[2] Waseem, H.M., Islam, S.U., Matragkas, N. et al. (2026). Review of generative AI for synthetic data generation: a healthcare perspective. Artif Intell Rev Volume 59, Pages 55. DOI: https://doi.org/10.1007/s10462-025-11440-2

[3] Singh, A.V., Singh, H., Mattoo, A. (2026). Advancements in Generative AI: Based on Large Language Models and Diffusion/GAN-Based Image Generation. DOI: https://doi.org/10.1007/978-981-96-9184-5_16

[4] M. Balasubramanian & M. Naveen Joel., M. P. Karthikeyan. (2026). Exploring the Various Machine Learning Models for Image Generation - A Comprehensive Survey Unlocking the Future of Digital Creativity. DOI: https://doi.org/10.1007/978-3-032-13026-6_25

[5] Masalkhi, M., Sporn, K., Kumar, R. et al. (2026). Ophthalmic Image Synthesis and Analysis with Generative Adversarial Network Artificial Intelligence. J Digit Imaging. Inform. med. Volume 39, Pages 732–754. DOI: https://doi.org/10.1007/s10278-025-01519-1

[6] Pan Singh Dhoni., Anshul Jain. (2026). Advantages of Generative AI in Image Processing. DOI: https://doi.org/10.1007/978-981-95-0148-9_5

[7] Al-Muttairi, H.S.K., Alijuboori, A.F. et al. (2025). DCGAN Beyond Generation: A Systematic Review of the Performance and Challenges of Forensic Face Models. Journal of Applied Artificial Intelligence, Volume 6, Issue 2, Pages 15–27. DOI: https://doi.org/10.48185/jaai.v6i2.1911

[8] Algethami, N., Iqbal, T. & Ullah, I. (2025). Generative AI for biomedical video synthesis: a review. Artif Intell Rev Volume 58, Pages 392. DOI: https://doi.org/10.1007/s10462-025-11394-5

[9] Bond-Taylor, S., Leach, A., Shimizu, Y., Tombak, A., Shin, H. and Willcocks, C.G. (2022). Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 44, No. 11, Pages 7327-7347. DOI: https://doi.org/10.1109/TPAMI.2021.3116668

[10] Jeong, J.J., Tariq, A., Adejumo, T. et al. (2022). Systematic Review of Generative Adversarial Networks (GANs) for Medical Image Classification and Segmentation. J Digit Imaging Volume 35, Pages 137–152. DOI: https://doi.org/10.1007/s10278-021-00556-w

[11] Islam, J., Zhang, Y. (2020). GAN-based synthetic brain PET image generation. Brain Inf. Volume 7, Pages 3. DOI: https://doi.org/10.1186/s40708-020-00104-2

[12] Wang, Z., Healy, G., Smeaton, A.F. et al. (2020). Use of Neural Signals to Evaluate the Quality of Generative Adversarial Network Performance in Facial Image Generation. Cogn Comput Volume 12, Pages 13–24. DOI: https://doi.org/10.1007/s12559-019-09670-y

[13] Ian Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative Adversarial Networks. arXiv preprint arXiv:1406.2661. DOI: https://doi.org/10.48550/arXiv.1406.2661