Ex. 12
10

Implement a deep convolutional GAN it generate complex color images

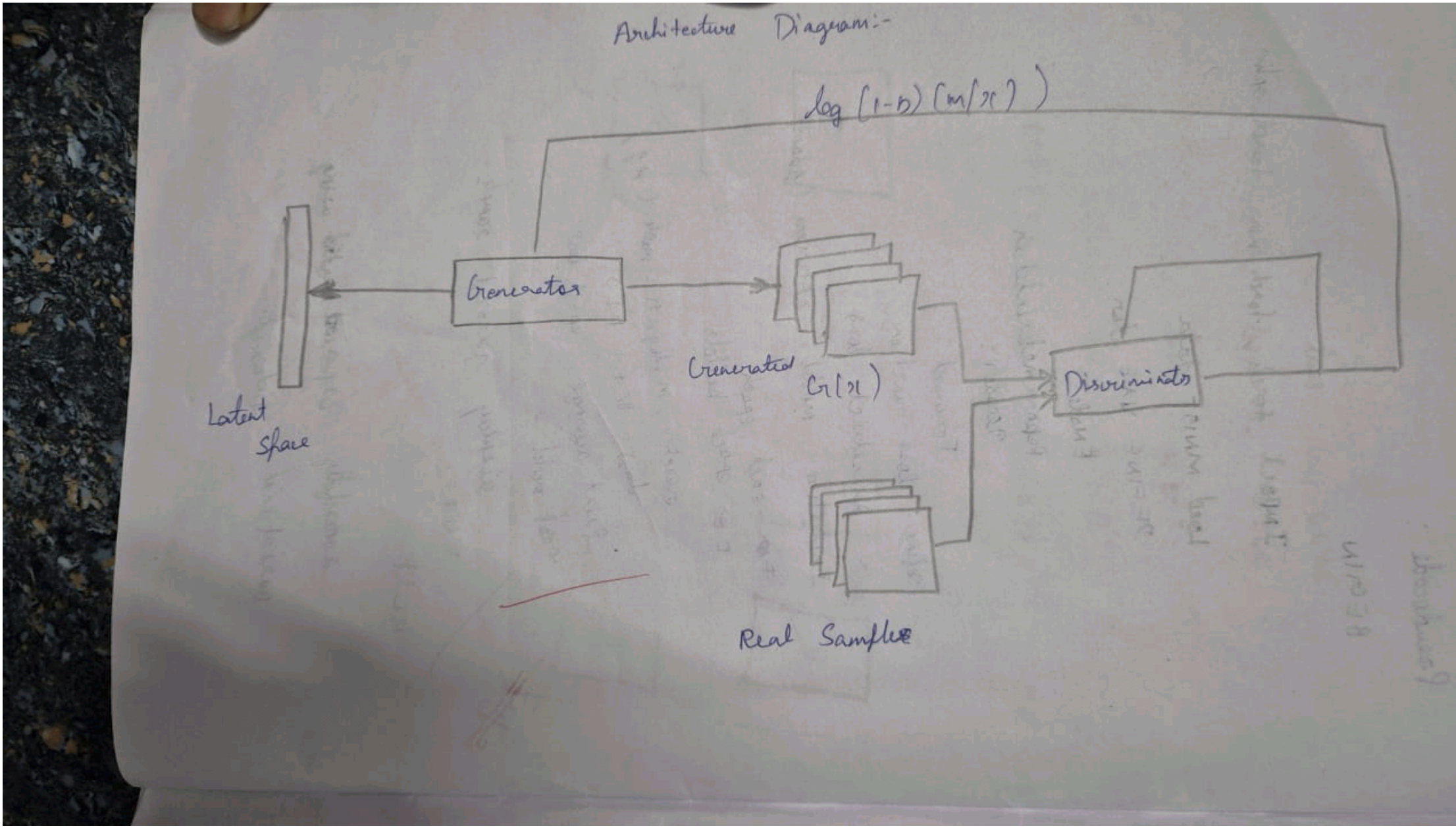
Aim :

To implement and train a deep convolutional generative adversarial network (DCGAN) capable of generating complex color images using convolutional and deconvolutional neural network

objectives :

1) To study the structure and components of a Pre-trained deep learning model
2) To understand how convolutional layer extract image features
3) To analyze the role of each layer in feature learning and classification
4) To observe how Pre-trained model can be used for training learning.
5) To visualize and interpret feature maps from different layer

Architecture Diagram:-



Pseudo code:

1. Load and normalize dataset to [-1, 1]
2. define generator (ConvTranspose 2D, ReLU, Tanh)
3. Refine discriminator (Conv2D, LeakyReLU + Sigmoid)
4. Initialize weights, set loss = BCE, optimizer = Adam.
5. for each epoch:
   a. Train D on real and fake images
   b. Train G to fool D using random noise
6. generate and save fake color images

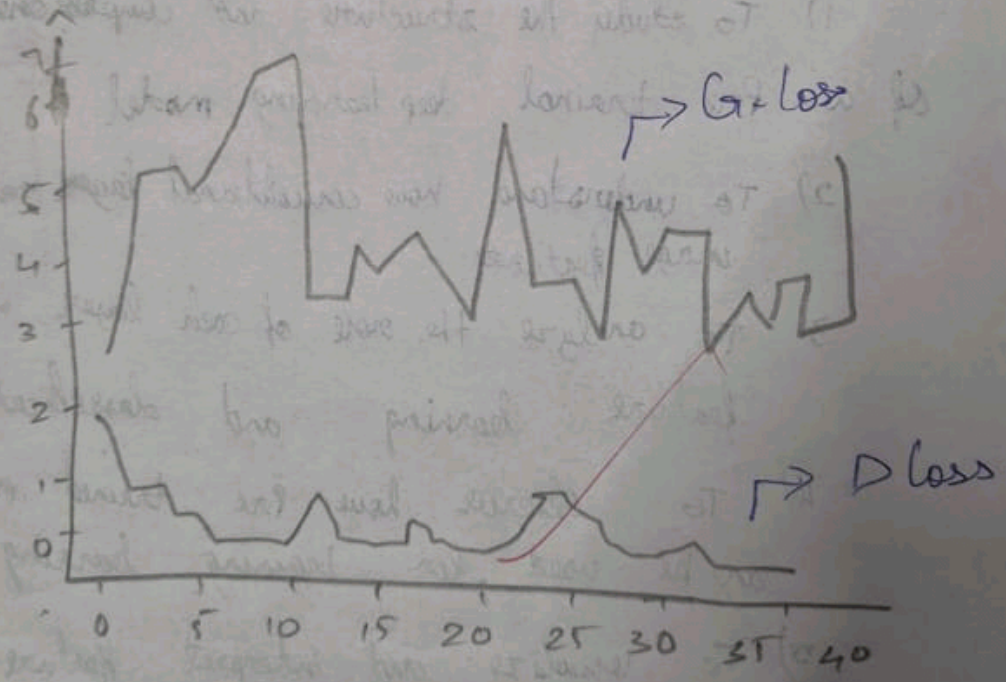
Result :-

Successfully Implemented a deep convolutional Gan to Generate complex images.

lassification Report :-

	Precision	recall	f1-Score	Support
Fake	0.993	0.993	0.993	143.000
real	0.994	0.994	0.994	177.000
uracy	0.994	0.994	0.994	0.994
acro-avg	0.994	0.994	0.994	320.000
ight avg	0.994	0.992	0.994	320.000

raph:-



Ex:13.

understanding the Architecture of a Pre-trained model

Aim:

To study and understand the structure layers and working of a Pre-defind deep learning model

objective:

1. To study the structure and componants of a pre-trained deep learning model.
2. To understand how convolutional layer extract image features.
3. To analyze the role of each layer in feature learning and classification.
4. To observer how pre-trained models can be used for training learning
5. To visualize and interpret feature maps from different layers.

Classification Report:-

	Precision	Recall	f1-Score	Support
Airplane	0.78	0.89	0.84	1000
Automobile	0.87	0.89	0.88	1000
Bird	0.81	0.69	0.74	1000
Cat	0.66	0.71	0.68	1000
deer	0.76	0.79	0.77	1000
dog	0.77	0.76	0.76	1000
frog	0.82	0.83	0.82	1000
horse	0.84	0.81	0.82	1000
Ship	0.90	0.81	0.86	1000
Truck	0.90	0.84	0.87	1000
Acc	0.81	0.80	0.80	10000
M.Acc	0.81	0.80	0.80	10000
W.Acc			0.80	10000

Pseudo code:

1. Import required libraries (torchvision/verax)
2. Load a Pre-trained model (eg, resnets)
3. Print and examine the model architecture
4. Inputs an image and preprocess it (resize, normalize)
5. Pass the image through the model
6. visualize outputs or feature maps from intermediate layers
7. Analyze how early, middle, and deep layers capture features.

Result:

successfully understand the architecture of a Pre trained model.