

25/10

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 images
- 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 G & D.
- 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.

expt

output

epoch

Training no-3

166-23

1

152.45

2

3

143.10

4

132.87

10

121.34

11

114.89

20

109.76

Architecture of GAN

latent random variable

Real data samples

Generate fake samples.

Generator

Discriminator

Condition is it correct

