

LAB 12: Implement a Deep convolutional GAN to generate complex color images.

AIM: To implement a Deep convolutional Generative Adversarial Network (DcGAN)

Pseudo code:

- Import libraries:
 torch, matplotlib
- Load dataset.
 Use & label dataset
 Normalize image
- Define Generator Network
- Define Discriminator Network
 Input: Real or fake images
- Initialize both networks w/ optimized
 → Adam optimizer
- Training loop:
 - Visualize
- Display generated color images.
- Compare evolution across epochs.

Observation:

- During the initial epochs, generated images are random noise with no structure
- As training progresses, the Generator learns color patterns, features and shapes resembling real images.

- The discriminator loss oscillates between 0 and 1.
- After sufficient training, DCGAN produces visually realistic and colorful synthetic images
- The quality of generated images depends on dataset complexity, network depth and training stability.

Result:

Successfully implemented a DCGAN capable of generating complex, realistic color images.

~~Experiments~~

1. MNIST

MNIST

2. CelebA

3. CIFAR

4. LSUN

5. Cityscapes

6. COCO

7. CarNets

8. Pix2pix

9. CycleGAN

10. SPADE

11. ESRGAN

12. SAGAN

13. DCGAN

output

epoch

1

Training no-5

166-23

2

152-45

3

143-10

4

132-87

10

121-34

11

114-89

20

109-76

Architecture

GAN

latent random variable

