Expt 12:

Implement a Deep Convoluted GAN to Generate Complex Color Images

Aim:

To generate complex color images using Deep Convoluted Generative Adversarial Networks

Objective

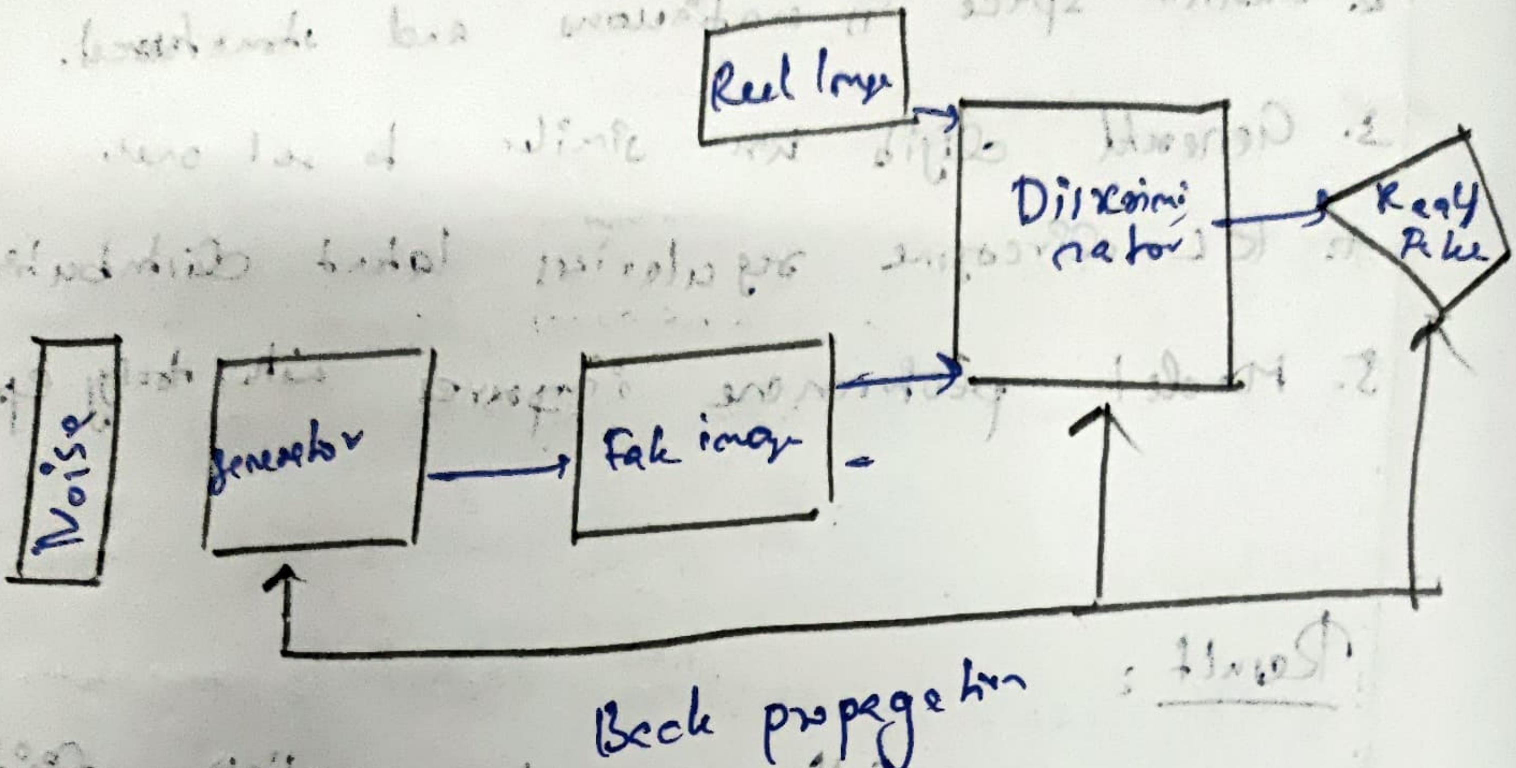
1. To understand GAN architecture and training mechanism
2. To design generator and discriminator network
3. to train model adversarially on color image data.
4. To analyze convergence behavior
5. To ~~generate~~ generate realistic ~~colors~~ color images.

Pseudocode:

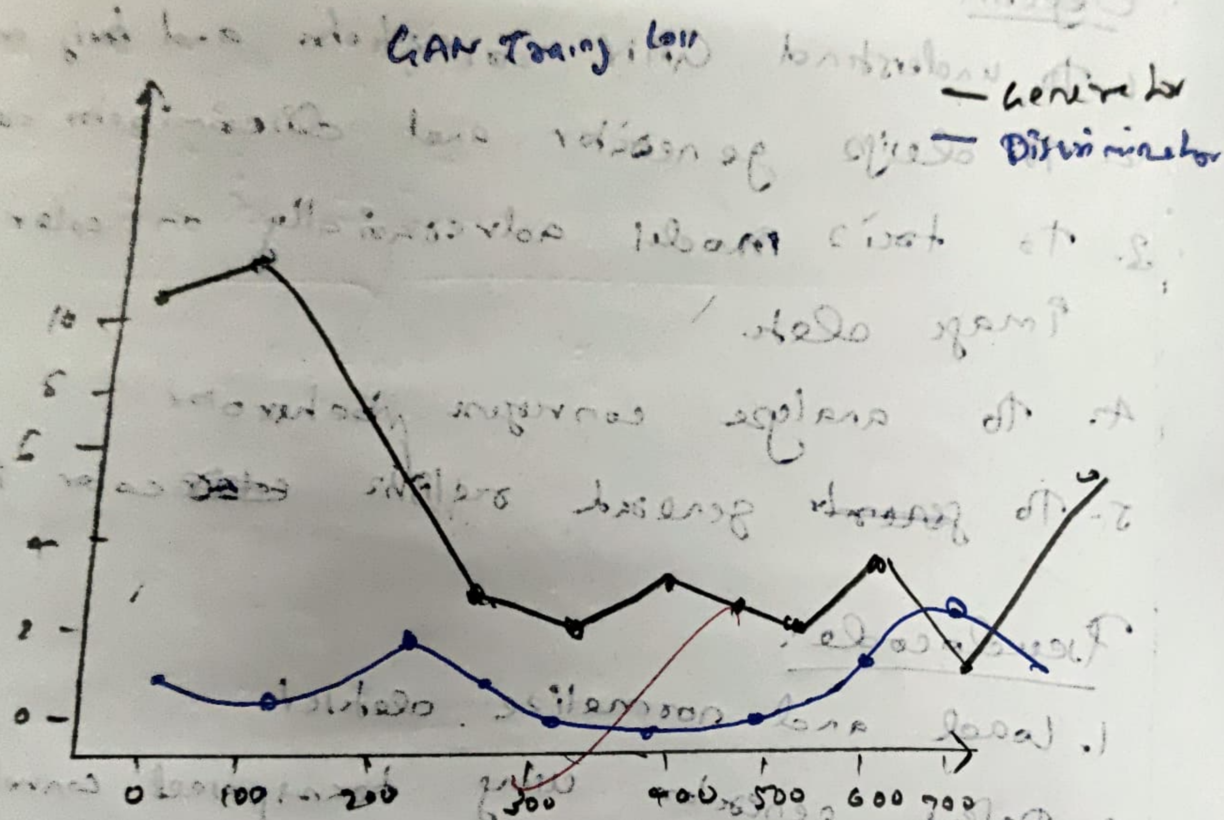
1. Load and normalize dataset
2. Define generator using transposed convolutions
3. Define discriminator with convolutional layer
4. Train both networks using adversarial loss.
5. Generate and visualize new images.

Observation

1. Generator and discriminator losses oscillate initially
2. Image quality improved gradually.


Real Image
Discriminator
Real Fake
Noise
Generator
Fake image
Back propagation

Epoch	Generator loss	Discriminator loss
1	4.85	0.90
10	2.40	1.10
20	1.60	1.25
40	1.20	1.30



3. Generated samples resemble real images after several epochs

4. Proper learning rate ~~[illegible]~~ tuning stabilizes training

5. DCGAN effectively captures texture and col-

Result

DCGAN successfully generated realistic and visually appealing color images.

Exp 13.

Understanding the Architecture of Pre-trained Model

Aim:

To study and understand the layer-wise architecture of pre-trained CNN model such as VGG16 or ResNet.

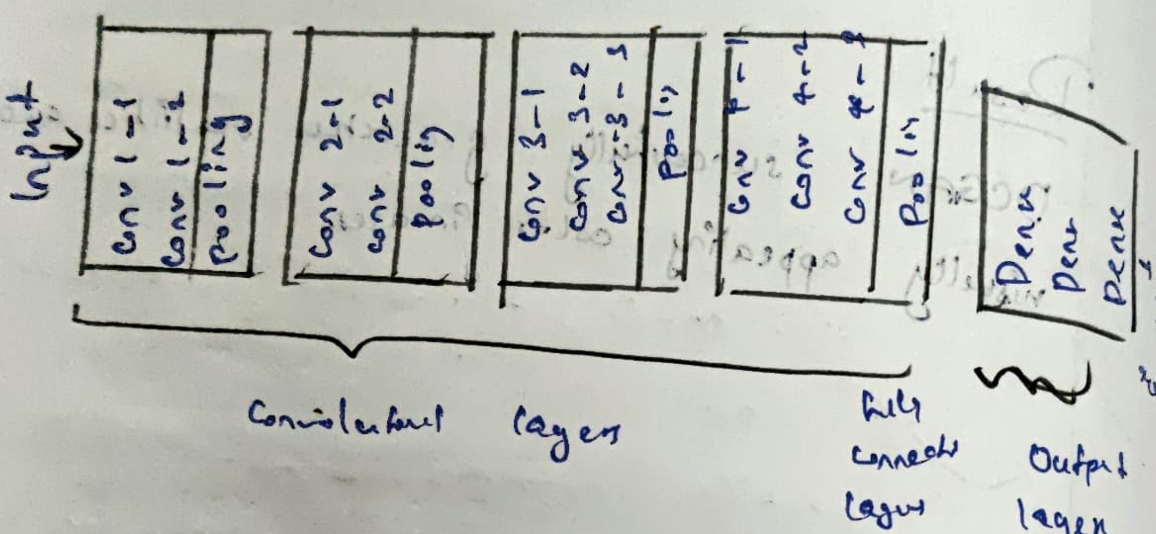
Objective:

1. To load pre-trained CNN models
2. To analyze convolutional, pooling, and fully connected layers
3. To understand hierarchical feature extraction.
4. To explore model parameters and depth
5. To visualize feature maps.

Pseudocode.

1. Import pre-trained model (eg. VGG16) from Keras
2. Display model summary
3. Visualize initial and deep layer activation
4. Observe layer type and parameters
5. Interpret extracted features.

VGG16 Architecture



Observation:

1. Early layers detect edges and textures
2. Deeper layers identify complex shapes.
3. Model parameters are pretrained on ImageNet
4. CNN depth increases abstraction level
5. Model layers can be reused for transfer learning

Result:

Successfully understood hierarchical feature learning and architecture of pre-trained CNN models.

Layer Type	Output Shape	Param#
Conv 2D	(224,224,64)	1792
Max Pooling2D	(112,112,64)	0
Conv.2D	(112,112,128)	73856
Flatten	(25088)	0
Dense	(4096)	102764544