Google drive link :

https://drive.google.com/drive/folders/1VfcQNmcNryvUMP0OHNIRiI3xw0_oZcwL?usp=sharing

---

DBL system :

project@hebbel:~/pytorch-CycleGAN-and-pix2pix

Cyclegan :

- images fetched from directory : ./datasets/bau2nonbau

- Should contain sub-folders trainA , trainB , testA and testB for cyclegan code to execute successfully

- trainA and testA : 100 RGB images of varied sizes of Bauhaus style architecture
- trainB and testB : 100 RGB images of varied sizes of Non-Bauhaus style architecture (Indian, Chinese, Greek, European, Modern and trending , Egyptian , Middle-eastern , Churches)

using sneha_test_cyclegan.ipynb : train and test bidirectional AB and BA.

Trained images stored under ./checkpoints/bau2nonbau/web/images

epoch:200
learning rate (lr) : 0.0002
batch_size: 1
beta1 : 0.5
crop_size: 256
resnet : 9 blocks

Test results stored under : ./results/bau2nonbau/test_latest/images

batch_size: 1
preprocessing : resize and crop
crop_size: 256
resnet : 9 blocks

Pix2Pix :

- images fetched from directory : ./datasets/bau2nonbau

folder A: 100 RGB images of train, test and val images for Bauhaus style architecture (with same naming as corresponding folders in B eg: 1.jpg, 2.jpg etc)

folder B: 100 RGB images each , in train, test and val subfolders for non-bauhaus style architecture (Indian, Chinese, Greek, European, Modern and trending , Egyptian , Middle-eastern , Churches)

- images in folders A and B made into equal dimensions (256X256) in sneha_test_pix2pix.ipynb

- images in folders A and B paired into AB using ./datasets/combine_A_and_B.py

- final train, test and val paired images are thus obtained in folder AB

- using sneha_test_pix2pix.ipynb : train bidirectional(or as required) and test as required

Trained images originally stored under ./checkpoints/bau_pix2pix/web/images (I have trained Bauhaus to Non-Bauhaus direction (AB))

epoch:200
learning rate (lr) : 0.0002
batch_size: 1
beta1 : 0.5

Tested images originally stored under ./results/bau_pix2pix/test_latest/images (tested AB direction)

batch_size: 1
preprocessing : resize and crop
crop_size: 256