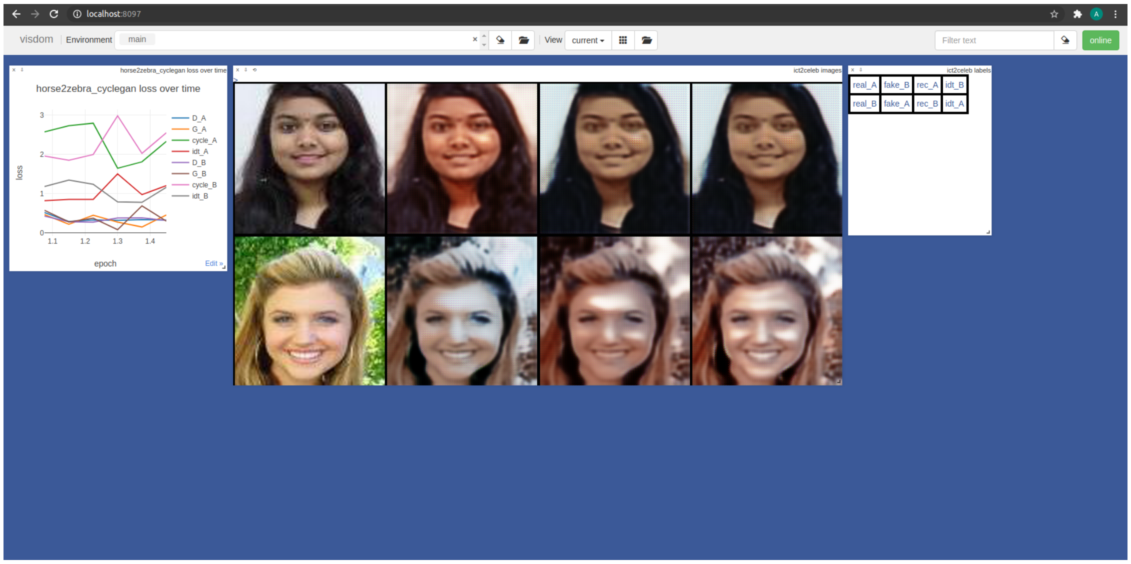
lab8 - report

March 18, 2021

1 Lab 8 - st121413

1.1 Visdom

The screenshot below is the prove that my visdom server is working.



real_A	fake_B	rec_A	idt_B
real_B	fake_A	rec_B	idt_A

Explaining the 8-grid pictures are as follow

- real_A: Original input photo
- fake_B: The result from generator A when input is real_A G_A(real_A)
- rec_A: The result reconstructing real_A back from fake_B G_B(G_A(real_A)) = G_B(fake_b)
- idt_B: The result from generator B when input is real A G_B(real_A)

Same case for second row

To get this lab runningm I just need to follow the instruction given in the lab manual

[1]:
```
# !git clone https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix.git
# !pip3 install dominate visdom
```

1.2 1. horse 2 zebra

This task is one of the example in the github. Follow the github tutorial or lab manual both fine.

1. Download the dataset

[2]:

```
# !cd pytorch-CycleGAN-and-pix2pix
# !./datasets/download_cyclegan_dataset.sh horse2zebra
```

Sneak peak at the downloaded dataset. It structured in this manner.

- datasets
  - horse2zebra
    - testA
    - testB
    - trainA
    - trainB

Folder *A contains picture of horse(s) and* B contains zebra/(s)

2. Train cycalGAN using the following comands

[3]:

```
# !cd pytorch-CycleGAN-and-pix2pix
# !unset http_proxy
# !unset https_proxy
# !nohup python3 -u train.py --dataroot ./datasets/horse2zebra --gpu_ids 0␣
 ↪--name horse2zebra_cyclegan --model cycle_gan > nohup_hourse2zebra.out
```

In the “server behind proxy” setup, we need to unset the proxy from the environment. If not, python code will try to ask the proxy for “http://localhost:8097” which unknown to the proxy server. (or find another way to escape http://localhost from proxy)

Below is what printed out from the training script.

[4]:

```
# ----------------- Options ---------------
#               batch_size: 1
#                    beta1: 0.5
#          checkpoints_dir: ./checkpoints
#           continue_train: False
#                crop_size: 256
#                 dataroot: ./datasets/horse2zebra                [default:␣
 ↪None]
#             dataset_mode: unaligned
#                direction: AtoB
#              display_env: main
#             display_freq: 400
#               display_id: 1
#            display_ncols: 4
#             display_port: 8097
#           display_server: http://localhost
#          display_winsize: 256
#                     epoch: latest
#               epoch_count: 1
#                  gan_mode: lsgan
#                   gpu_ids: 0
#                 init_gain: 0.02
#                 init_type: normal
#                  input_nc: 3
#                   isTrain: True                                 [default:␣
 ↪None]
#                  lambda_A: 10.0
#                  lambda_B: 10.0
#           lambda_identity: 0.5
#                 load_iter: 0                                    [default: 0]
#                 load_size: 286
#                        lr: 0.0002
#            lr_decay_iters: 50
#                 lr_policy: linear
#          max_dataset_size: inf
#                     model: cycle_gan
#                  n_epochs: 100
#            n_epochs_decay: 100
#                n_layers_D: 3
#                      name: horse2zebra_cyclegan                 [default:␣
 ↪experiment_name]
#                       ndf: 64
#                      netD: basic
#                      netG: resnet_9blocks
#                       ngf: 64
#                no_dropout: True
#                   no_flip: False
#                   no_html: False
#                      norm: instance
#               num_threads: 4
#                 output_nc: 3
#                     phase: train
#                 pool_size: 50
#                preprocess: resize_and_crop
#                print_freq: 100
#              save_by_iter: False
#           save_epoch_freq: 5
#          save_latest_freq: 5000
#            serial_batches: False
#                    suffix:
#          update_html_freq: 1000
#                   verbose: False
# ----------------- End -------------------
# dataset [UnalignedDataset] was created
# The number of training images = 1334
# initialize network with normal
# initialize network with normal
# initialize network with normal
# initialize network with normal
# model [CycleGANModel] was created
# ---------- Networks initialized -------------
# [Network G_A] Total number of parameters : 11.378 M
# [Network G_B] Total number of parameters : 11.378 M
# [Network D_A] Total number of parameters : 2.765 M
# [Network D_B] Total number of parameters : 2.765 M
# -----------------------------------------------
# /usr/lib/python3/dist-packages/requests/__init__.py:80:␣
 ↪RequestsDependencyWarning: urllib3 (1.26.3) or chardet (3.0.4) doesn't match␣
 ↪a supported version!
#   RequestsDependencyWarning)
# Setting up a new session...
# create web directory ./checkpoints/horse2zebra_cyclegan/web...
# /usr/local/lib/python3.6/dist-packages/torch/optim/lr_scheduler.py:82:␣
 ↪UserWarning: Detected call of `lr_scheduler.step()` before `optimizer.
 ↪step()`. In PyTorch 1.1.0 and later, you should call them in the opposite␣
 ↪order: `optimizer.step()` before `lr_scheduler.step()`.  Failure to do this␣
 ↪will result in PyTorch skipping the first value of the learning rate␣
 ↪schedule.See more details at https://pytorch.org/docs/stable/optim.
 ↪html#how-to-adjust-learning-rate
#   "https://pytorch.org/docs/stable/optim.html#how-to-adjust-learning-rate",␣
 ↪UserWarning)
# learning rate 0.0002000 -> 0.0002000
```

Below is an example report from the training script

[5]:

```
# (epoch: 1, iters: 100, time: 0.269, data: 0.223) D_A: 0.516 G_A: 0.465␣
 ↪cycle_A: 2.570 idt_A: 0.816 D_B: 0.425 G_B: 0.571 cycle_B: 1.955 idt_B: 1.
 ↪180
# (epoch: 1, iters: 200, time: 0.271, data: 0.001) D_A: 0.287 G_A: 0.218␣
 ↪cycle_A: 2.726 idt_A: 0.850 D_B: 0.280 G_B: 0.276 cycle_B: 1.849 idt_B: 1.
 ↪343
# (epoch: 1, iters: 300, time: 0.274, data: 0.001) D_A: 0.328 G_A: 0.445␣
 ↪cycle_A: 2.792 idt_A: 0.851 D_B: 0.273 G_B: 0.371 cycle_B: 1.994 idt_B: 1.
 ↪234
# (epoch: 1, iters: 400, time: 0.657, data: 0.001) D_A: 0.316 G_A: 0.272␣
 ↪cycle_A: 1.644 idt_A: 1.501 D_B: 0.377 G_B: 0.078 cycle_B: 2.981 idt_B: 0.
 ↪787
# (epoch: 1, iters: 500, time: 0.278, data: 0.001) D_A: 0.334 G_A: 0.147␣
 ↪cycle_A: 1.808 idt_A: 0.972 D_B: 0.379 G_B: 0.686 cycle_B: 2.022 idt_B: 0.
 ↪776
```

```
# (epoch: 1, iters: 600, time: 0.278, data: 0.001) D_A: 0.325 G_A: 0.454␣
 ↪cycle_A: 2.329 idt_A: 1.201 D_B: 0.315 G_B: 0.297 cycle_B: 2.550 idt_B: 1.
 ↪160
# (epoch: 1, iters: 700, time: 0.286, data: 0.001) D_A: 0.174 G_A: 0.388␣
 ↪cycle_A: 2.788 idt_A: 1.446 D_B: 0.179 G_B: 0.588 cycle_B: 3.010 idt_B: 1.
 ↪005
# (epoch: 1, iters: 800, time: 0.417, data: 0.001) D_A: 0.180 G_A: 0.372␣
 ↪cycle_A: 2.465 idt_A: 0.916 D_B: 0.316 G_B: 0.522 cycle_B: 1.902 idt_B: 0.
 ↪830
# (epoch: 1, iters: 900, time: 0.287, data: 0.001) D_A: 0.211 G_A: 0.320␣
 ↪cycle_A: 4.017 idt_A: 0.885 D_B: 0.225 G_B: 0.551 cycle_B: 1.917 idt_B: 1.
 ↪389
# (epoch: 1, iters: 1000, time: 0.286, data: 0.001) D_A: 0.251 G_A: 0.161␣
 ↪cycle_A: 2.847 idt_A: 1.350 D_B: 0.221 G_B: 0.436 cycle_B: 2.746 idt_B: 1.
 ↪292
# (epoch: 1, iters: 1100, time: 0.296, data: 0.001) D_A: 0.204 G_A: 0.548␣
 ↪cycle_A: 2.252 idt_A: 1.531 D_B: 0.284 G_B: 0.355 cycle_B: 3.095 idt_B: 1.
 ↪049
# (epoch: 1, iters: 1200, time: 0.423, data: 0.001) D_A: 0.284 G_A: 0.451␣
 ↪cycle_A: 2.065 idt_A: 0.945 D_B: 0.179 G_B: 0.347 cycle_B: 1.690 idt_B: 0.
 ↪872
# (epoch: 1, iters: 1300, time: 0.290, data: 0.001) D_A: 0.346 G_A: 0.379␣
 ↪cycle_A: 1.353 idt_A: 1.002 D_B: 0.345 G_B: 0.358 cycle_B: 2.060 idt_B: 0.
 ↪733
# End of epoch 1 / 200          Time Taken: 375 sec
# learning rate 0.0002000 -> 0.0002000
```

Each line report a loss. - D__A is loss of discriminator A - G__A is loss of Generator A - cycle__A is cycle loss of A - idt__a is identity loss of A

Same case for *__B

1.3 Result

[6]:

```
# (epoch: 200, iters: 34, time: 0.276, data: 0.001) D_A: 0.103 G_A: 0.532␣
 ↪cycle_A: 0.613 idt_A: 0.229 D_B: 0.107 G_B: 0.637 cycle_B: 0.747 idt_B: 0.
 ↪213
# (epoch: 200, iters: 134, time: 1.413, data: 0.001) D_A: 0.101 G_A: 0.464␣
 ↪cycle_A: 0.662 idt_A: 0.278 D_B: 0.064 G_B: 0.675 cycle_B: 0.571 idt_B: 0.
 ↪209
# (epoch: 200, iters: 234, time: 0.275, data: 0.001) D_A: 0.072 G_A: 0.561␣
 ↪cycle_A: 0.656 idt_A: 0.326 D_B: 0.145 G_B: 0.532 cycle_B: 0.889 idt_B: 0.
 ↪240
# (epoch: 200, iters: 334, time: 0.275, data: 0.001) D_A: 0.043 G_A: 0.461␣
 ↪cycle_A: 0.654 idt_A: 0.242 D_B: 0.090 G_B: 0.384 cycle_B: 0.556 idt_B: 0.
 ↪201
```

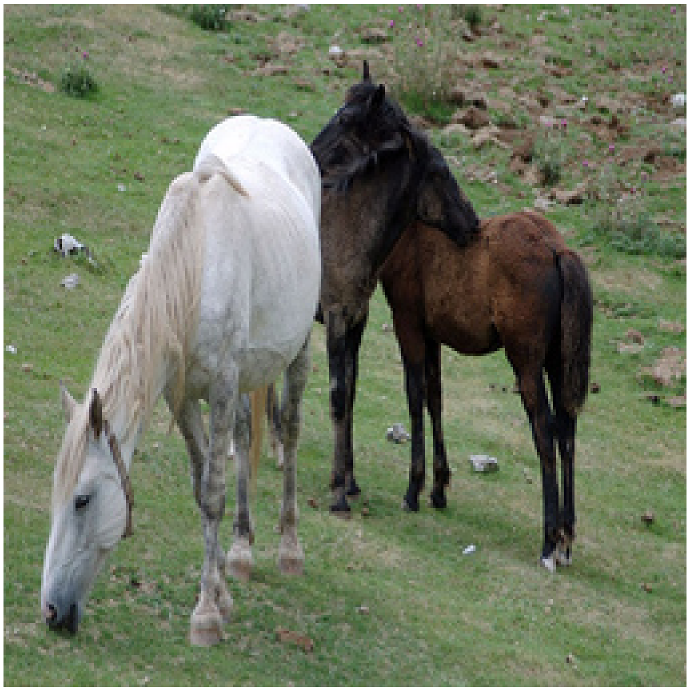
```
# (epoch: 200, iters: 434, time: 0.275, data: 0.001) D_A: 0.070 G_A: 0.636␣
↪cycle_A: 0.440 idt_A: 0.206 D_B: 0.100 G_B: 0.581 cycle_B: 0.573 idt_B: 0.
↪144
# (epoch: 200, iters: 534, time: 1.431, data: 0.001) D_A: 0.048 G_A: 0.647␣
↪cycle_A: 0.713 idt_A: 0.179 D_B: 0.290 G_B: 0.537 cycle_B: 0.579 idt_B: 0.
↪277
# (epoch: 200, iters: 634, time: 0.276, data: 0.001) D_A: 0.084 G_A: 0.475␣
↪cycle_A: 0.749 idt_A: 0.248 D_B: 0.063 G_B: 0.688 cycle_B: 0.782 idt_B: 0.
↪226
# (epoch: 200, iters: 734, time: 0.276, data: 0.001) D_A: 0.107 G_A: 0.455␣
↪cycle_A: 0.532 idt_A: 0.208 D_B: 0.104 G_B: 0.550 cycle_B: 0.770 idt_B: 0.
↪170
# (epoch: 200, iters: 834, time: 0.274, data: 0.001) D_A: 0.039 G_A: 0.676␣
↪cycle_A: 0.491 idt_A: 0.193 D_B: 0.258 G_B: 0.638 cycle_B: 0.652 idt_B: 0.
↪174
# (epoch: 200, iters: 934, time: 0.417, data: 0.001) D_A: 0.181 G_A: 0.551␣
↪cycle_A: 0.602 idt_A: 0.164 D_B: 0.188 G_B: 0.498 cycle_B: 0.526 idt_B: 0.
↪188
# (epoch: 200, iters: 1034, time: 0.275, data: 0.001) D_A: 0.134 G_A: 0.484␣
↪cycle_A: 0.635 idt_A: 0.192 D_B: 0.096 G_B: 0.544 cycle_B: 0.510 idt_B: 0.
↪228
# (epoch: 200, iters: 1134, time: 0.275, data: 0.001) D_A: 0.150 G_A: 1.109␣
↪cycle_A: 0.539 idt_A: 0.210 D_B: 0.095 G_B: 0.484 cycle_B: 0.482 idt_B: 0.
↪211
# (epoch: 200, iters: 1234, time: 0.275, data: 0.001) D_A: 0.036 G_A: 0.727␣
↪cycle_A: 0.538 idt_A: 0.240 D_B: 0.097 G_B: 0.597 cycle_B: 0.701 idt_B: 0.
↪138
# (epoch: 200, iters: 1334, time: 0.428, data: 0.001) D_A: 0.113 G_A: 0.469␣
↪cycle_A: 0.669 idt_A: 0.156 D_B: 0.230 G_B: 0.462 cycle_B: 0.512 idt_B: 0.
↪182
# End of epoch 200 / 200          Time Taken: 372 sec
```

[7]:

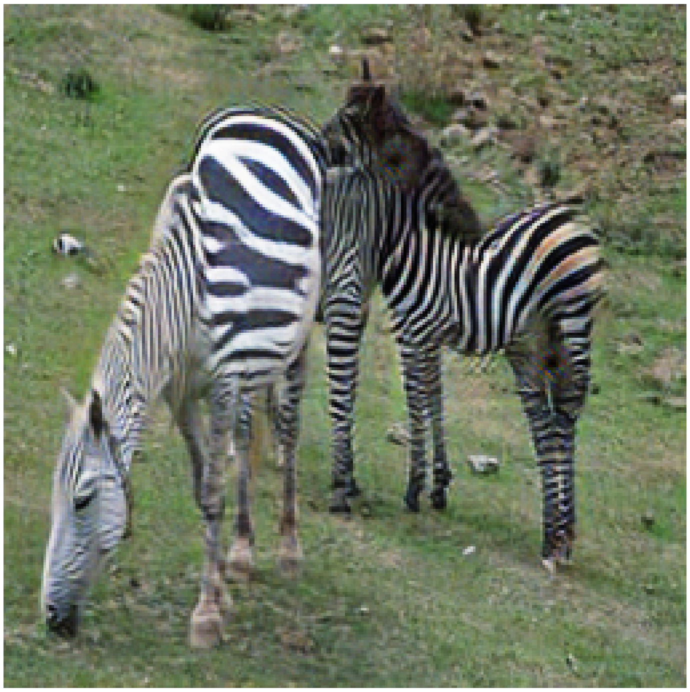
```
# $ python3 test.py --dataroot ./datasets/horse2zebra --gpu_ids 1 --name␣
↪horse2zebra_cyclegan --model cycle_gan
# ----------------- Options ---------------
#              aspect_ratio: 1.0
#                batch_size: 1
#           checkpoints_dir: ./checkpoints
#                 crop_size: 256
#                  dataroot: ./datasets/horse2zebra                [default:␣
↪None]
#              dataset_mode: unaligned
#                 direction: AtoB
#           display_winsize: 256
#                     epoch: latest
#                      eval: False
#                   gpu_ids: 1                                      [default: 0]
#                 init_gain: 0.02
#                 init_type: normal
#                  input_nc: 3
#                   isTrain: False                                  [default:␣
↪None]
#                 load_iter: 0                                      [default: 0]
#                 load_size: 256
#          max_dataset_size: inf
#                     model: cycle_gan                              [default:␣
↪test]
#                n_layers_D: 3
#                      name: horse2zebra_cyclegan                   [default:␣
↪experiment_name]
#                       ndf: 64
#                      netD: basic
#                      netG: resnet_9blocks
#                       ngf: 64
#                no_dropout: True
#                   no_flip: False
#                      norm: instance
#                  num_test: 50
#               num_threads: 4
#                 output_nc: 3
#                     phase: test
#                preprocess: resize_and_crop
#               results_dir: ./results/
#            serial_batches: False
#                    suffix:
#                   verbose: False
# ----------------- End -------------------
# dataset [UnalignedDataset] was created
# initialize network with normal
# initialize network with normal
# model [CycleGANModel] was created
# loading the model from ./checkpoints/horse2zebra_cyclegan/latest_net_G_A.pth
# loading the model from ./checkpoints/horse2zebra_cyclegan/latest_net_G_B.pth
# ---------- Networks initialized -------------
# [Network G_A] Total number of parameters : 11.378 M
# [Network G_B] Total number of parameters : 11.378 M
# -----------------------------------------------
# creating web directory ./results/horse2zebra_cyclegan/test_latest
# processing (0000)-th image... ['./datasets/horse2zebra/testA/n02381460_1000.
↪jpg']
# processing (0005)-th image... ['./datasets/horse2zebra/testA/n02381460_1110.
↪jpg']
```

```
# processing (0010)-th image... ['./datasets/horse2zebra/testA/n02381460_1260.jpg']
# processing (0015)-th image... ['./datasets/horse2zebra/testA/n02381460_1420.jpg']
# processing (0020)-th image... ['./datasets/horse2zebra/testA/n02381460_1690.jpg']
# processing (0025)-th image... ['./datasets/horse2zebra/testA/n02381460_1830.jpg']
# processing (0030)-th image... ['./datasets/horse2zebra/testA/n02381460_2050.jpg']
# processing (0035)-th image... ['./datasets/horse2zebra/testA/n02381460_2460.jpg']
# processing (0040)-th image... ['./datasets/horse2zebra/testA/n02381460_2870.jpg']
# processing (0045)-th image... ['./datasets/horse2zebra/testA/n02381460_3040.jpg']
```

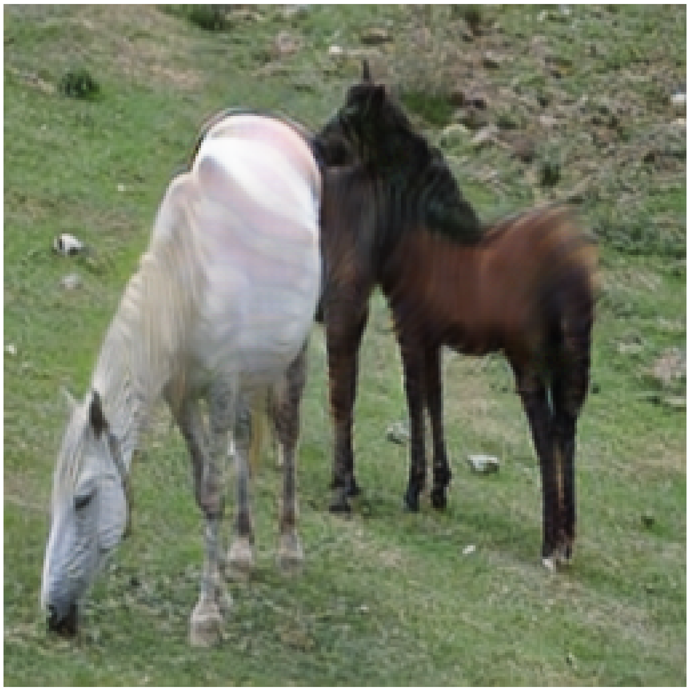
1.3.1 Real A



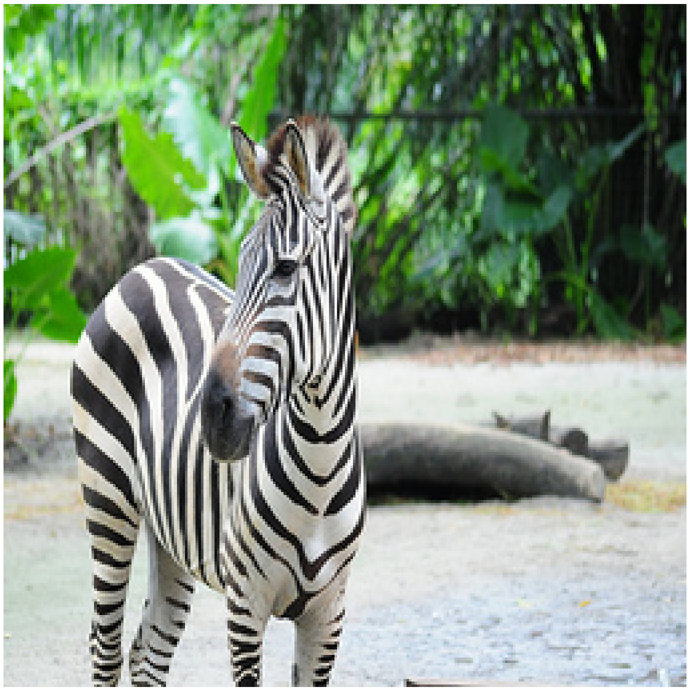
### G_A(Real_A)



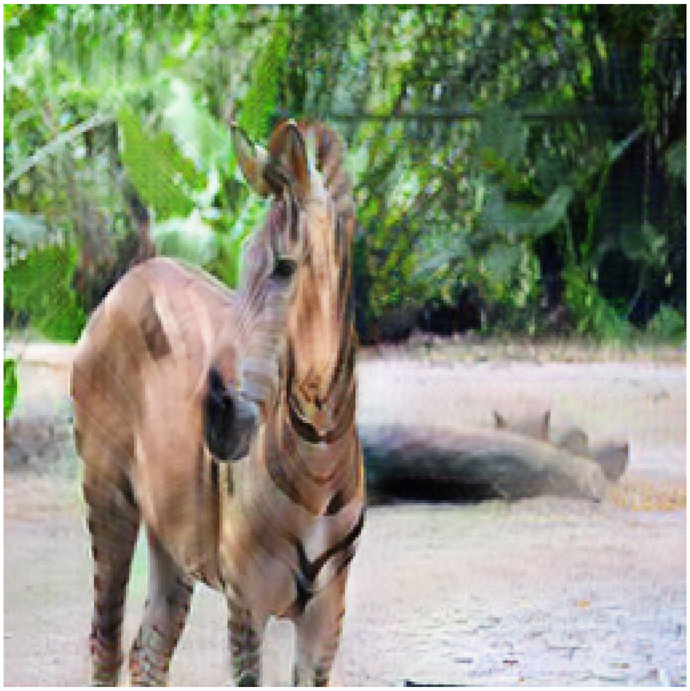
### G_B(G_A(Real_A))



1.3.2 Real B



### G_B(Real_B)



### G_A(G_B(Real_B))

1.4 2. AITICT 2 Celeb

The last updated dataset are given in the piazza

https://www.cs.ait.ac.th/~mdailey/celebA-317.zip

https://www.cs.ait.ac.th/~mdailey/ait-ict.zip

I created a folder int2celeb inside the datasets folder. The extracted ait-ict.zip is named trainA and trainB for the celebA-317.zip case. In a nutshell, here is how the folder is constructed.

- datasets
  - ict2caleb
    - trainA (ait-ict)
    - trainB (celebA-317)

The training commands are as follow.

[8]:
```
# unset http_proxy
# unset https_proxy
# nohup python3 -u train.py --dataroot ./datasets/ict2celeb --gpu_ids 1 --name␣
↪ict2celeb --model cycle_gan > nohup_ict2celeb.out
```

The last 4 epochs loss

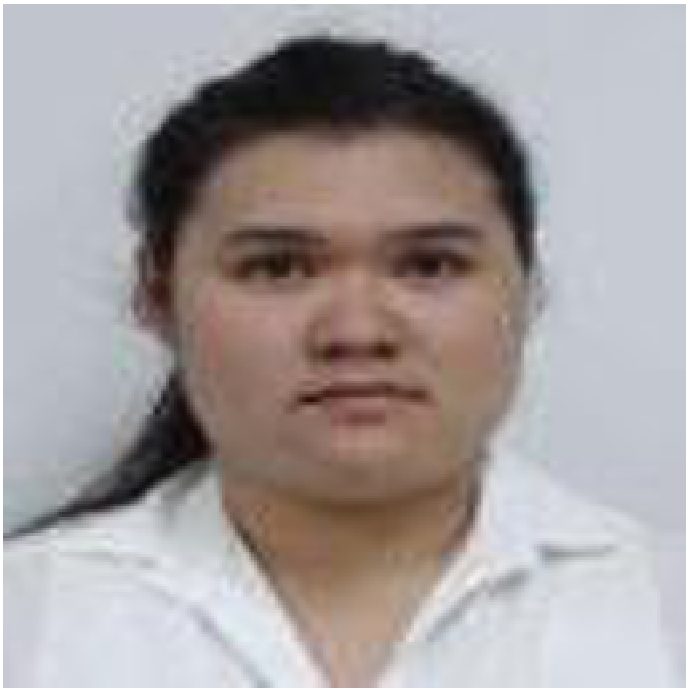
[9]:
```
# learning rate 0.0000079 -> 0.0000059
# (epoch: 197, iters: 68, time: 0.270, data: 0.001) D_A: 0.036 G_A: 0.923␣
↪cycle_A: 0.523 idt_A: 0.279 D_B: 0.060 G_B: 0.780 cycle_B: 0.808 idt_B: 0.
↪227
```

```
# (epoch: 197, iters: 168, time: 0.267, data: 0.001) D_A: 0.057 G_A: 0.855␣
 ↪cycle_A: 0.485 idt_A: 0.159 D_B: 0.126 G_B: 0.825 cycle_B: 0.494 idt_B: 0.
 ↪183
# (epoch: 197, iters: 268, time: 1.540, data: 0.001) D_A: 0.117 G_A: 0.734␣
 ↪cycle_A: 0.527 idt_A: 0.139 D_B: 0.076 G_B: 0.598 cycle_B: 0.395 idt_B: 0.
 ↪172
# End of epoch 197 / 200          Time Taken: 86 sec
# learning rate 0.0000059 -> 0.0000040
# (epoch: 198, iters: 51, time: 0.270, data: 0.001) D_A: 0.088 G_A: 0.745␣
 ↪cycle_A: 0.542 idt_A: 0.177 D_B: 0.052 G_B: 0.822 cycle_B: 0.517 idt_B: 0.
 ↪158
# (epoch: 198, iters: 151, time: 0.271, data: 0.001) D_A: 0.056 G_A: 0.921␣
 ↪cycle_A: 0.460 idt_A: 0.198 D_B: 0.088 G_B: 0.620 cycle_B: 0.524 idt_B: 0.
 ↪146
# (epoch: 198, iters: 251, time: 0.271, data: 0.001) D_A: 0.043 G_A: 0.769␣
 ↪cycle_A: 0.427 idt_A: 0.249 D_B: 0.174 G_B: 0.400 cycle_B: 0.736 idt_B: 0.
 ↪147
# End of epoch 198 / 200          Time Taken: 85 sec
# learning rate 0.0000040 -> 0.0000020
# (epoch: 199, iters: 34, time: 1.505, data: 0.001) D_A: 0.034 G_A: 0.949␣
 ↪cycle_A: 0.471 idt_A: 0.201 D_B: 0.083 G_B: 0.779 cycle_B: 0.504 idt_B: 0.
 ↪278
# (epoch: 199, iters: 134, time: 0.269, data: 0.001) D_A: 0.127 G_A: 0.469␣
 ↪cycle_A: 0.408 idt_A: 0.206 D_B: 0.086 G_B: 0.647 cycle_B: 0.564 idt_B: 0.
 ↪165
# (epoch: 199, iters: 234, time: 0.269, data: 0.001) D_A: 0.159 G_A: 0.871␣
 ↪cycle_A: 0.421 idt_A: 0.206 D_B: 0.091 G_B: 0.672 cycle_B: 0.500 idt_B: 0.
 ↪156
# End of epoch 199 / 200          Time Taken: 86 sec
# learning rate 0.0000020 -> 0.0000000
# (epoch: 200, iters: 17, time: 0.270, data: 0.001) D_A: 0.057 G_A: 0.704␣
 ↪cycle_A: 0.413 idt_A: 0.252 D_B: 0.049 G_B: 0.426 cycle_B: 0.597 idt_B: 0.
 ↪277
# (epoch: 200, iters: 117, time: 1.578, data: 0.001) D_A: 0.074 G_A: 0.770␣
 ↪cycle_A: 0.397 idt_A: 0.198 D_B: 0.062 G_B: 0.687 cycle_B: 0.584 idt_B: 0.
 ↪147
# (epoch: 200, iters: 217, time: 0.268, data: 0.001) D_A: 0.056 G_A: 0.837␣
 ↪cycle_A: 0.428 idt_A: 0.158 D_B: 0.090 G_B: 0.629 cycle_B: 0.467 idt_B: 0.
 ↪127
# (epoch: 200, iters: 317, time: 0.268, data: 0.001) D_A: 0.045 G_A: 0.867␣
 ↪cycle_A: 0.472 idt_A: 0.259 D_B: 0.119 G_B: 0.787 cycle_B: 0.806 idt_B: 0.
 ↪174
# saving the model at the end of epoch 200, iters 63400
# End of epoch 200 / 200          Time Taken: 89 sec
```

1.5 Result

```
[10]: # $ python3 test.py --dataroot ./datasets/ict2celeb --gpu_ids 1 --name␣
       ↪ict2celeb --model cycle_gan
      # ----------------- Options ---------------
      #              aspect_ratio: 1.0
      #                batch_size: 1
      #           checkpoints_dir: ./checkpoints
      #                 crop_size: 256
      #                  dataroot: ./datasets/ict2celeb                   [default:␣
       ↪None]
      #              dataset_mode: unaligned
      #                 direction: AtoB
      #           display_winsize: 256
      #                     epoch: latest
      #                      eval: False
      #                   gpu_ids: 1                                      [default: 0]
      #                 init_gain: 0.02
      #                 init_type: normal
      #                  input_nc: 3
      #                   isTrain: False                                  [default:␣
       ↪None]
      #                 load_iter: 0                                      [default: 0]
      #                 load_size: 256
      #          max_dataset_size: inf
      #                     model: cycle_gan                              [default:␣
       ↪test]
      #                n_layers_D: 3
      #                      name: ict2celeb                              [default:␣
       ↪experiment_name]
      #                       ndf: 64
      #                      netD: basic
      #                      netG: resnet_9blocks
      #                       ngf: 64
      #                no_dropout: True
      #                   no_flip: False
      #                      norm: instance
      #                  num_test: 50
      #               num_threads: 4
      #                 output_nc: 3
      #                     phase: test
      #                preprocess: resize_and_crop
      #               results_dir: ./results/
      #            serial_batches: False
      #                    suffix:
      #                   verbose: False
      # ----------------- End -------------------
# dataset [UnalignedDataset] was created
# initialize network with normal
# initialize network with normal
# model [CycleGANModel] was created
# loading the model from ./checkpoints/ict2celeb/latest_net_G_A.pth
# loading the model from ./checkpoints/ict2celeb/latest_net_G_B.pth
# ---------- Networks initialized -------------
# [Network G_A] Total number of parameters : 11.378 M
# [Network G_B] Total number of parameters : 11.378 M
# -----------------------------------------------
# creating web directory ./results/ict2celeb/test_latest
# processing (0000)-th image... ['./datasets/ict2celeb/testA/000003.jpg']
```

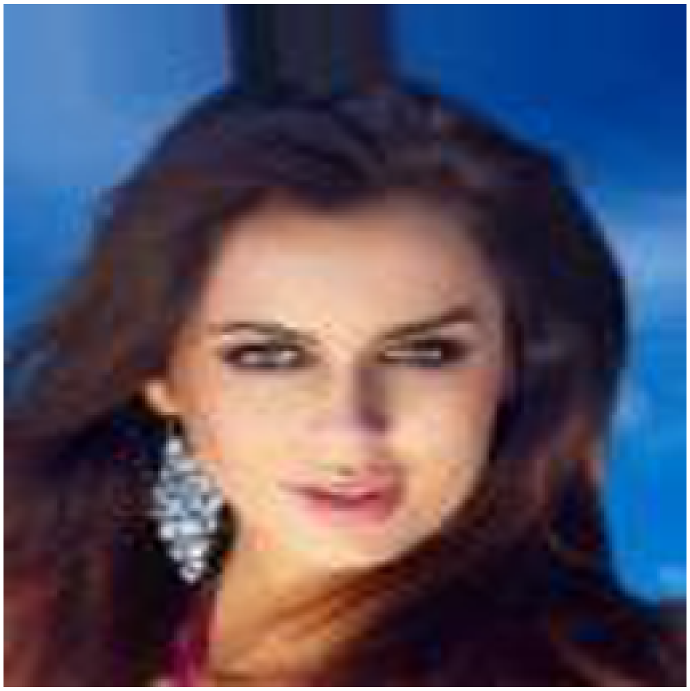
1.5.1 Real A



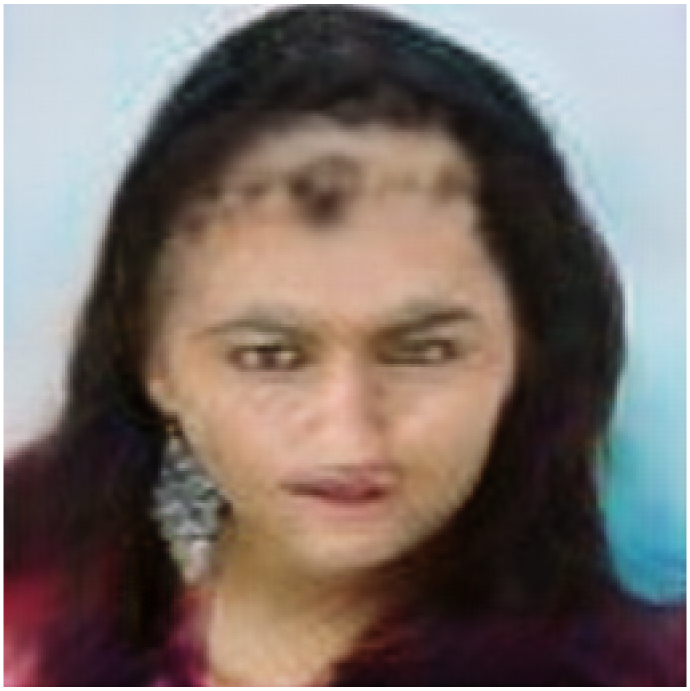
### G_A(Real_A)

### G_B(G_A(Real_A))

1.5.2 Real B



### G_B(Real_B)



### G_A(G_B(Real_B))