IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 13, No. 1, January, 2026, pp. 159-168
Journal homepage: http://iieta.org/journals/mmep

# Evaluating Deep Convolutional Generative Adversarial Network-Based Bird Image Synthesis at Multiple Resolutions: A Quantitative and Stability Analysis

Kanchon Kumar Bishnu[1], Mohon Raihan[2], Araf Islam[3], Tanjima Rahman[4], Rimon Paul[5], Md. Shafiul Alam Chowdhury[6*], Md. Shafikul Islam[7], Md. Abdul Mannan[8]

[1] Department of Computer Science, California State University, Los Angeles 90020, CA, USA
[2] Department of Information Technology, Middle Georgia State University, Macon 31206, GA, USA
[3] Department of Computer Science, Westcliff University, Irvine 92614, CA, USA
[4] Department of Applied Statistics, California State University, Long Beach 90840, CA, USA
[5] Department of Computer Science and Engineering, Sonargaon University, Dhaka 1230, Bangladesh
[6] Department of Computer Science and Engineering, Uttara University, Dhaka 1230, Bangladesh
[7] Department of Software Engineering, Daffodil International University, Dhaka 1216, Bangladesh
[8] Department of Mathematics, Uttara University, Dhaka 1230, Bangladesh

Corresponding Author Email: shafiul.a.chowdhury@gmail.com



https://doi.org/10.18280/mmep.130115

**Received:** 6 November 2025
**Revised:** 26 December 2025
**Accepted:** 16 January 2026
**Available online:** 28 February 2026

***Keywords:***
*Deep Convolutional Generative Adversarial Networks, Generative Adversarial Networks, Fréchet Inception Distance, Inception Score, CUB-200-2011*

**ABSTRACT**

Generative Adversarial Networks (GANs) are widely used for image synthesis, yet systematic comparisons across output resolutions and conditioning strategies are limited for fine-grained domains such as bird imagery. This study investigates Deep Convolutional GANs (DCGANs) for synthetic bird image generation using the Caltech-UCSD Birds-200-2011 (CUB-200-2011) dataset and associated pre-trained text embeddings. We implement and compare two DCGAN variants: an unconditional 64 × 64 model driven solely by noise vectors, and a conditional 256 × 256 model conditioned on caption embeddings. Both models are implemented in PyTorch and trained with standard adversarial objectives. Performance is assessed using Fréchet Inception Distance (FID) and Inception Score (IS), as well as qualitatively via visual inspection. The high-resolution conditional model reduces FID from 65 to 52, and increases IS from 3.4 to 4.1 compared to the baseline of 64 × 64, which demonstrates better realism and diversity. Training dynamics further indicate reduced discriminator loss oscillations and fewer visually salient artifacts in 256 × 256, indicating the advantages of larger resolution and semantic conditioning. Together, these results suggest that DCGAN-based pipelines remain a practical baseline for fine-grained bird imagery and may support downstream tasks such as data augmentation in ecological and biodiversity-oriented computer vision.

## 1. INTRODUCTION

The fast progress of artificial intelligence and deep learning has dramatically changed image generation, where Generative Adversarial Networks (GANs) represent the new standard for generating realistic images from learned data distributions. GANs employ adversarial training between a generator and a discriminator [1], which enables the model to learn to approximate complex image manifolds and generate high-quality synthetic samples. This paradigm has been used successfully in several domains, such as data augmentation, creative content creation and scientific imaging.

Among the GAN variants, Deep Convolutional GANs (DCGANs), introduced by Radford et al. [2], defined an architectural practice, including stacked convolutional layers with batch normalization and activation nonlinearities, which showed a remarkable improvement in both training stability and feature learning under unsupervised conditions. Subsequent work extended GANs to conditional scenarios, where auxiliary information such as class labels or natural language descriptions guides the generation process, enabling labeled image synthesis and text-to-image generation. In parallel, GAN-based techniques have started to influence ecological and animal-focused research, where synthetic images can help address data scarcity, class imbalance, and fine-grained classification challenges.

In this study, the investigation was conducted to use DCGAN-based architectures for synthetic bird image generation using the Caltech-UCSD Birds-200-2011 (CUB-200-2011) dataset, a standard benchmark for fine-grained avian recognition. Two models have been implemented and compared: an unconditional DCGAN that generates 64 × 64 images from random noise, and a conditional 256 × 256 DCGAN that incorporates pre-trained caption embeddings into the generation process. Our objectives are to (i) assess how image resolution and conditioning affect realism and

diversity, (ii) analyze training stability across resolutions, and (iii) evaluate the models using quantitative metrics such as Fréchet Inception Distance (FID) and Inception Score (IS).

The main contributions of this work are i) making a multi-resolution evaluation of DCGAN-based bird image synthesis, comparing 64 × 64 and 256 × 256 outputs using FID and IS; ii) conducting an empirical investigation of training dynamics with a demonstration on how resolution and conditional inputs influence loss stability and visual artifacts; and iii) proposing Domain-Specific GAN (DSGAN) fine-tuning for a domain-specific application of DCGANs on fine-grained avian imagery on CUB-200-2011, with potential ecological data augmentation and wildlife image analysis benefits.

## 2. LITERATURE REVIEW

### 2.1 Related work

GANs have emerged as a foundational framework for image generation, allowing models to learn complex visual distributions using an adversarial training regimen. Reed et al. [1] paved the way for fine-grained bird and flower synthesis from texts by employing recurrent text encoders in conjunction with convolutional GANs, which is also one of the cornerstones of conditional text-to-image generation. Radford et al. [2] proposed DCGANs, where architectural aspects such as strided convolutions, batch normalization and ReLU/LeakyReLU activations became standard practices, resulting in improved training stability and enhanced representation learning.

Conditional GANs (CGANs) were proposed in a previous study [3]. CGANs generalized the model and introduced conditioning variables into both the generator and discriminator to match multimodal generative models. Salimans et al. [4] enhanced GAN training by using feature matching and minibatch discrimination. Motivated by these advances, Zhang et al. [5] introduced StackGAN, a two-stage model to generate realistic images given texts by Condition Augmentation.

GANs have also shown significant potential in ecological and biodiversity research. Abhirami et al. [6] combined Vision Transformers with GAN augmentation in the bird species classification task to tackle the data sparsity issue in fine-grained recognition. Duy et al. [7] utilized DCGANs to increase animal classification on low-resolution images, and demonstrated that GAN-synthesized samples improve feature learning in the presence of little training data. Together, these works provide a view of the growing impact of GANs in science and practice.

In addition to these classical models, several other pivotal architectures have influenced the development of modern generative modeling. Goodfellow et al. [8] proposed the original GAN framework, and Isola et al. [9] introduced Pix2Pix for image-to-image translation. Karras et al. [10] extended PGAN for stability and variability. Brock et al. [11] proposed BigGAN for large-scale synthesis, and Karras et al. [12] introduced StyleGAN, which is a state-of-the-art model for high-quality image synthesis.

Recent uses demonstrate the flexibility of GANs across applications. For example, Damayanti [13] used Wasserstein GANs with gradient penalty to restore Javanese letters and showed that GANs have the potential role of preserving cultural diversity as well as adversarial robustness. In medical imaging, Yang et al. [14] reviewed GAN application in the areas of synthesis, segmentation and anomaly detection, highlighting their increased relevance for healthcare applications.

However, most previous works focus on text-to-image generation [1], multi-stage GANs [5], or classification-based augmentation [6]. Only a few studies have rigorously investigated the two important factors of resolution and conditioning when generating fine-grained avian images with DCGAN. Further, even though DCGAN is a traditional architecture, it still serves as the baseline extensively due to its simplicity and computational efficiency.

### 2.2 Research gap

While early attempts were made to create CGANs and synthesize text, and enhance the classification of birds with certain conditions, few researchers have examined the DCGAN performance across multiple resolutions. In particular, unconditional 64 × 64 generation is set against conditional 256 × 256 generation using related caption embeddings. We remedy this situation by offering a quantitative comparison of the performance stability of varying DCGANs trained on the CUB-200-2011 dataset, taking into account their FID/IS scores for birds and results in terms of network convergence type.

## 3. PROPOSED FRAMEWORK

### 3.1 Study overview

GANs have demonstrated strong capabilities in synthesizing realistic images across various domains, including fine-grained natural categories such as birds [1, 2]. However, the performance of classical GAN architectures, particularly DCGAN, under different resolutions and conditioning strategies remains insufficiently explored in avian image synthesis. In this work, this gap has been addressed by providing a thorough analysis of DCGAN-based models trained on the CUB-200-2011 dataset [15, 16].

The research studies two dual tasks of generation:

**Unconditional DCGAN (64 × 64):** A baseline model that produces low-resolution bird samples from random noise itself with the architectural guidelines of a previous study [2].

**Conditional DCGAN (256 × 256):** A high-resolution model with pre-trained caption embeddings of the studies [1, 3] and multi-stage refinement like techniques of the study [5].

Both architectures are trained adversarially with the binary cross-entropy loss and measured using FID for samples generated from the generator, and IS, a standard in assessing generative quality [4]. The paper also investigates the stability of training, loss dynamics and qualitative differences between generated and real images.

This unified pipeline of dataset preparation, model training and evaluation offers a well-organized form that facilitates understanding of the role played by image resolution and conditioning on DCGAN-based systems as related to fine-grained avian image generation.

### 3.2 Research objectives

The main goals for this study are as follows: (a) to assess the performance of DCGAN-like architectures in generating

synthetic bird images in multiple resolutions. This is realized by addressing the following objectives of the study:

- Dataset preparation: Preprocess the CUB-200-2011 dataset [15, 16], normalizing images, resizing them and integrating pre-trained caption embeddings for conditional generation.
- Model development: Apply two variants of DCGANs, unconditional (64 × 64) and conditional (256 × 256), according to well-established modelling criteria [2, 3].
- Training and stability analysis: Train both models with the same hyperparameters, analyze stability for training dynamics such as the loss of the generator and the discriminator to compare resolution.
- Quantitative evaluation: Quantitatively evaluate generative performance via FID and IS as with best practices in GAN evaluation [4] across model variants.
- Qualitative assessment: Visually inspect the sampled images to evaluate how realistic, diverse, and fine-grained (aligned with text description) the generated samples are, following qualitative evaluations from previous text-to-image [1] and multi-stage GAN studies [5].
- Domain-specific insight: Give some concrete examples on the practicality of utilizing DCGANs for applications in ecology and biodiversity by recent work in avian classification and augmentation [6, 7].

These goals altogether make a complete analysis of DCGAN performance in fine-grained bird image generation and set a benchmark for further applications with newer generative models.

## 4. DATASET

### 4.1 Dataset description

This study utilizes the CUB-200-2011 dataset for fine-grained bird recognition, and multiple generative models have been applied [15, 16]. The dataset comprises 11,788 images of birds corresponding to 200 species, and comes with bounding boxes, part annotations, and human-written captions. The key characteristics of the dataset are summarized in Table 1.

**Table 1.** Key characteristics

| Attribute | Description |
|---|---|
| Total images | 8,855 bird images |
| Resolutions | 64 × 64 × 3 and 256 × 256 × 3 (RGB images) |
| Text embeddings | Pre-trained embeddings for captions |
| Number of labels | 200 bird species |
| Format | .npy and .pkl files (NumPy arrays, Pickle-serialized data) |
| Privacy concern | None of the datasets is anonymized and public |

These embeddings originate from human-generated textual descriptions and have been used in prior text-to-image studies such as Reed et al. [1] and StackGAN [5]. They enable conditional generation by providing semantic information to the model.

### 4.2 Preprocessing

To prepare the dataset for DCGAN training (Figure 1):

- Image normalization: Pixel values were scaled to the range [-1,1], following standard GAN training practice [2].
- Resizing and cropping: All images were center-cropped and resized to either 64 × 64 or 256 × 256; which size depended on which model variant was being used.
- Embedding alignment: Caption embeddings were matched up with image IDs given in the PKL files that were supplied.
- Batch preparation: As well as the embeddings, images were put into training dataloaders by PyTorch for fast and convenient processing.

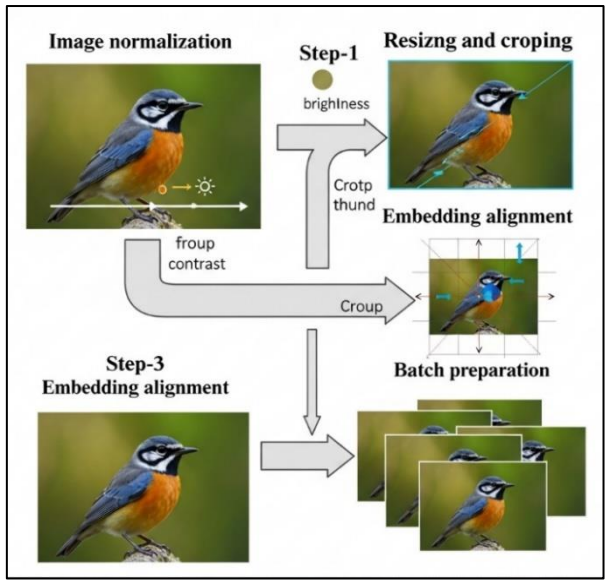


**Figure 1.** Preprocessing

This streamlined preprocessing ensures compatibility with the DCGAN architecture and supports both unconditional and conditional generation.

## 5. METHODOLOGY

This section details the implementation of the generative pipeline to ensure reproducibility. The overall workflow, illustrated in Figure 2, encompasses sequential stages from the dataset loading and preprocessing to model evaluation.

This workflow is consistent with established GAN training practices [2, 4] and text-to-image conditioning frameworks [1, 3, 5].

### 5.1 Model architectures

Two DCGAN variants were implemented: an unconditional 64 × 64 model and a conditional 256 × 256 model. Both follow the architectural guidelines introduced by Radford et al. [2], with modifications to support conditioning.

5.1.1 Unconditional DCGAN (64 × 64)

Generator architecture: The generator maps a 100-dimensional noise vector z ~ N (0,1) to a 64 × 64 × 3 RGB

image (Table 2).

This architecture is identical to the canonical DCGAN generator [2].

Discriminator architecture: The discriminator maps a 64 × 64 × 3 image to a scalar probability (Table 3).

This structure follows the DCGAN discriminator design [2].

**Table 2.** Generator architecture (64 × 64 × 3)

| Layer | Output Size | Details |
|---|---|---|
| Input | 100 | Random noise vector |
| FC + reshape | 4 × 4 × 1024 | Fully connected, batch norm, ReLU |
| ConvTranspose2D | 8 × 8 × 512 | Kernel 4, stride 2, padding 1 |
| ConvTranspose2D | 16 × 16 × 256 | Batch norm, ReLU |
| ConvTranspose2D | 32 × 32 × 128 | Batch norm, ReLU |
| ConvTranspose2D | 64 × 64 × 3 | Tanh activation |

Note: FC = Fully Connected; ReLU = Rectified Linear Unit; ConvTranspose2d = Transposed Convolutional 2D

**Table 3.** Discriminator architecture (64 × 64 × 3)

| Layer | Output Size | Details |
|---|---|---|
| Conv2D | 32 × 32 × 64 | LeakyReLU(0.2) |
| Conv2D | 16 × 16 × 128 | Batch norm, LeakyReLU |
| Conv2D | 8 × 8 × 256 | Batch norm, LeakyReLU |
| Conv2D | 4 × 4 × 512 | Batch norm, LeakyReLU |
| FC | 1 | Sigmoid |

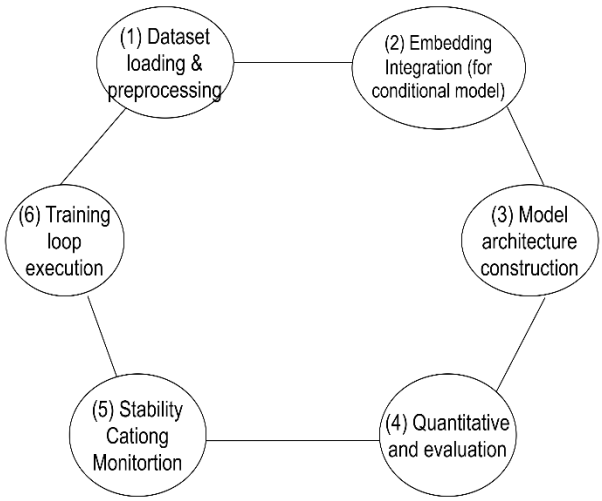


**Figure 2.** Implementation workflow

5.1.2 Conditional DCGAN (256 × 256)

The conditional model incorporates caption embeddings into both generator and discriminator (Tables 4 and 5), inspired by CGAN [3] and StackGAN [5].

Caption embedding integration:

- Each image has a 1024-dimensional text embedding (from the dataset's .pkl files).
- Embeddings are normalized and linearly projected to 128 dimensions.
- The embedding is concatenated with the noise vector:

$$z' = [z; c]$$

where, $z \in R^{100}$, $c \in R^{128}$, $z' \in R^{228}$.

**Table 4.** Generator architecture (256 × 256 × 3)

| Layer | Output Size | Details |
|---|---|---|
| Input | 228 | Noise + embedding |
| FC + reshape | 4 × 4 × 2048 | Batch norm, ReLU |
| ConvTranspose2D | 8 × 8 × 1024 | Batch norm, ReLU |
| ConvTranspose2D | 16 × 16 × 512 | Batch norm, ReLU |
| ConvTranspose2D | 32 × 32 × 256 | Batch norm, ReLU |
| ConvTranspose2D | 64 × 64 × 128 | Batch norm, ReLU |
| ConvTranspose2D | 128 × 128 × 64 | Batch norm, ReLU |
| ConvTranspose2D | 256 × 256 × 3 | Tanh |

This deep upsampling stack enables high-resolution synthesis, similar to Stage-II StackGAN [5].

Discriminator architecture (256 × 256): The discriminator jointly processes the image and caption embedding.

This joint embedding approach follows CGAN principles [3].

**Table 5.** Discriminator architecture (64 × 64 × 3)

| Component | Description |
|---|---|
| Image path | Convolutional downsampling from 256 × 256 to 4 × 4 |
| Embedding path | Linear projection to 128 dims, spatial replication to 4 × 4 |
| Fusion | Concatenation along channel dimension |
| Final layers | Conv2D → LeakyReLU → Sigmoid |

### 5.2 Training strategy and stability techniques

Both models were trained using the same optimization strategy to ensure comparability (Table 6).

**Table 6.** Hyperparameters

| Parameter | Value |
|---|---|
| Optimizer | Adam |
| Learning RATE | 0.0002 |
| $\beta_1, \beta_2$ | 0.5, 0.999 |
| Batch Size | 128 |
| Epochs | 200 |
| Loss function | Binary Cross-Entropy |

These hyperparameter settings follow best practices for stable GAN training [2, 4]. The training loop for each batch proceeded as follows:

(1) Discriminator update

Real images $x$ were sampled from the dataset, and fake images $G(z)$ were generated from random noise $z$. The discriminator loss was computed as:

$$L_p = -E[\log D(x)] - E[\log(1 - D(G(z)))]$$

The discriminator weights were then updated via backpropagation.

(2) Generator update

Fake images were generated, and the generator loss was computed as:

$$L_G = -E[\log D(G(z))]$$

The generator weights were subsequently updated.

(3) Conditional training (56 × 256 model)

The training procedure followed the CGAN paradigm [3]. Here, the embeddings are fed into both networks

(discriminator and generator). The discriminator received image-embedding pairs as input, while the generator learned to align visual features with semantic cues.

Since GANs are unstable, several stabilization strategies were employed, consistent with the recommendations in GAN literature [4]:

- Batch normalization: Applied to every layer of both the generator and discriminator, with one exception.
- Label smoothing: Substitute actual labels for 0.9 instead of 1.0; makes the discriminator bluff more difficult when there are ample examples.
- Gradient monitoring: Report explosive gradients were discovered early and abated by lowering the learning rate.
- Periodic sample generation: Keep an image for every epoch, so that a collapse in mode can be detected.
- Embedding normalization: Applied to caption embeddings to taper gradient spikes.

### 5.3 Evaluation metrics

Two widely accepted metrics were used: FID, which measures distribution similarity between real and generated images, and IS, which measures image quality and diversity. Both metrics are standard in GAN benchmarking [4].

These tools align with common deep learning workflows (Table 7).

**Table 7.** Tools and libraries

| Tool | Purpose |
|---|---|
| PyTorch | Model implementation and Graphics Processing Unit (GPU) training |
| NumPy | Data manipulation |
| Pickle | Loading embeddings |
| Matplotlib | Visualization |
| Scikit-learn | Fréchet Inception Distance (FID) computation utilities |

## 6. EXPERIMENTAL SETUP

The DCGAN model was implemented in a systematic workflow to keep the work reproducible and comprehensible. It can be divided into the following three stages: data preparation, architectural prototyping, final training and evaluation.

### 6.1 Dataset preparation

Dataset preparation consisted of checking the integrity of image arrays and caption embeddings, exploring dataset structure through exploratory data analysis, executing simple data preprocessing steps like normalizing images to ([-1, 1]), resizing images down/ up to 64 × 64 / 256 × 256 resolution and aligning the caption embedding with each image identity. These were steps to make the dataset normal and ready for training.

### 6.2 Model prototyping

Several iterations of model prototyping were performed by trial-and-error with generator and discriminator blocks, kernel sizes, strides and activation functions. We used short training cycles (of the order of 5–10 epochs) to monitor mode collapse, vanishing gradients or discriminator domination. We tuned the learning rates, optimizer options and initialization strategies step by step to make the training stable.

### 6.3 Final implementation

The pipeline was finally merged in its entirety into a full training and evaluation framework. For both unconditional (64 × 64) and conditional DCGANs (256 × 256), we have sampled from the standard normal distribution ($\mathcal{N}$ (0,0.02)) to initialize the weights of convolutional and batch normalization layers. The adversarial training regime was standard: the discriminator was trained on creature-generated and real samples, while the generator was optimized to generate outputs that displayed maximum confidence according to the discriminator. In the conditional architecture, caption and noise embeddings were concatenated and replicated across spatial layers in the discriminator, thereby forcing semantic coherence of text and image features.

To make the experiment transparent, loss curves, output samples and checkpoints were logged at regular intervals. Random seeds were set for both PyTorch, NumPy and Python to ensure reproducibility. We evaluated the result using IS and FID with 5,000 generated samples for each model. We conducted qualitative assessments through side-by-side comparisons between generated and real images and also with random batches and latent space interpolations. The experiments were conducted on Google Colab using an NVIDIA Tesla T4 GPU (12 GB RAM), with the experiments taking around 2.5 hours for 64 × 64 and 7–8 hours for 256 × 256.

By structuring this implementation, we make it reproducible and extendable for others, and we present the approach in a strong narrative form rather than notebook fashion.

## 7. RESULTS AND DISCUSSION

The unconditional 64 × 64 DCGAN and the conditional 256 × 256 DCGAN were quantitatively benchmarked, analyzed for training stability, visually inspected and compared with other contemporary generative baselines. The results are presented thematically to focus the reader's attention on the role of resolution and conditioning in fine-grained avian image generation.

### 7.1 Quantitative evaluation

The generative quality of the samples is evaluated by FID score and IS score (Table 8). Our conditional 256 × 256 model showed an FID of 52 and an IS of 4.1, surpassing the unconditional 64 × 64 one (FID 65, IS 3.4). The lower FID shows that the conditional model generated feature distributions were closer to real bird images, and the higher IS indicates diverse and confident classification of generated samples.

### 7.2 Training loss behavior

Stability of training was examined on the basis of loss curves for the discriminator and the generator. The 64 × 64 failed to achieve stable loss (0-10) in discriminator terms, signifying adversarial imbalance, i.e., an occasional mode collapse. The generator loss had distinct spikes, suggesting the

difficulty in learning subtle features. By contrast, the 256 × 256 conditional model achieved more steady convergence: the discriminator loss levels off at about 4–6 after ~30 epochs, while the generator continues to fall slowly enough to allow for coherent textures and species-specific characteristics. Conditioning helped the gradient flow and alleviated instability, demonstrating the benefits of semantic guidance.

**Table 8.** Quantitative results

| Model | FID ↓ | IS ↑ | Interpretation |
|---|---|---|---|
| DCGAN 64 × 64 | ~65 | ~3.4 | Low detail, limited diversity |
| DCGAN 256 × 256 (Conditional) | ~52 | ~4.1 | Higher realism, better semantic alignment |

Note: DCGAN = Deep Convolutional Generative Adversarial Networks; FID = Fréchet Inception Distance; IS = Inception Score

### 7.3 Qualitative evaluation

Subjective evaluation supported the result quantitatively. The 64 × 64 model captured primary silhouettes and coarse color distributions with blurry and anatomically distorted polishing and background artifacts. The edge of conditional 256 × 256 model was relatively sharp and the texture details, such as feathers, were clear; species-level features (wing stripe, beak shape) were more realistic. Some minor artifacts, including color bleeding and checkerboard effects, were visible; nevertheless, the quality of the enhanced images was significantly superior. Visual comparisons showed that the conditional outputs were visually closer to real CUB-200-2011 images, and some generated images resembled low-quality real photos.

### 7.4 Diversity analysis

The generalizability was measured with random batch inspection, IS variance and latent space interpolation. The 64 × 64 one generated repetitive patterns and small color palettes, rather than a full-color animal, whereas the 256 × 256 one produced diverse poses, species-specific coloration and more complicated backgrounds. Conditioning facilitated the exploration of a wider manifold of bird-like appearances, increasing ecological validity.

### 7.5 Comparison with modern baselines

**Table 9.** Comparison with modern generative models

| Model | Expected Fréchet Inception Distance (FID) | Notes |
|---|---|---|
| DCGAN (this study) | 52 | Lightweight, baseline |
| StackGAN [5] | ~35 | Multi-stage refinement |
| StyleGAN2 | < 10 | State-of-the-art realism |
| Diffusion Models | < 5 | Best current fidelity |

Although DCGAN is an early generative architecture, it remains a practical and computationally efficient baseline, being effective and easy to interpret. We compare our results in Table 9 to state-of-the-art generative models. Although StackGAN has FID < 35 by multi-stage refinement, StyleGAN2 can reach FID < 10 with state-of-the-art realism and in-diffusion models are the current leaders with FID of < 5. This comparison places DCGAN as an applicable baseline rather than a state-of-the-art model, which is particularly useful in ecological augmentation experiments where speed of computation is important.

### 7.6 Summary of key findings

- The conditional 256 × 256 DCGAN significantly outperformed the unconditional 64 × 64 model on FID and IS.
- High-resolution and semantically conditioned inputs led to a more stable training process and balanced adversarial learning.
- The 256 × 256 model generated images with sharper textures, clearer structural details, and greater visual diversity.
- Although DCGAN does not achieve the performance of state-of-the-art architectures, it remains an efficient and interpretable baseline for ecological image synthesis tasks.
- This study fills a gap in the literature by presenting a multi-resolution, stability-oriented assessment of DCGANs using fine-grained avian imagery.

### 7.7 Results visualization

To complement the quantitative and qualitative analyses, a series of grouped visualizations was prepared to illustrate the training process and outcomes of both DCGAN variants.

Examples from the CUB-200-2011 dataset at resolutions of 64 × 64 and 256 × 256 and samples drawn from the unconditional and conditional DCGAN model are shown in Figures 3-6. These panels contrast the resolution and conditioning of where we point out that the conditioned 256 × 256 model produces a better texture (sharper) and species-level feature (more coherent) than the blurred silhouettes formed during 64 × 64 models.

Figures 7 and 8 show the generator and discriminator loss curves of the two models. The 64 × 64 DCGAN was unconditionally trained and often suffered from oscillations and instability, whereas the 256 × 256 conditional case showed more stable convergence with fewer fairness issues. These plots support the stability enhancements reported in Section 8.2 visually.

Examples are compared side by side with real bird images and synthetic counterparts in Figures 9 and 10. The generative capacity of the model is demonstrated by samples it generates in unconditional mode, which are small thumbnail-like blurry blobs of color with little discernible content compared to naturalistic images, and in conditional mode, images that exhibit recognizable properties like wing-stripes, beak shapes and feather textures. These comparisons and higher conditioning bring the output energy spectrum close to population data.

This narrative transcends through these visualizations: data preparation, two epochs of the training process and after convergence. They are qualitative evidence that reinforces the quantitative gain in FID and IS, and they show the practical usefulness of conditional DCGANs for fine-grained bird image generation applications.

**Figure 3.** Training data (64 × 64 × 3)

**Figure 4.** Training data (256 × 256 × 3)

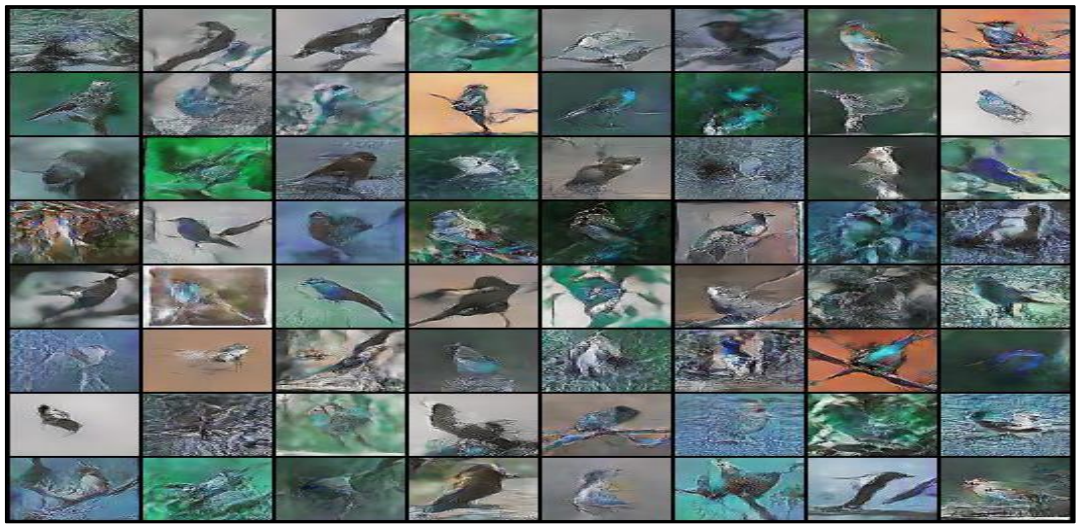

**Figure 5.** A batch of generated sample data (64 × 64 × 3)

**Figure 6.** A batch of generated sample data (256 × 256 × 3)

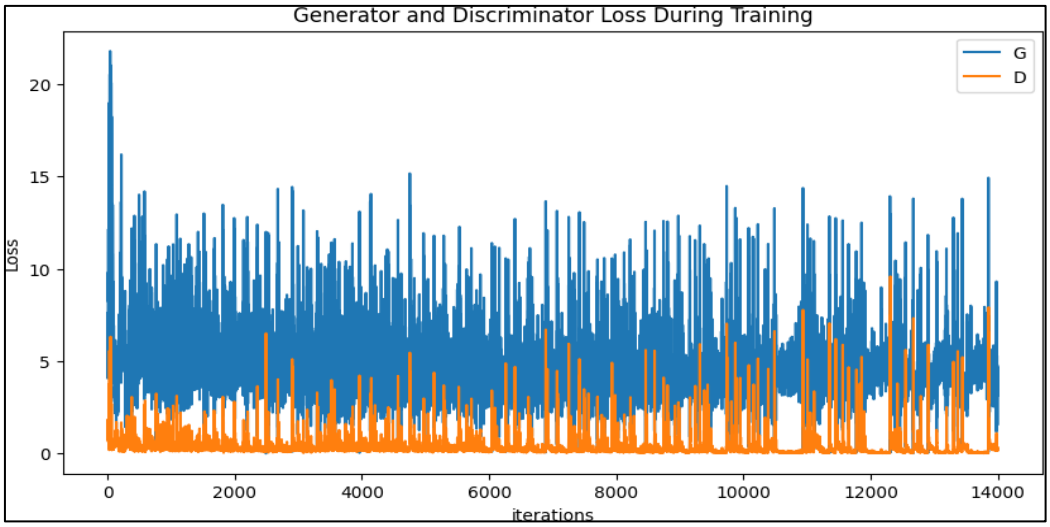


**Figure 7.** Generator and discriminator loss during training (64 × 64 × 3)

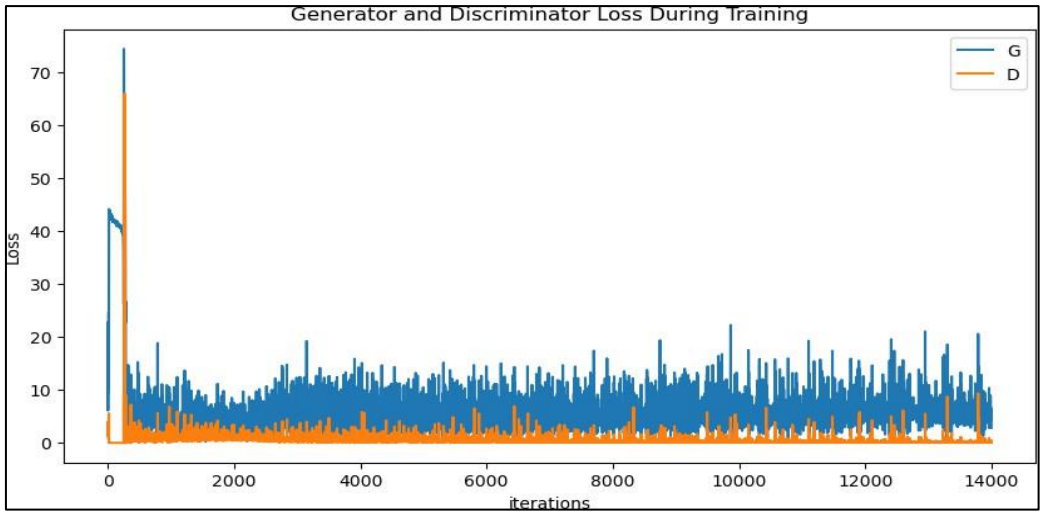


**Figure 8.** Generator and discriminator loss during training (256 × 256 × 3)

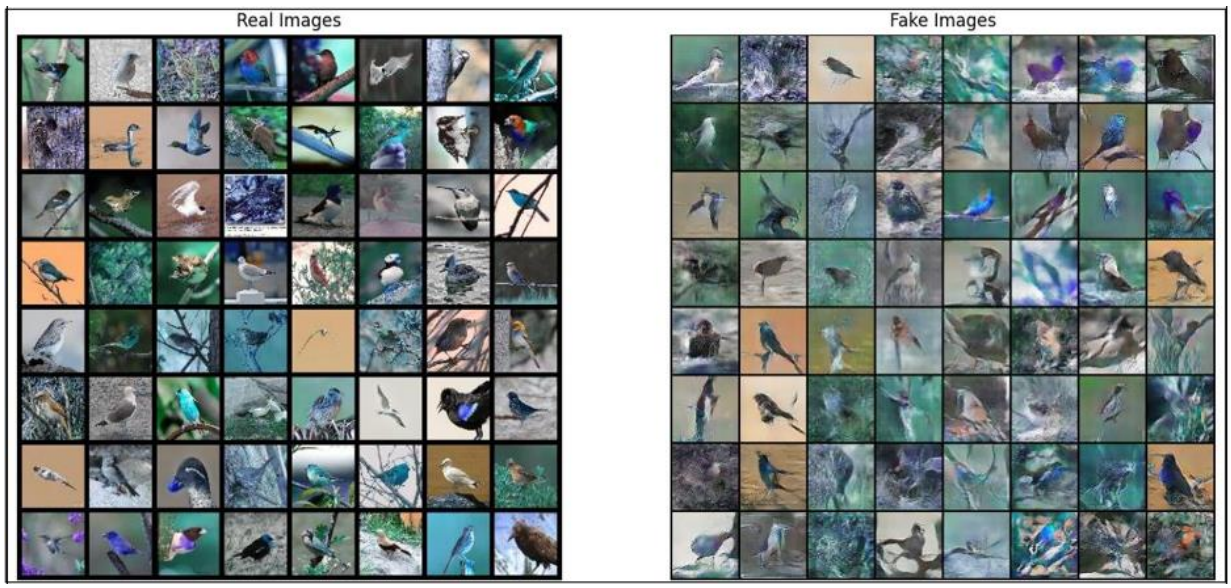


**Figure 9.** Real images vs. fake images comparison (64 × 64 × 3)

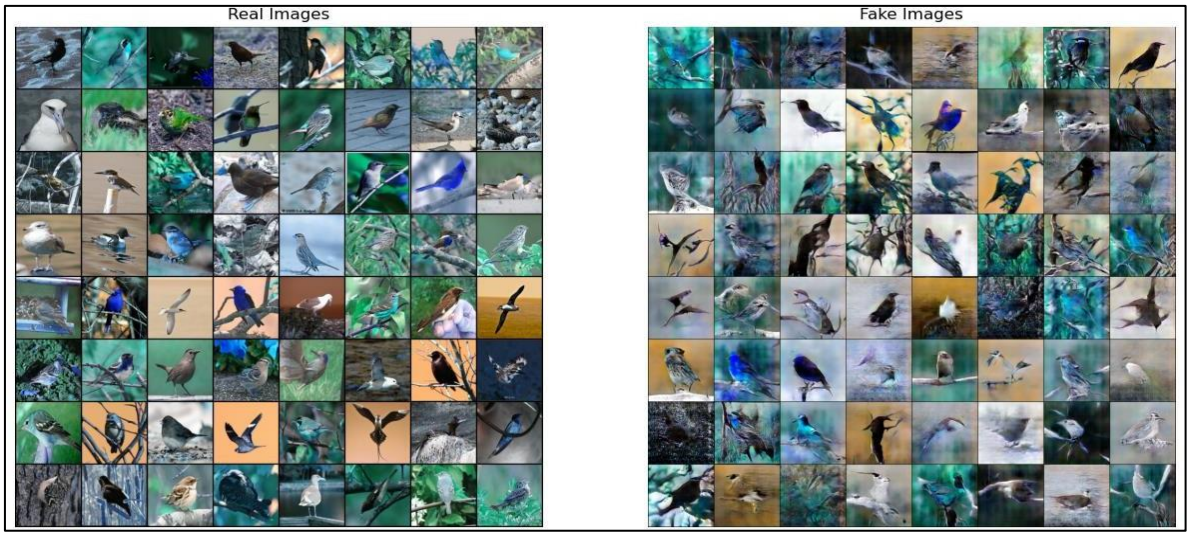


**Figure 10.** Real images vs. fake images comparison (256 × 256 × 3)

### 7.8 Challenges

Several technical challenges emerged during the study:

a) Missing text descriptions

The dataset provided caption embeddings but did not include the original textual descriptions due to privacy restrictions. This limitation restricted the ability to validate embedding semantics, a constraint also noted in prior text-to-image work [1].

b) Computational constraints

Training 256 × 256 GANs is computationally demanding. Limited GPU resources restricted the scope of hyperparameter exploration and prolonged debugging cycles, consistent with challenges reported in GAN training literature [4].

c) Sensitivity to hyperparameters

DCGANs are highly sensitive to hyperparameter configurations, including learning rate, batch size, and normalization strategies. In our early experiments, getting these wrong often led to common problems like discriminator overpowering, generator collapse and gradient instability. Addressing these issues required iterative tuning of architectural components and optimization settings.

d) Stability issues at low resolution

The 64 × 64 model exhibited training instabilities, confirming that low-resolution GANs struggle to pick up smaller, finer-grained features. These problems occurred mainly in datasets with high intra-class variability, such as CUB-200-2011 [15, 16]. These challenges provided valuable insights into adversarial training dynamics and informed the final design of our model.

## 8. CONCLUSION AND FUTURE WORK

With the CUB-200-2011 dataset as a testbed, this paper investigates Deep GANs' performance on fabricating bird images artificially. The following variant models were developed: unconditional 64 × 64 DCGANS and a conditional 256 × 256 DCGAN with label conditioning that incorporates embeddings of image captions. Results showed that higher resolutions, together with semantic conditionings, significantly improve generative modeling performance. The conditional 256 × 256 DCGAN achieves an FID of 52 and an IS of 4.1, further outperforming the unconditional 64 × 64 DCGAN, which yielded an FID of 65 and an IS of approximately 3.4. These quantitative improvements reflect enhanced texture fidelity and improved visual realism. Training also became more stable at higher resolution. The generator converged more smoothly and the discriminator exhibited less erratic behavior. Qualitative inspection confirmed that this 256 × 256 model produced clearer, more coherent bird images with bird species-level features recognizable in each one.

Despite these improvements, the study has limitations, including high computational cost and significant sensitivity to hyperparameter settings, which often led to training instability at lower resolutions—a challenge consistent with known behaviors of traditional GANs. These observations highlight the need for generative models that can operate in ways akin to linear programming models, with a threshold for the size of gradients and zero tolerance on deviating from given error boundaries if a highly accurate or "optimized" result is hardware-wise.

One direction for future work should be in advancing architectures of StyleGAN2, BigGAN, or diffusion models, which have shown exceptional fidelity and stability in the recent literature. Integrating such models into ecological research pipelines would enable much better data augmentation, visualizing species and biodiversity analysis. Expanding upon data sets with more species, habitats, or environmental impacts could further support these ecologically specific generative tasks, as well as help any models trained on the additional data generalise better.

Overall, this survey provides a foundational baseline for evaluating GAN performance across multiple resolutions in avian image synthesis, and it has implications that may carry forth into future research on generative modeling for ecological applications.

## REFERENCES

[1] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H. (2016). Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396. https://doi.org/10.48550/arXiv.1605.05396

[2] Radford, A., Metz, L., Chintala, S. (2015). Unsupervised representation learning with deep convolutional

generative adversarial networks. arXiv preprint arXiv:1511.06434. https://doi.org/10.48550/arXiv.1511.06434

[3] Mirza, M., Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784. https://doi.org/10.48550/arXiv.1411.1784

[4] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. (2016). Improved techniques for training GANs. arXiv preprint arXiv:1606.03498. https://doi.org/10.48550/arXiv.1606.03498

[5] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, pp. 5907-5915.

[6] Abhirami, K., Puvaneswari, S., Lakshmi, R.S., Mangalambigai, D., Revathy, G. (2024). Advanced bird species classification using self-attention and GAN-based data augmentation on BIRDS 525 dataset. In 2024 8th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 1087-1090. https://doi.org/10.1109/ICECA63461.2024.10800941

[7] Duy, H.A., Khoa, H.A., Hung, P.D. (2024). DCGAN-based method for improving animal classification in low resolution image. In 17th International Conference on Multi-Disciplinary Trends in Artificial Intelligence, Pattaya, Thailand, pp. 152-163. https://doi.org/10.1007/978-981-96-0692-4_13

[8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative adversarial nets. In NIPS'14: Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, pp. 2672-2680. https://doi.org/10.48550/arXiv.1406.2661

[9] Isola, P., Zhu, J Y., Zhou, T., Efros, A.A. (2017). Image-to-image translation with conditional adversarial networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 5967-5976. https://doi.org/10.1109/CVPR.2017.632

[10] Karras, T., Aila, T., Laine, S., Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196. https://doi.org/10.48550/arXiv.1710.10196

[11] Brock, A., Donahue, J., Simonyan, K. (2019). Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096. https://doi.org/10.48550/arXiv.1809.11096

[12] Karras, T., Laine, S., Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, pp. 4396-4405. https://doi.org/10.1109/CVPR.2019.00453

[13] Damayanti, F., Suzanti, I.O., Jauhari, A., Herawati, S. (2024). Restoration of Javanese characters based on Wasserstein Generative Adversarial Network-Gradient Penalty. Mathematical Modelling of Engineering Problems, 11(12): 3447-3457. https://doi.org/10.18280/mmep.111223

[14] Yang, H., Ma, Y., Khan, F. G., Khan, A., Ali, F., AlZubi, A.A., Hui, Z. (2024). Survey: Application and analysis of generative adversarial networks in medical images. Artificial Intelligence Review, 58(2): 39. https://doi.org/10.1007/s10462-024-10992-z

[15] Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S. (2011). The Caltech-UCSD birds-200-2011 dataset.

[16] Kundar, V. (2021). GANData20 Dataset. Kaggle. https://www.kaggle.com/datasets/vishalkundar/gandata20.