**🧾 Summary of What We Did**

**🎯 Your Goal**

You're building a **photo-to-cartoon conversion model** using **CycleGAN** — a type of Generative Adversarial Network (GAN) great for unpaired image translation.

**🛠️ Steps You've Completed**

**1. Loaded Your Dataset**

* You used the UnalignedDataset class.
* Found **61 images** — used for training the model to learn translation from photos → cartoons and vice versa.

**2. Created and Initialized the CycleGAN Model**

* 4 networks were initialized:
  + G\_A (Photo → Cartoon)
  + G\_B (Cartoon → Photo)
  + D\_A, D\_B (Discriminators for each domain)
* Parameters were initialized with normal distribution.
* Model was set up and ready for training.

**3. Visualized Training**

* Used a Visualizer class to print out losses:
  + G\_A, G\_B → generator losses
  + D\_A, D\_B → discriminator losses
  + cycle\_A, cycle\_B, idt\_A, idt\_B → cycle consistency and identity losses
* Observed logs like:

yaml

CopyEdit

(epoch: 2, iters: 39) D\_A: 0.327 G\_A: 0.203 cycle\_A: 1.069 ...

**4. GPU Support Added**

* Detected and successfully used your **NVIDIA GeForce RTX 3050 Laptop GPU** to speed up training (from CPU).
* Added torch.cuda.set\_device() and model.to(device) under the hood using your framework.

**5. Reduced Epochs for Quick Run**

* Changed the config to only train for **10 epochs** for tonight — enough to see early results.
* You’ll later increase epochs for better results once you're rested and back.

**6. Cleared GPU Memory Before Restart**

* Used torch.cuda.empty\_cache() and gc.collect() to clear GPU memory before restarting the session to avoid any out-of-memory issues or stale sessions.