Generative Adversarial Nets GANs

- and how it’s used to generate art

Final project presentation

December 14, 2020

Team # 33

Team # 33

- Molika Meas: First year Master student (Software Engineering)

Team # 33

- Molika Meas: First year Master student (Software Engineering)

- Triinu Tasa: First year Master student (Data Science)

Team # 33

- Molika Meas: First year Master student (Software Engineering)
- Triinu Tasa: First year Master student (Data Science)
- Modar Sulaiman: First year PhD student (Computer Science)

Outline

1. Claude Monet
2. CycleGAN
3. Implementation
4. Evaluation
5. Lessons Learned

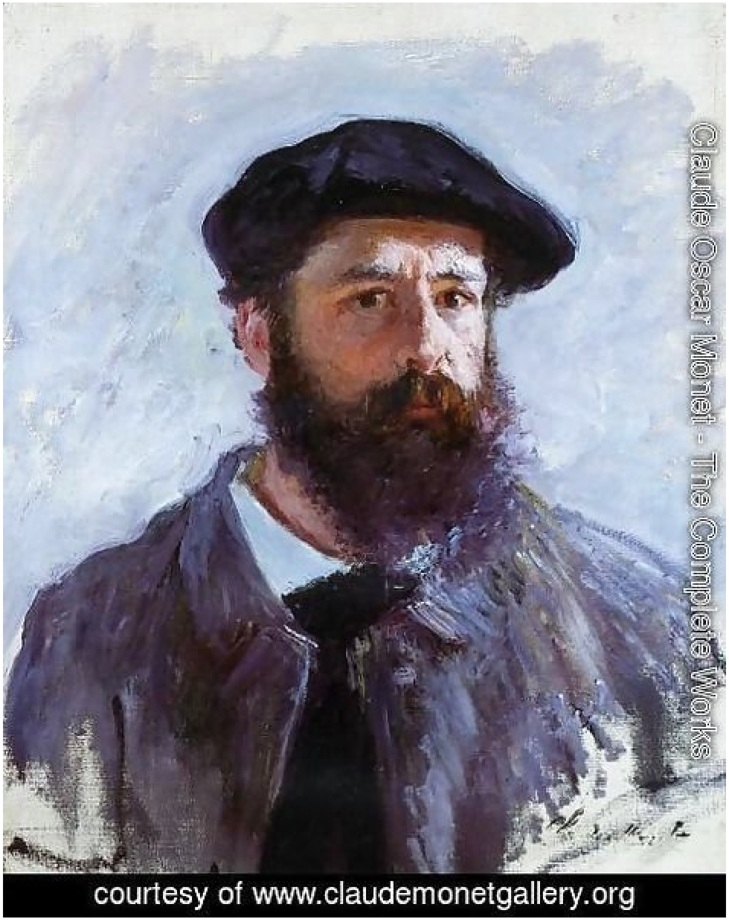
Claude Monet

- □ Oscar-Claude Monet was a French painter (1840 –1926)

- □ He is one of the founders of the impressionism (an art movement) along with his friends Renoir, Sisley and Bazille.

Reference: `https://www.claudemonetgallery.org/`

Style of Monet



courtesy of www.claudemonetgallery.org

Use GANs to create art

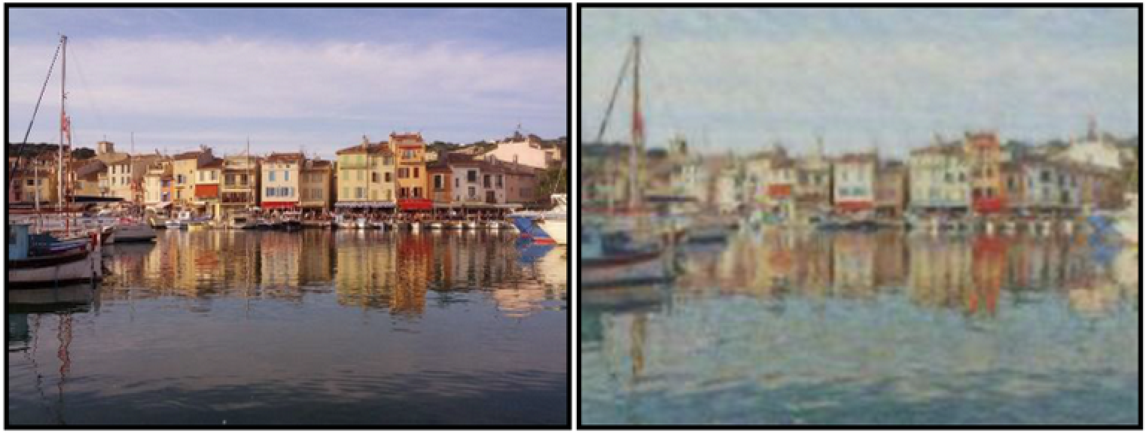


photo ⟶ Monet

Reference: (Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks) https://arxiv.org/pdf/1703.10593.pdf

CycleGAN

CycleGAN is a very popular GAN architecture primarily being used to learn transformation between images of different styles.

CycleGAN

CycleGAN is a very popular GAN architecture primarily being used to learn transformation between images of different styles.

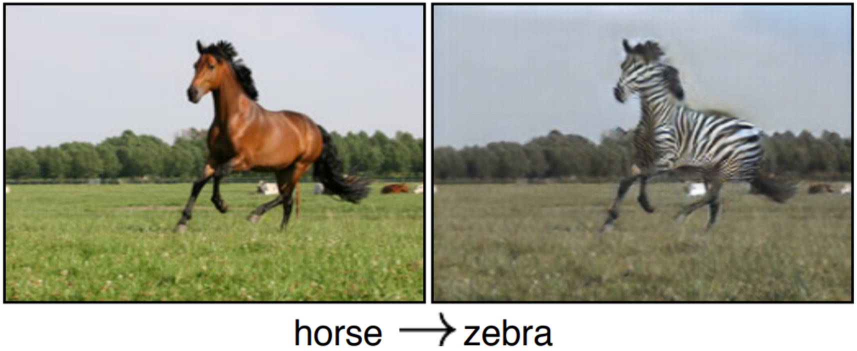
Unpaired Image-to-Image Translation.

CycleGAN

CycleGAN is a very popular GAN architecture primarily being used to learn transformation between images of different styles.

Unpaired Image-to-Image Translation.

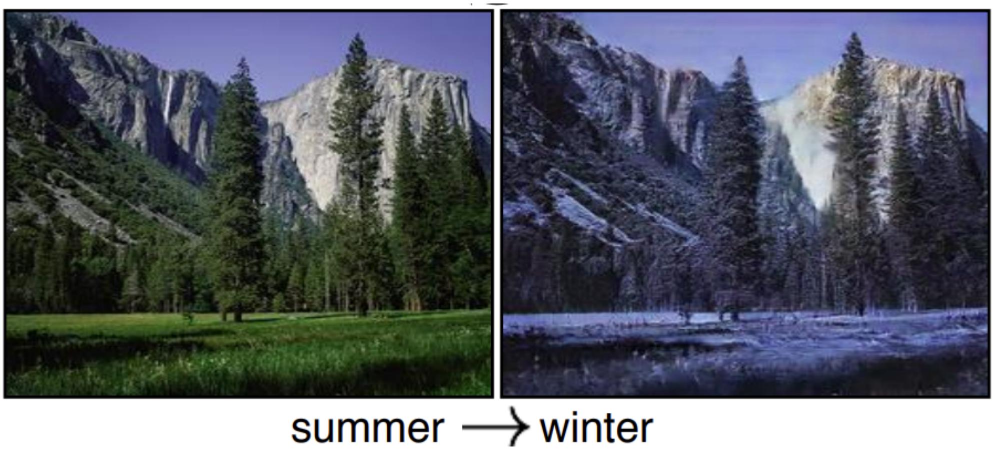
- As an example: a transformation between images of horse and zebra,



Reference: (Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks) https://arxiv.org/pdf/1703.10593.pdf

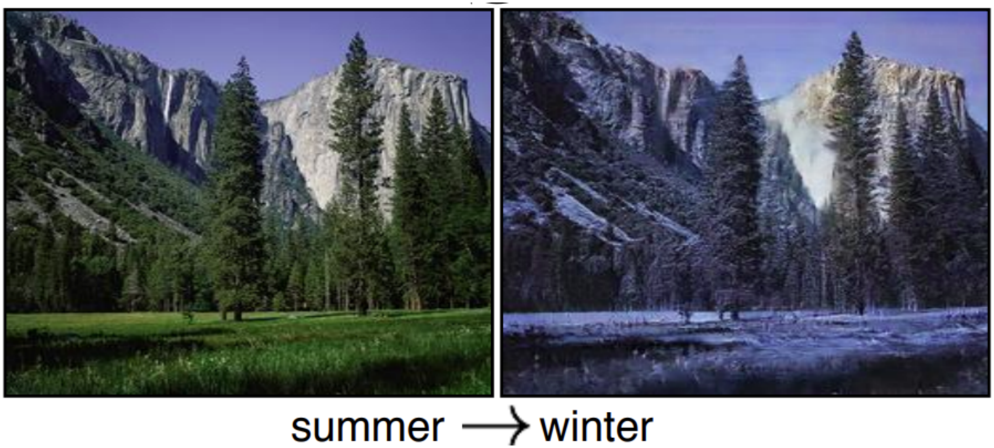
CycleGAN

- A transformation between winter image and summer image and so on.



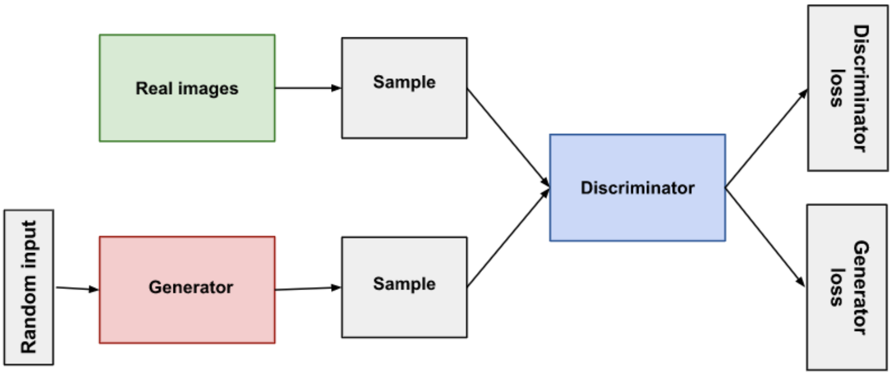
CycleGAN

- A transformation between winter image and summer image and so on.

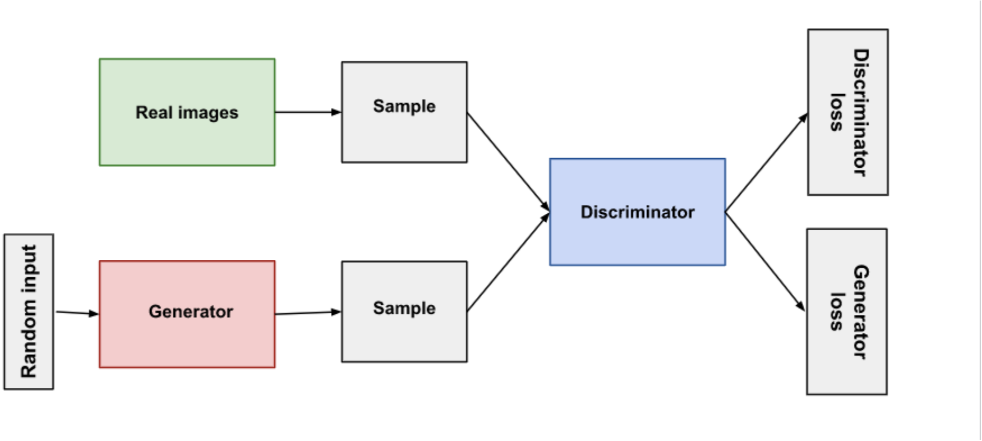


FaceApp is one of the most popular examples of CycleGAN where human faces are transformed into different age groups.

GAN Architecture - Part II

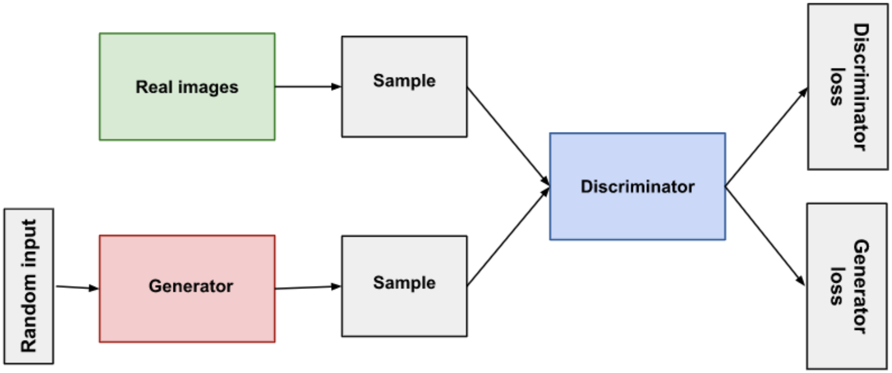


GAN Architecture - Part II



- G converts real images to Monet style painting and Dy is used to distinguish whether the image is real or generated.

GAN Architecture - Part II



- G converts real images to Monet style painting and Dy is used to distinguish whether the image is real or generated.
- CycleGAN builds 2 networks G and F to construct images from one domain to another (a real image to a Monet style picture) and in the reverse direction (a Monet style picture to a real image).

Cost function - Part III

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F),$$

Cost function - Part III

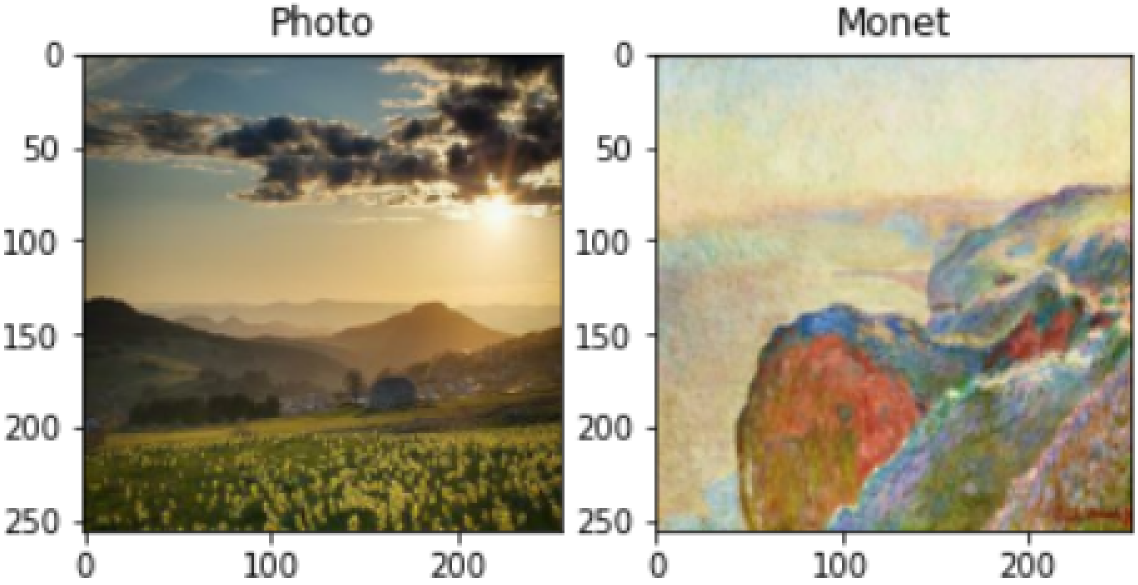
$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F),$$

- Cycle consistency loss which measures the L1-norm reconstruction cost for the real image and the Monet paintings.

Reference: (Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks) https://arxiv.org/pdf/1703.10593.pdf

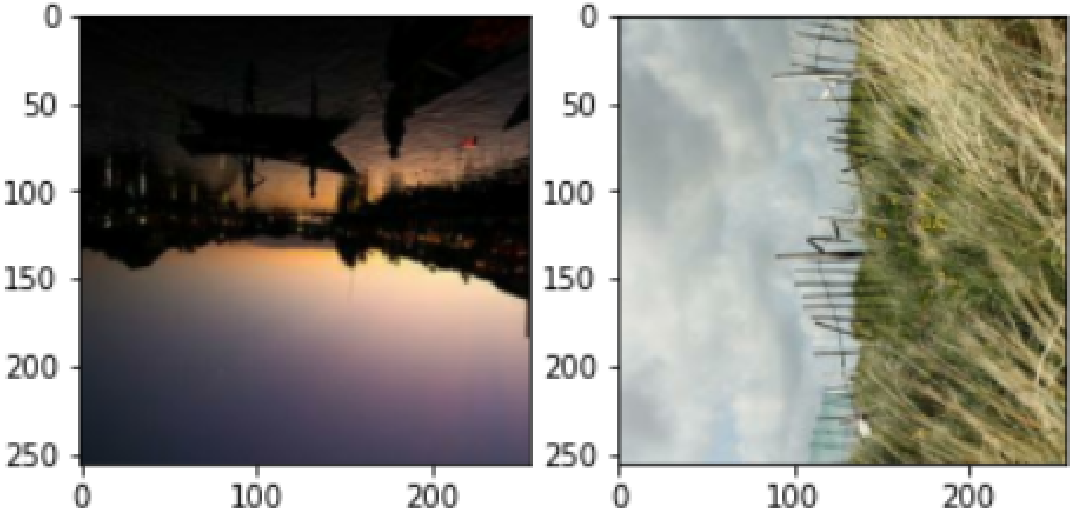
Pre-processing

Monet images: 300
Fake photos: 7038
(256 x 256 x 3)



Applying Augmentation

- Applied random jittering
- Applied random rotation (270º, 180º, 90º)
- Applied random mirroring (flipping left/right/up/down)



Building CycleGAN

- Downsampling
- Upsampling
- Build the generator
- Build the discriminator
- Define the discriminator and generator loss function (BinaryCrossentropy)
- Define the optimizer

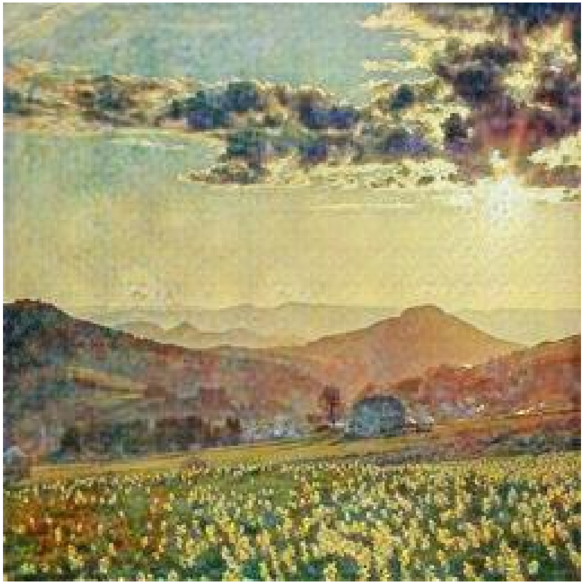
The architecture

- Generator has 16 CNNs (Conv2D and Conv2DTranspose), Dropout, instance normalization, 8 LeakyReLu activations and 7 ReLu activations
- Discriminator has 5 Conv2D, 3 Dropouts, Batch normalization, 4 LeakyReLu
- Total 54,414,979 trainable parameters

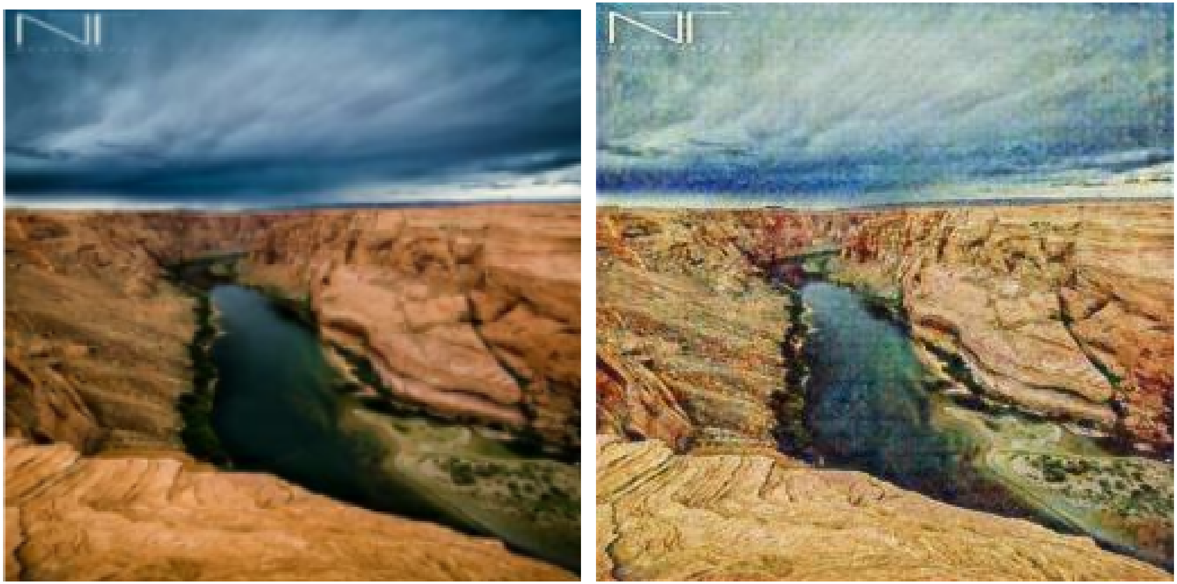
Training

- EPOCHS = 35
- Each epoch takes about 61s (with TPU)

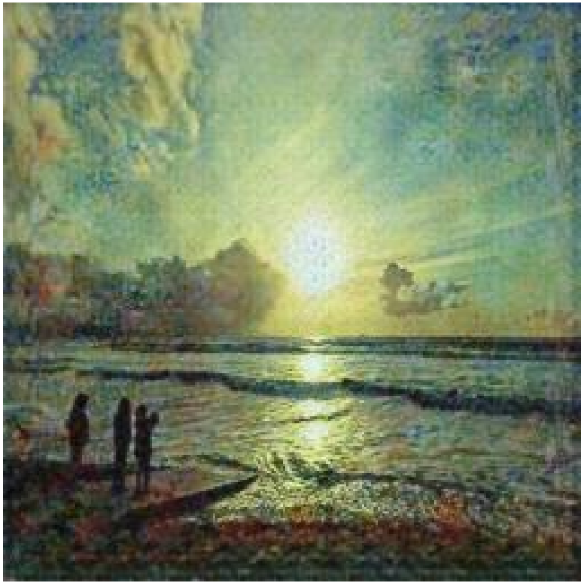
Generate the dataset



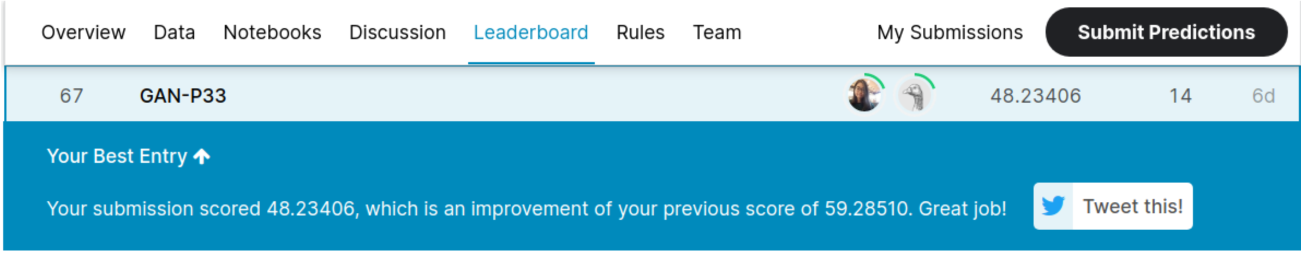
Generate the dataset



Generate the dataset



Evaluation



67	GAN-P33	48.23406	14	6d

Your Best Entry ↑

Your submission scored 48.23406, which is an improvement of your previous score of 59.28510. Great job!

- Augmentation does improve the performance
- More number of epochs helps to increase the performance but number of epochs too high can cause over-fitting

Lessons Learned

- Had fun exploring Kaggle competitions
- Learn about GANs and image processing techniques

Kernels used

Kernel used:
- https://www.kaggle.com/amyjang/monet-cyclegan-tutorial
- https://www.kaggle.com/dimitreoliveira/improving-cyclegan-monet-paintingsAugmentations
- https://www.kaggle.com/swepat/cyclegan-to-generate-monet-style-images

Our implementation:
- https://github.com/measmolika/GANs-P33

Thank You For Listening