Winter in Data Science - WiDS
An Intro to Generative Models

Shaik Rehna Afroz(22B3932)
Electrical Engineering, IIT Bombay

1 Introduction

Deep Generative Models (DGMs) have revolutionized the field of artificial intelligence by enabling machines to learn and generate new data. This project explores the theoretical and practical aspects of three primary classes of DGMs: Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Diffusion Models. Each model type presents unique advantages and challenges, which will be explored in-depth.

2 Generative Adversarial Networks (GANs)

GANs consist of two neural networks, a generator and a discriminator, engaged in an adversarial game.

2.1 Theory Behind GANs

Introduced by Goodfellow et al. (2014), GANs optimize the min-max game:

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))] \tag{1}$$

where G generates samples from latent space z, and D differentiates real from fake samples.

Challenges in GAN training include:

- **Mode Collapse**: The generator produces limited variations, leading to repeated samples.
- **Training Instability**: The discriminator quickly overpowers the generator or vice versa.
- **Gradient Saturation**: Poor gradient flow results in slow or failed training.

Several improvements such as Wasserstein GANs (WGANs), Spectral Normalization, and Progressive Growing have been proposed to enhance stability and performance.

3 Variational Autoencoders (VAEs)

VAEs use probabilistic encoding and decoding, introducing a latent variable z governed by a prior distribution.

3.1 Mathematical Formulation

VAEs optimize the evidence lower bound (ELBO):

$$\log p(x) \geq \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z)) \tag{2}$$

where $q(z|x)$ approximates the true posterior $p(z|x)$.

Advantages of VAEs:

- Structured latent space allows controlled and meaningful generation.
- Regularization ensures smooth interpolation and representation learning.
- The reparameterization trick facilitates gradient-based optimization.

However, VAEs often generate blurry images due to over-regularization, leading to trade-offs between sharpness and latent space organization.

4 Diffusion Models

Diffusion models model data generation as a Markovian process of iterative denoising.

4.1 Diffusion Process

The forward diffusion process gradually adds Gaussian noise:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{3}$$

A neural network learns to reverse this process via a learned denoising function.

4.2 Denoising Score Matching

The objective is to minimize the variance-weighted mean squared error:

$$L_{denoise} = \mathbb{E}_{t,x_t,x_0}[||\epsilon - \epsilon_\theta(x_t, t)||^2] \tag{4}$$

where ϵ_θ approximates the noise at each timestep t.

Diffusion models offer superior sample quality and diversity but require extensive computational resources due to iterative denoising steps.

5 Comparison of Generative Models

A comparative analysis of these three generative models is outlined below:

Model	Strengths	Weaknesses
GANs	High-quality images, efficient	Mode collapse, instability
VAEs	Structured latent space, probabilistic	Blurry images
Diffusion Models	Stability, superior quality	High computational cost

5.1 Key Differences

- **Training Paradigm**: GANs use adversarial training, VAEs rely on variational inference, and Diffusion Models employ iterative denoising.
- **Sample Quality**: GANs produce sharp images but suffer from mode collapse; VAEs offer diverse but blurrier samples; Diffusion Models generate high-fidelity outputs at the cost of increased computation.
- **Computational Cost**: Diffusion models require extensive computation, while GANs and VAEs are more efficient in inference.

Empirical results indicate that diffusion models outperform GANs and VAEs in terms of sample fidelity and diversity but are computationally expensive.

6 Results

Models were trained on different datasets like MNIST, CIFAR10, CelebA

6.1 Datasets

6.1.1 MNIST Dataset

MNIST is a dataset of handwritten digits (0-9), containing 60,000 training and 10,000 test images of size 28x28 in grayscale. It is widely used for benchmarking generative models.

6.1.2 CIFAR-10 Dataset

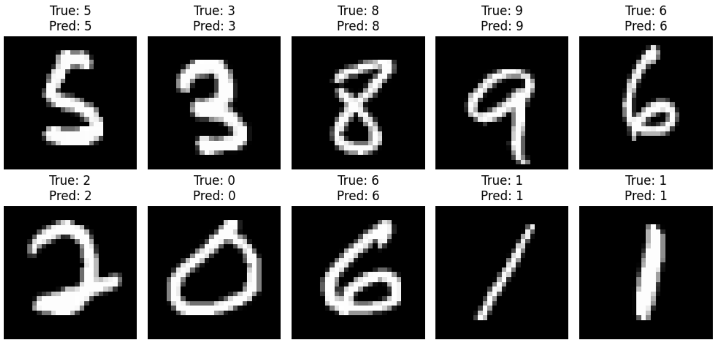
CIFAR-10 consists of 60,000 images across 10 classes, with 50,000 training and 10,000 test images. Each image is 32x32 pixels in RGB format, making it a more complex dataset compared to MNIST.

6.1.3 CelebA Dataset

CelebA is a large-scale dataset containing over 200,000 images of celebrities with various facial attributes. It is commonly used for facial image generation and editing tasks in deep learning.

6.2 Feed Forward Neural Network from scratch(MNIST)

- **Architecture**: layers: [784, 64, 32, 10], epochs: 100, batch size: 20
- **Test Accuracy**: 97.34%



6.3 Feed Forward Neural Network from scratch(CIFAR10)

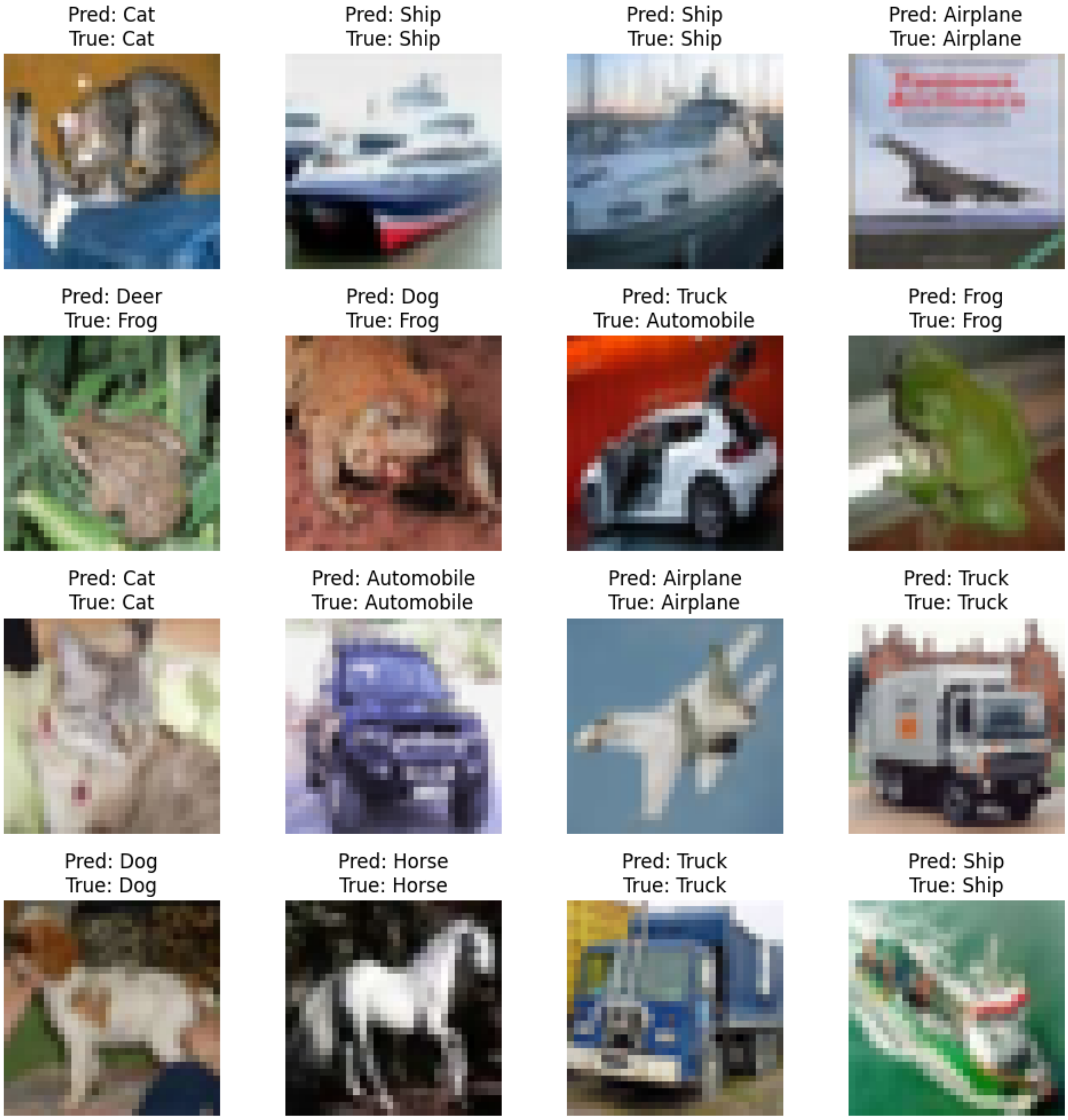
- **Architecture**: layers:[3072, 128, 64, 10], epochs: 100, batch size: 64
- **Test Accuracy**: 51.87%

- FNN doesnot give good results on complex datasets like CIFAR 10
- This is because Feedforward neural networks (FNNs) do not have built-in mechanisms to handle spatial information or relationships between pixels
- Recognizing digits is simpler than recognizing complex objects (like animals in CIFAR-10), so FNNs can perform decently on MNIST
- The lack of spatial invariance means FNNs cannot generalize well to datasets with large variations in object position, size, or orientation. In such cases, CNNs are vastly more efficient and effective because of their ability to focus on patterns regardless of their location in the image.

6.4 Convolutional Neural Network(CNN) on CIFAR10

- **Architecture**:
  CNN with 3 Convolutional layers and 2 Dense layers
  10 neurons in the final output layer
  epochs=100, batch size=20

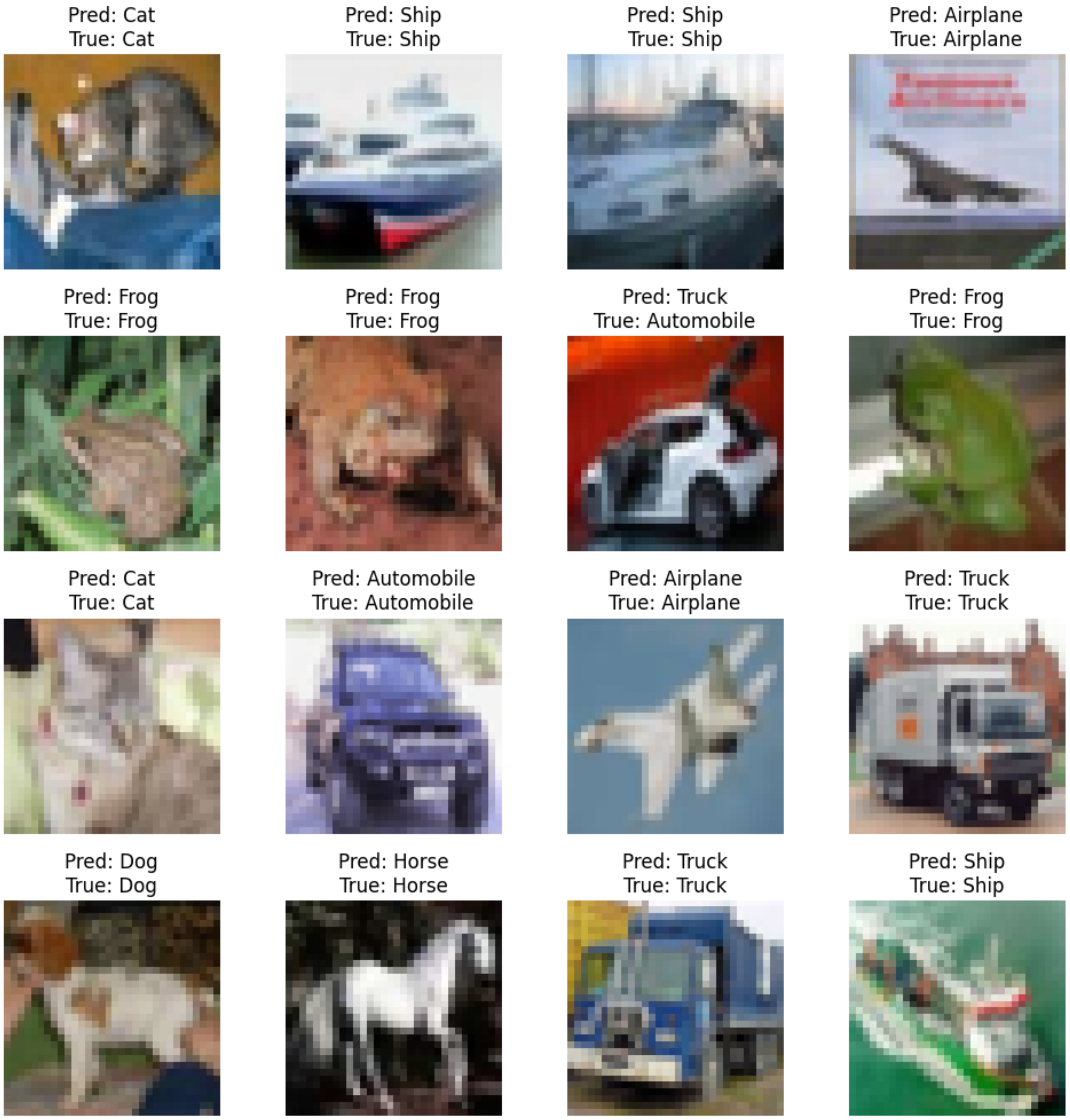
- **Test Accuracy**: 71.3%



6.5 AlexNet on CIFAR10

- **Architecture**: 5 Convolutional layers and 1 Dense Layer
  10 neurons in the final output layer
  epochs=100, batch size=50

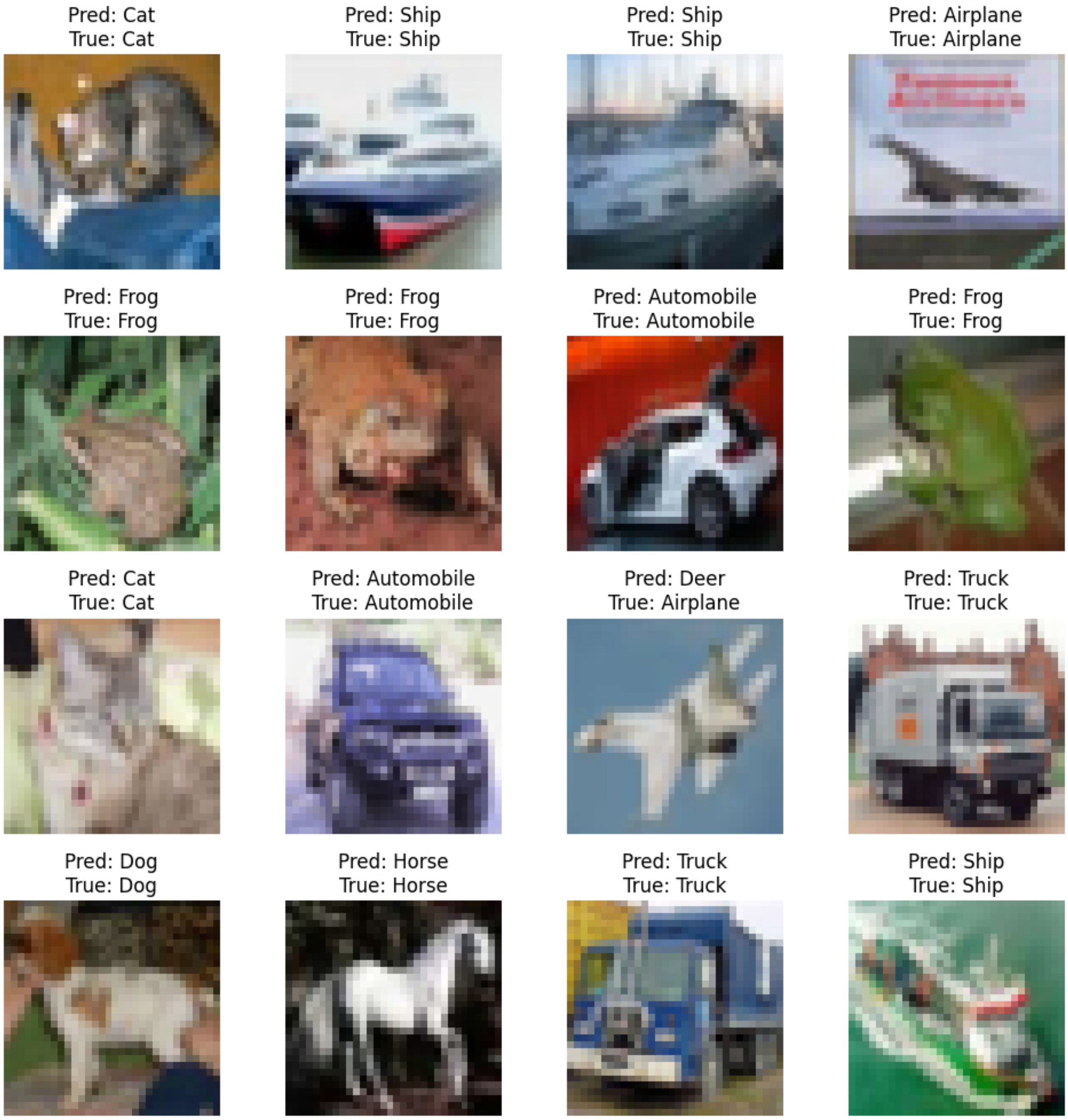
- **Test Accuracy**: 82.24%



6.6 ResNet(Residual Network) on CIFAR10

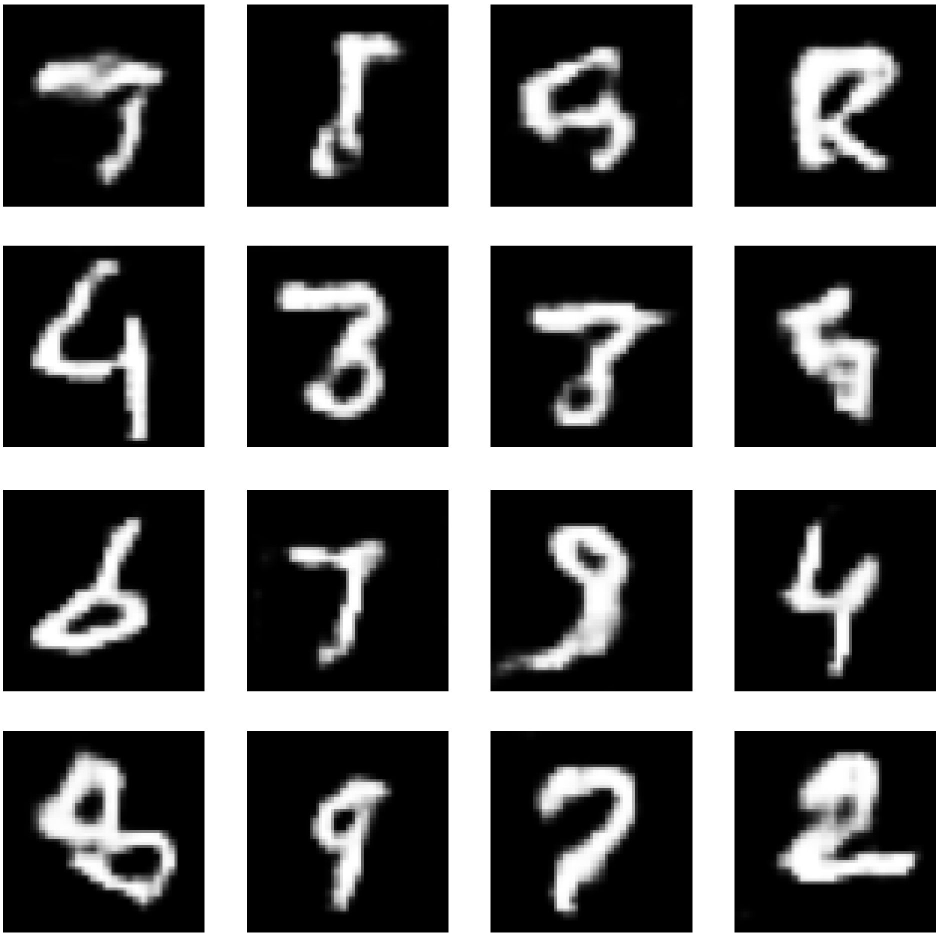
- **Architecture**: ResNet50
  10 neurons in the final output layer
  epochs=100, batch size=50

- **Test Accuracy**: 70.89%



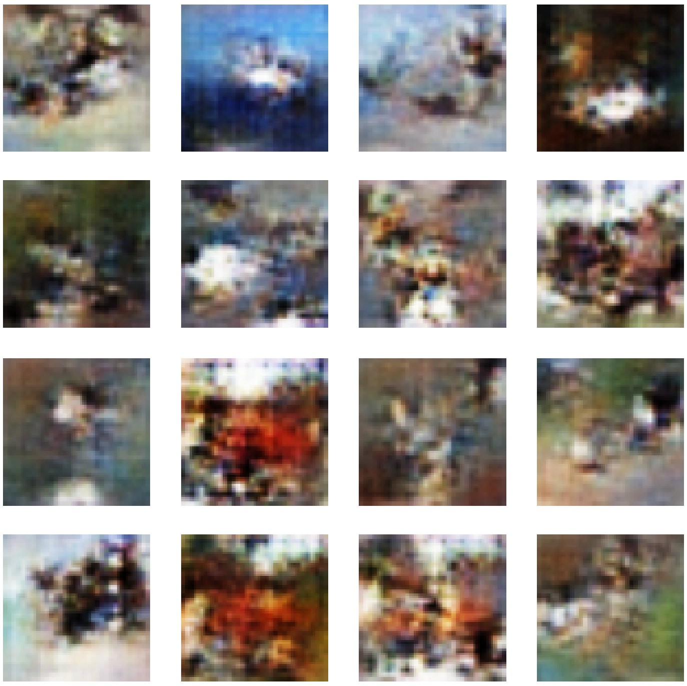
6.7 Deep Convolutional Generative Adversarial Network(DCGAN) on MNIST

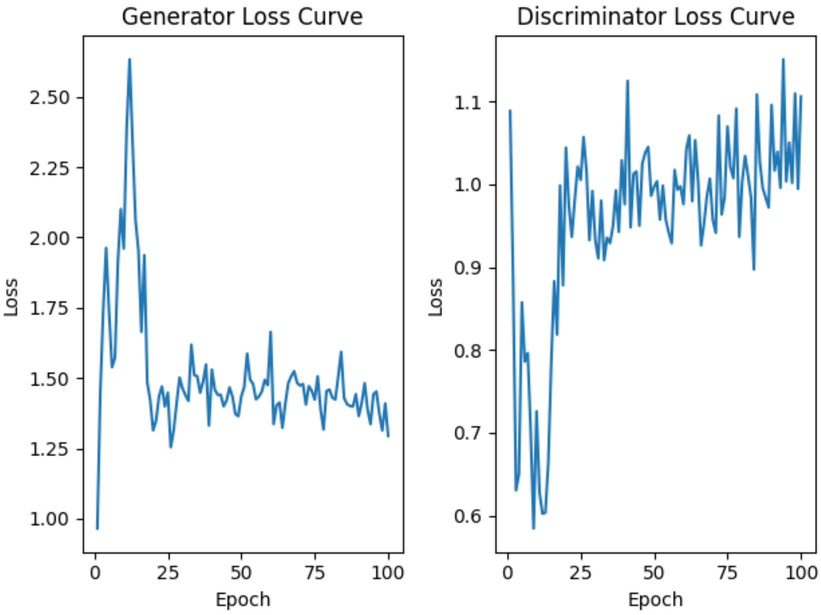
- **Architecture**:
  Generator: 1 Dense layer, 3 Conv2DTranspose layers
  Discriminator: 2 Conv layers , 1 Dense Layer
  epochs=100, batch size= 256



6.8 Deep Convolutional Generative Adversarial Network(DCGAN) on CIFAR10

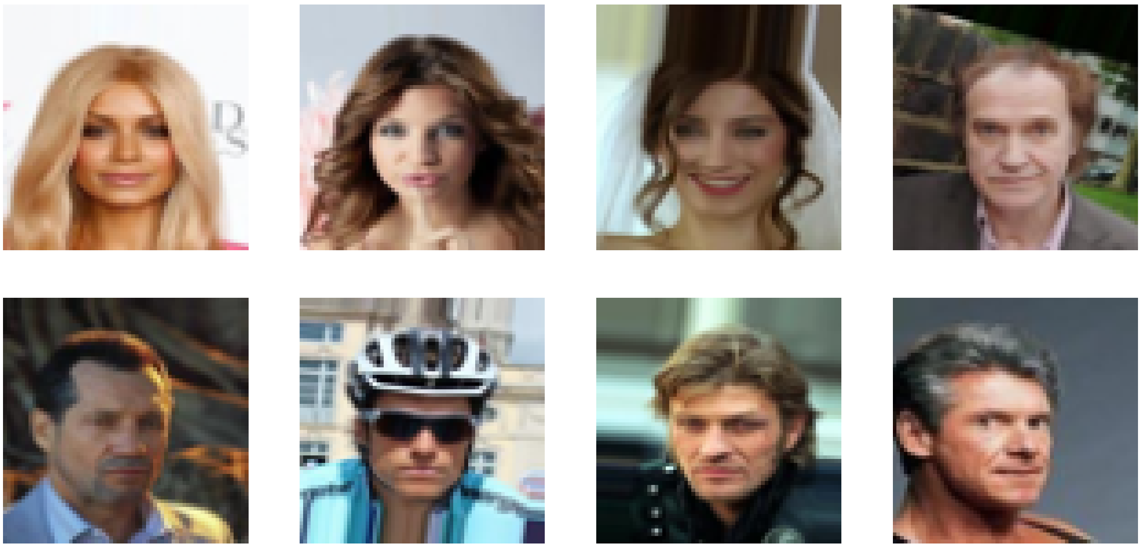
- **Architecture**:
  Generator: 1 Dense layer, 3 Conv2DTranspose layers
  Discriminator: 2 Conv layers , 1 Dense Layer
  epochs=100, batch size= 256
- **Minimum Generator Loss:** 0.965
- **Minimum Discriminator Loss:** 0.584

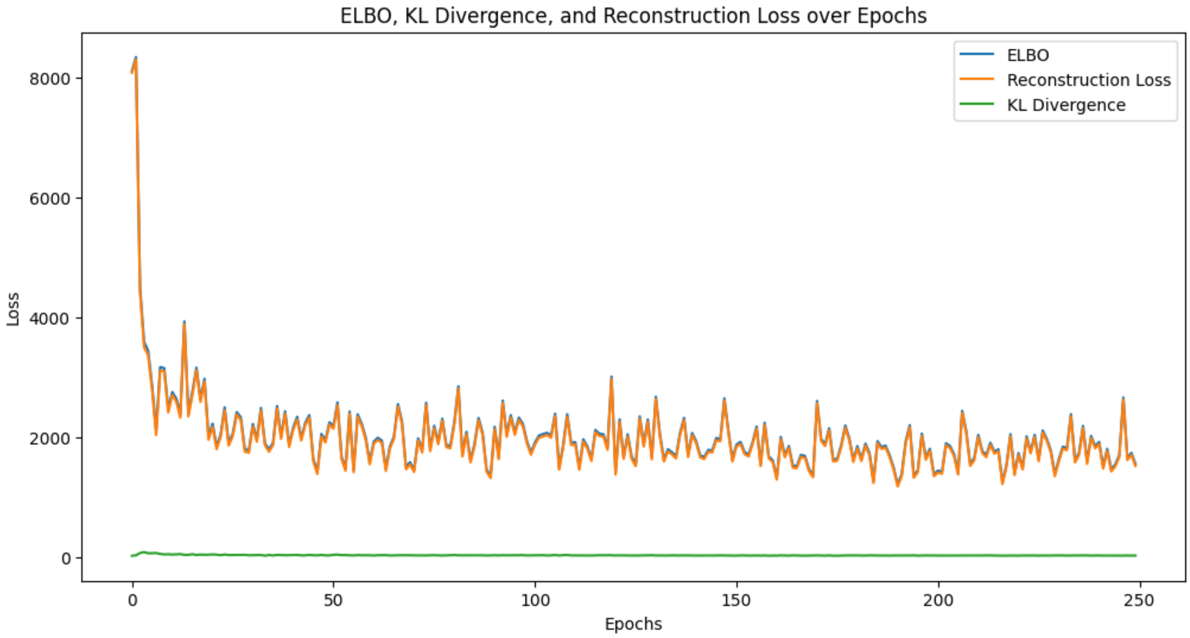




6.9 Variational Autoencoders(VAE) on CelebA

- **Architecture**:
  Encoder: 3 Conv layers , 1 Dense Layer
  Decoder: 1 Dense layer, 4 Conv2DTranspose layers
  epochs= 250





7 Conclusion

This project explores the theoretical and practical aspects of DGMs, comparing their performance in generating images. GANs offer efficiency but struggle with mode collapse, VAEs provide structured representations but produce blurrier outputs, and Diffusion Models generate high-quality images at the cost of increased computation.