DCGAN on CIFAR-10: Convolutional GAN Implementation

Project Overview & Objectives

🎯 Project Goals

- **Transform** original dense-layer GAN → Modern Convolutional GAN (DCGAN)
- **Upgrade** from MNIST (28×28 grayscale) → CIFAR-10 (32×32 RGB)
- **Focus** on single object class generation (airplanes)
- **Demonstrate** noise vector control techniques

📊 Key Improvements

- **Architecture:** Dense layers → Conv2D/Conv2DTranspose layers
- **Dataset:** Grayscale digits → Color object images
- **Quality:** Better feature learning with convolutional operations
- **Control:** Advanced latent space manipulation

DCGAN Architecture Design

🏗️ Generator Architecture

```
Noise (100D) → Dense(4×4×512) → Reshape
    ↓
4×4×512 --[Conv2DTranspose]--> 8×8×256
    ↓
8×8×256 --[Conv2DTranspose]--> 16×16×128
    ↓
16×16×128 --[Conv2DTranspose]--> 32×32×64
    ↓
32×32×64 --[Conv2DTranspose]--> 32×32×3 (RGB Image)
```

🔍 Discriminator Architecture

```
32×32×3 --[Conv2D]--> 16×16×64
    ↓
16×16×64 --[Conv2D]--> 8×8×128
    ↓
8×8×128 --[Conv2D]--> 4×4×256
    ↓
4×4×256 --[Conv2D]--> 4×4×512 → Flatten → Dense(1)
```

⚙️ Key Features

- **Batch Normalization** for stable training
- **LeakyReLU** activation functions
- **Dropout** in discriminator for regularization

CIFAR-10 Implementation & Training

📁 Dataset Processing

- **CIFAR-10:** 50,000 training images, 10 classes
- **Class Selection:** Filtered for airplanes (Class 0)
- **Preprocessing:** Normalized to [-1, 1] range
- **Result:** ~5,000 airplane images for training

🚀 Training Strategy

- **Epochs:** 3,000 iterations
- **Batch Size:** 32 (optimized for stability)
- **Optimizer:** Adam (lr=0.0002, β_1=0.5)
- **Alternating Training:** Discriminator → Generator each epoch

📈 Training Monitoring

- **Loss Tracking:** Both discriminator and generator losses
- **Image Saving:** Generated samples every 200 epochs
- **Progress Visualization:** Real-time quality assessment

Noise Vector Control Demonstrations

1️⃣ Fixed Noise Consistency

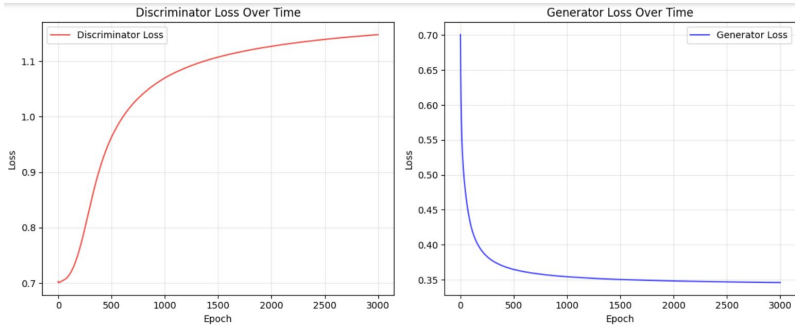
- **Same noise vector → Identical generated images**
- **Proves:** Deterministic generation process

2️⃣ Smooth Interpolation

- **Linear blending** between two noise vectors
- **Result:** Smooth morphing between different airplane designs

3️⃣ Magnitude Scaling

- **Different scaling factors** (0.2× to 2.0×) on noise vector
- **Effect:** Controls image intensity and feature prominence




```
Final Discriminator Loss: 1.1479
Final Generator Loss: 0.3459
```