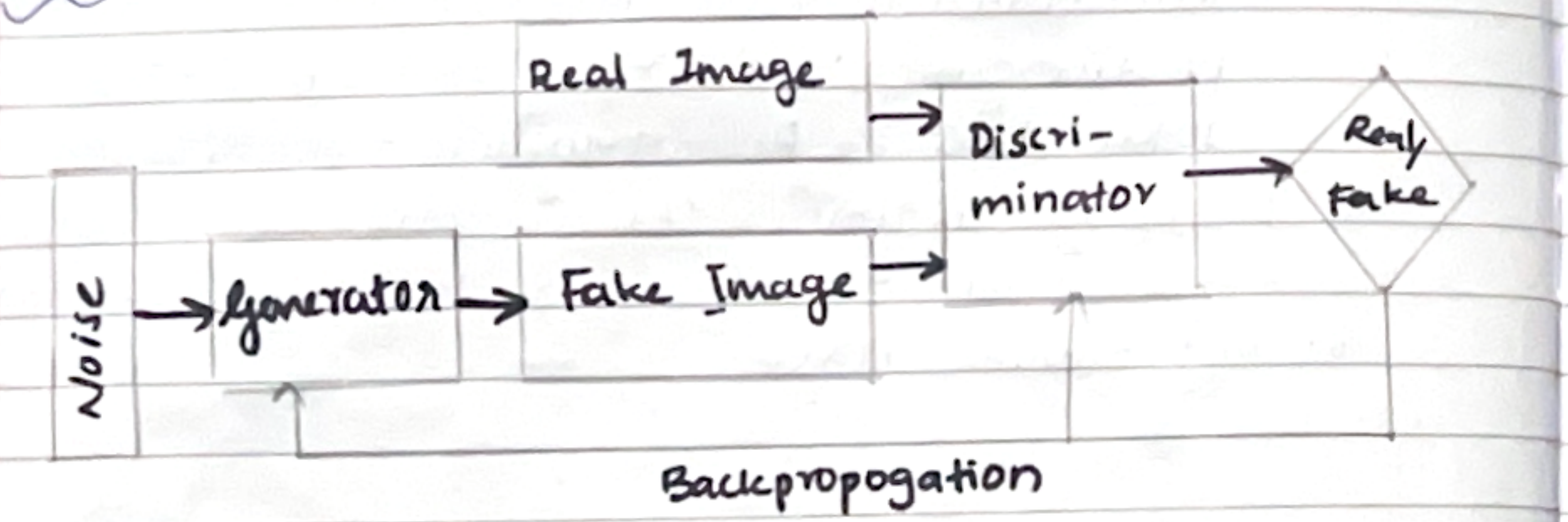
DCGAN



DCGAN improves GANs by using :-

- Convolutional layers instead of fully connected.
- Batch normalization for stable training.
- Leaky ReLU for discriminator and ReLU for generator.

27/10/25

LAB-12

Implement a Deep Convolutional GAN to Generate Complex Color Image

AIM:

To implement a DCGAN using PyTorch to generate complex color images from random noise.

OBJECTIVES:

1. To understand how GAN work.
2. To design and train a DCGAN using convolutional and transpose convolutional layers.
3. To generate synthetic color image similar to the dataset images
4. To visualize and analyze generator and discriminator performance during training.

PSEUDOCODE:

```
1. Import libraries and define
2. Load and preprocess dataset (CIFAR-10)
3. Define generator and Discriminator architectures
   using CNN layers.
4. Initialize models, loss function (BCE) and
   optimizers (Adam)
5. Train for epochs:
   (a) Train Discriminator :
       - Compute loss on real image (label = 1)
       - Compute loss on fake image (label = 0)
       - update weights
```