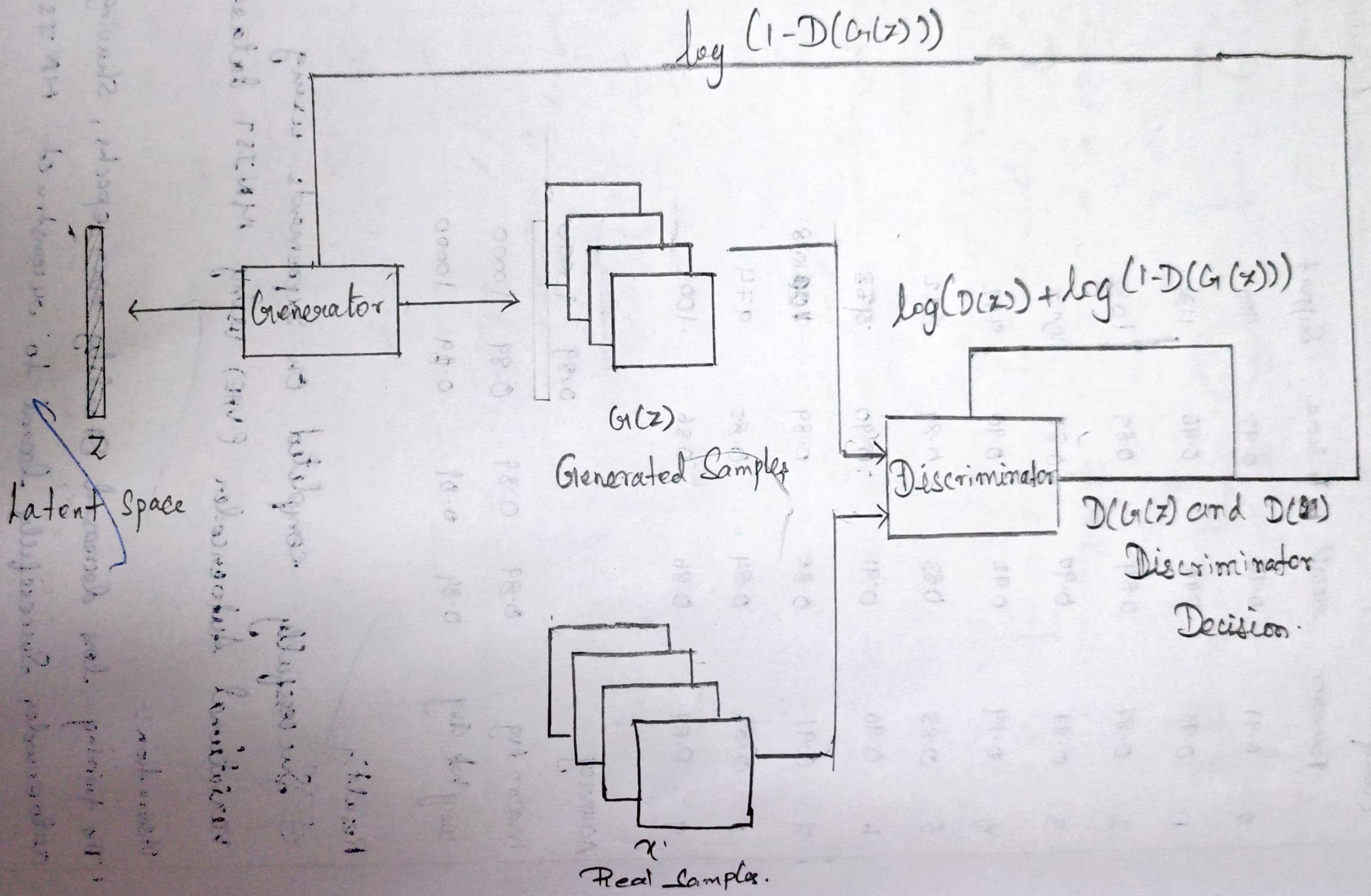
Architecture Diagram GAN...



[illegible]/10/25 Exp:- 12: Implement a Deep Convolution GAN To generate Complex color image.

Aim:-
To implement a Deep Convolution GAN that generates complex-looking color image and to Evaluate generated images via a simple classifier producing a classification report for real vs fake images.

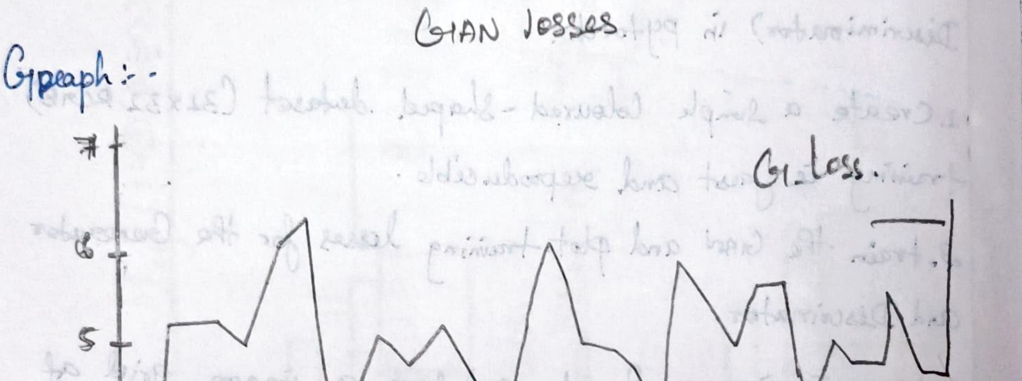
objectives:-
1. Implement a lightweight Deep Convolution GAN (Generator + Discriminator) in pytorch.
2. Create a simple coloured-shaped dataset (32×32 RGB) so training is fast and reproducible.
3. train the GAN and plot training losses for the Generator and Discriminator.
4. Generate images samples and save an image grid of samples.
5. train a small CNN classifier to distinguish real and fake images and output a classification report.
6. observe and report findings.

Pseudocode:-
```
1. Build a synthetic RGB datasets (colour shapes) that produces
   Coloured shapes on a 32×32 canvas.
2. Deep Convolution GAN ~~to~~
   • Generator : ConvTranspose layer → Batch Norm → ReLU →
     Tanh, output 32×32×3
   • Discriminator :- Conv2d → Batch Norm → Leaky Relu → Sigmoid,
     outputs probability real / fake.
```

Classification Report:-

	Precision	recall	f1-score	Support
Fake	0.993	0.993	0.993	143.000
real	0.994	0.994	0.994	177.000
accuracy	0.994	0.994	0.994	0.994
				320.000
macro-avg	0.994	0.994	0.994	
Weighted avg	0.994	0.994	0.994	320.000

Graph:-



3. Train loop:-
- For each batch:-
  - Update Discrimator on real batch (lable=1) and fake batch (label=0).
  - Update Generator to fool Discriminator (label=1 for generated outputs).
- Collect losses....

4. After training:-
- Generate a grid of Samples from fixed noise vector.
- Build evalution Set: N real images + N Generated images.
- Train a Small CNN classifier on real vs ~~fake~~ fake
- Comput Classification_report ~~on~~

5. Save grid image and report to disk; Plot losses.

Observations:-

- Observation 1 (training):- Generator and discriminator losses oscillate; Stable growth in sample quality comes from balanced updates
- Observation 2 (samples):- After training, generated images reproduce colored shaped Similar to the dataset = but may Show artifacts if the model capacity or epochs are limited.

Result:-

The implementated DCGAN Sucessfully generated color images that visually match training distribution; the classification report gives quantitative Support for Generator's quality.

[illegible]