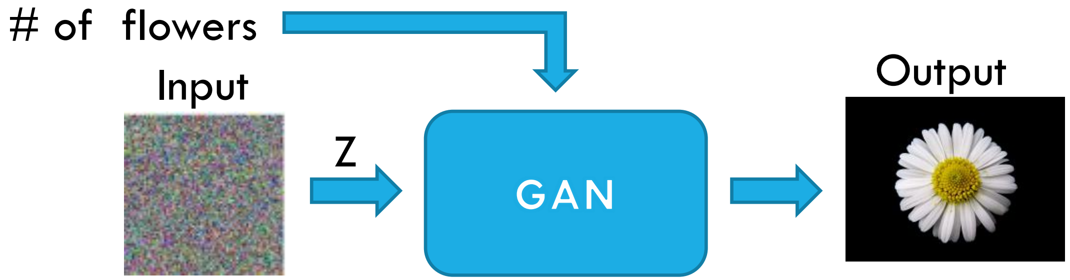
FLOWER IMAGE SYNTHESIS – DCGAN NETWORK

Ankur, Haitham, Rodolfo – 2020 Deep learning (University of Toronto)

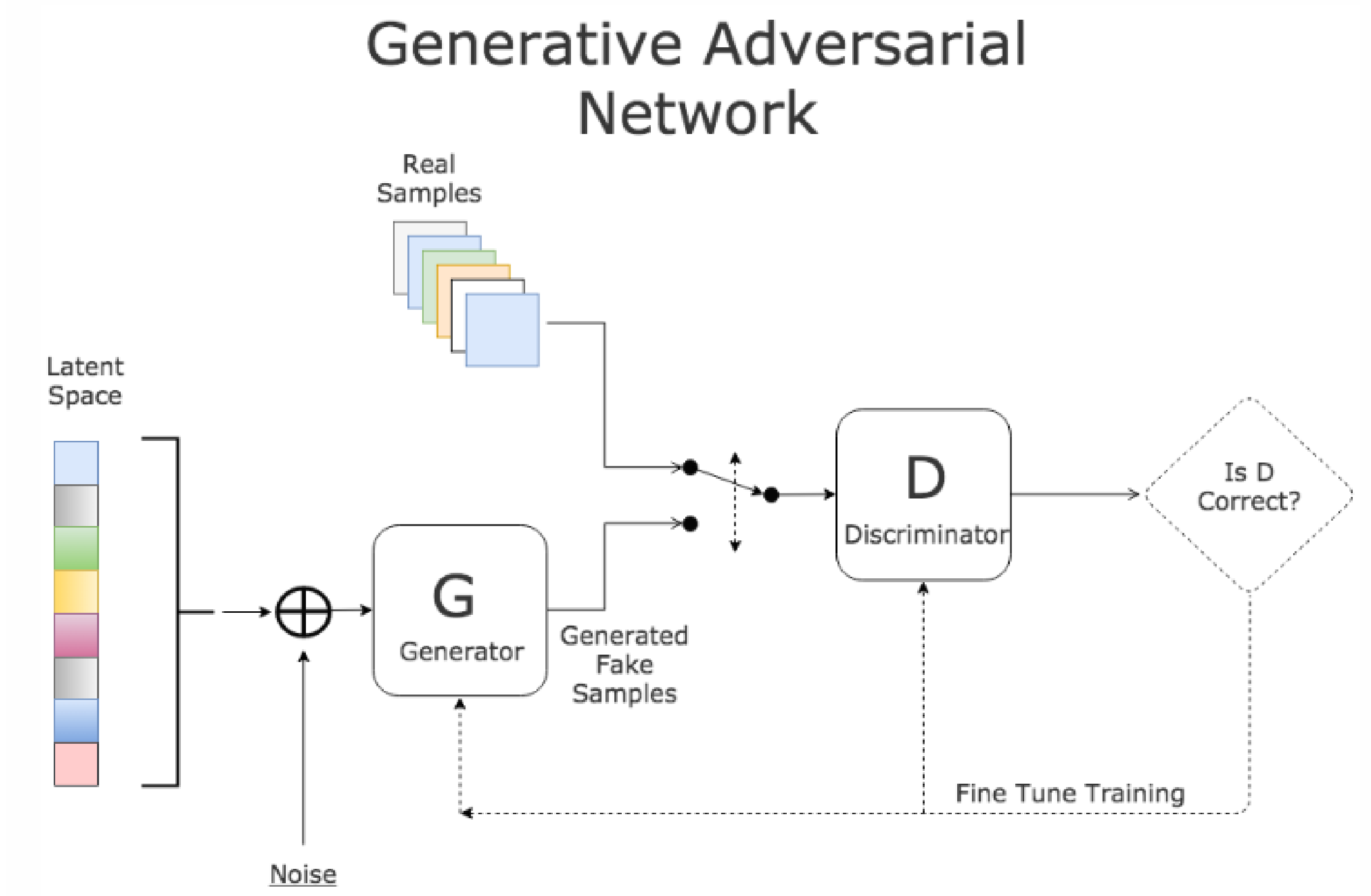
PROBLEM STATEMENT

To generate flowers from an input of a noise matrix which is trained on a set of flower images using the GAN's (Generative Adversarial Network).



Possible Applications :

1. Generate artwork
2. Image Synthesis



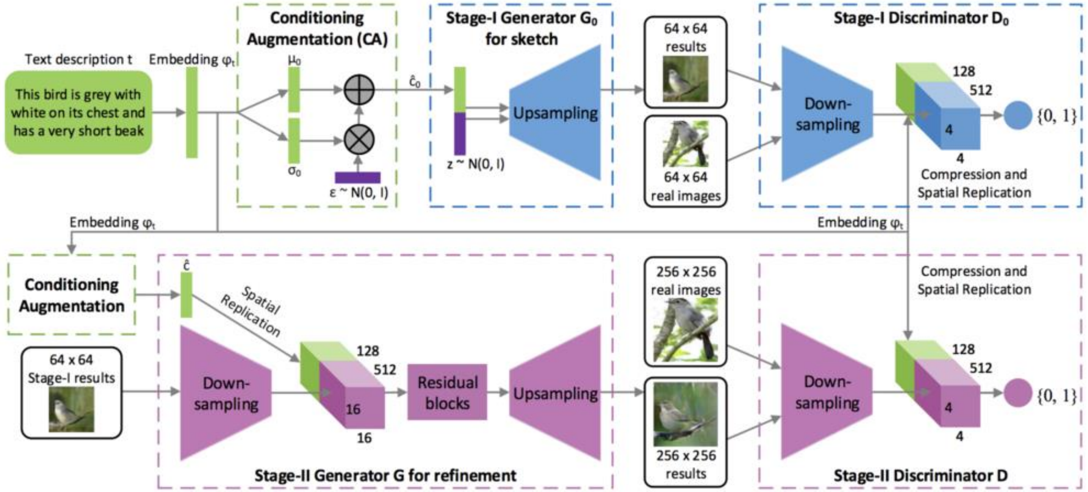
BACKGROUND - GAN

GAN is based on the approach of training 2 different networks

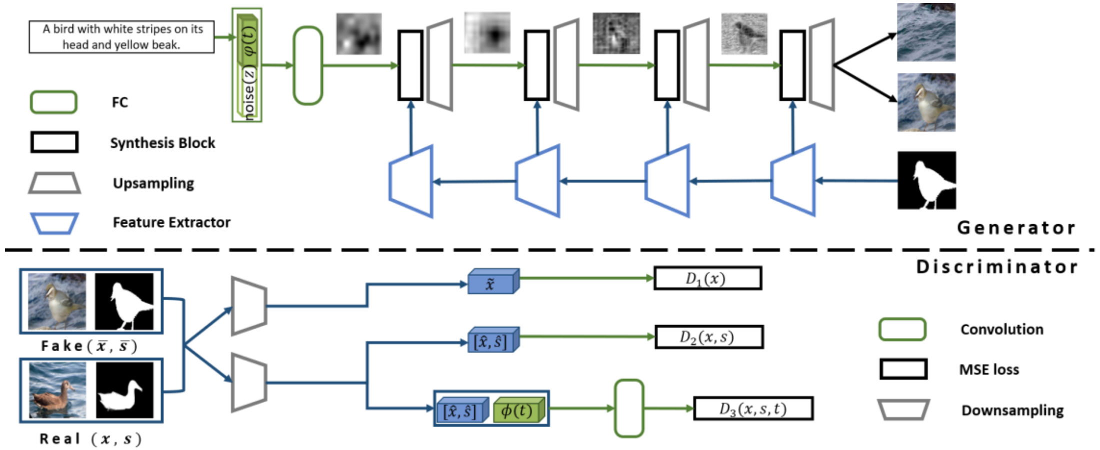
- Generator Network : Tries to generate realistic looking – samples
- Discriminator Network : Tries to figure out whether an image came from the training set or the generator network.

Objective : The generator tries to fool the discriminator network in order to predict that the image came from training set (when in reality was generated by Generator)

DIFFERENT GAN'S



Stack GAN – Used for Text to image synthesis



MC – GAN – Used for image generation from text attributes

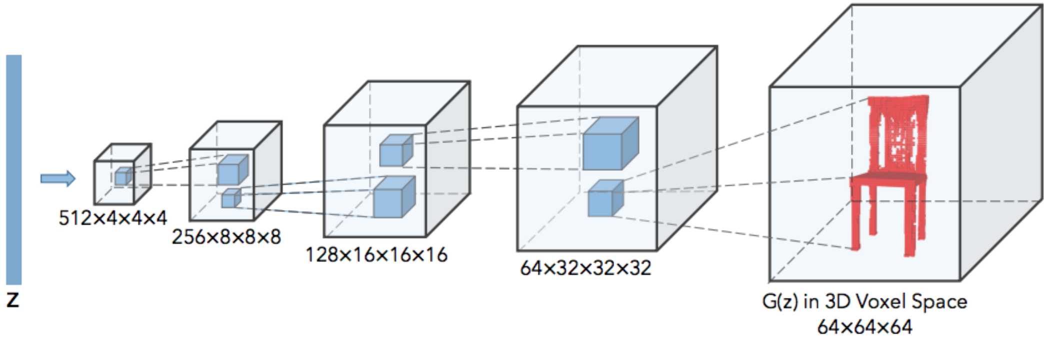


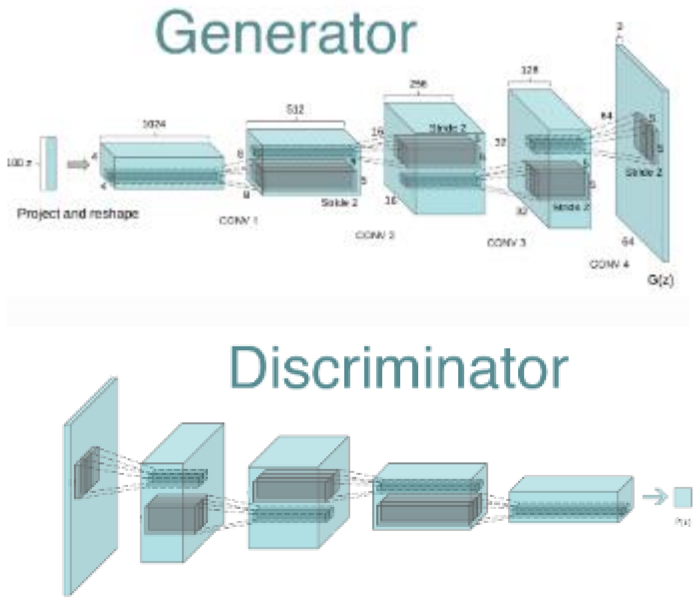
Figure 1: The generator in 3D-GAN. The discriminator mostly mirrors the generator.

3D GAN – Generation of 3D object from images

DCGAN NETWORK

DCGAN came into existence because CNN networks were not able to scale up the image generation process.
Core approach (3 changes to CNN architecture):

1. All convolution net functions. (convolution stride and transposed convolution for the down-sampling and the up-sampling)
2. No fully connected layer or max pooling layers
3. Batch Normalization : Stabilizes learning by normalizing the input to each unit to have zero mean and unit variance. (no batch normalization for output layer in generation and input layer in discriminator)



Additional notes
a. Use ReLU activation in generator for all layers except for the output, which uses Tanh.
b. Use LeakyReLU activation in the discriminator for all layers.

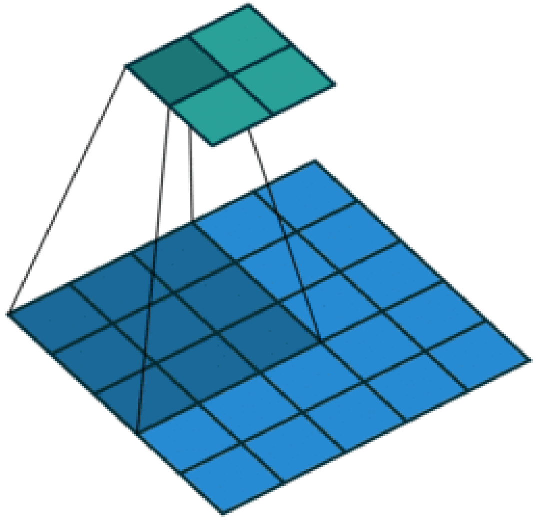
STRIDES / PADDING

Padding : The amount of pixels added to an image when it is being processed by the kernel of a CNN.

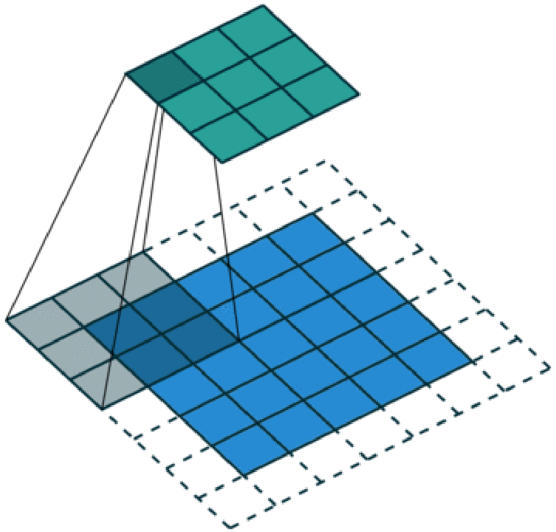
Stride : It is the number of pixels shifts over the input matrix. The value controls how the filter convolves around the input volume.

By default, the filter convolves around the input volume by shifting one unit at a time.

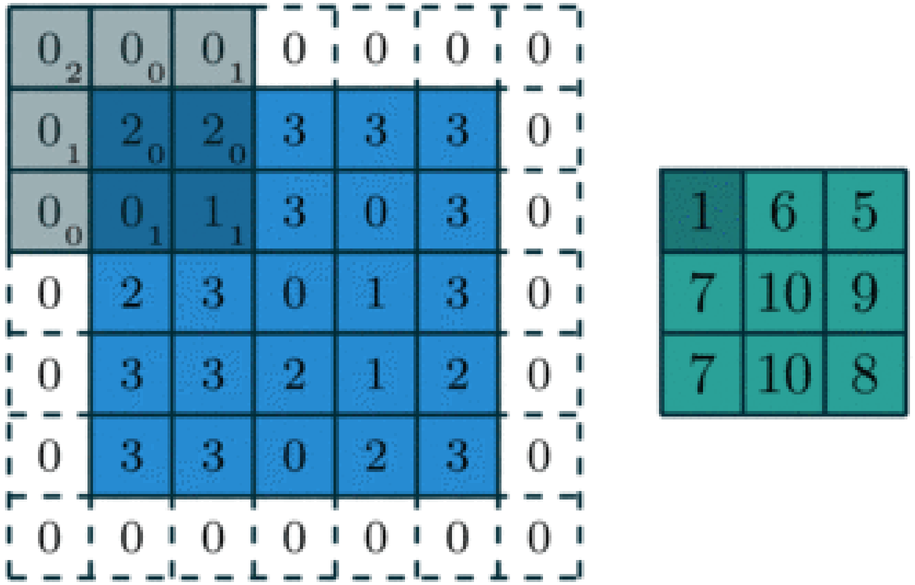
No – Padding / 1- Stride



1 – Padding / 1- Stride



1 – Padding / 2- Stride



LOSS FUNCTION

Loss Function : Discriminator in GAN uses a cross entropy loss, since discriminators job is to classify; cross entropy loss is the best one out there.

Below formula represents the cross entropy loss between p: the true distribution and q: the estimated distribution.

$$H(p, q) = -\sum_{i} p_i \log(q_i)$$

The intention of the loss function is to push the predictions of the real image towards 1 and the fake images to 0. We do so by log probability term.

In GAN, discriminator is a binary classifier. It needs to classify either the data is real or fake

IMAGE DATA PRE-PROCESSING

- Resizing pictures to 256x256 pixels.
- Image Normalization : Convert from 256 to 0 to 1.
- Data augmentation techniques were used to increase the no. of images required for training the dataset.

Original Image

Horizontal Flip Image

Vertical Flip Image

Random Rotation Image

Horizontal and Vertical Flip Image

The Journey to find the best Model

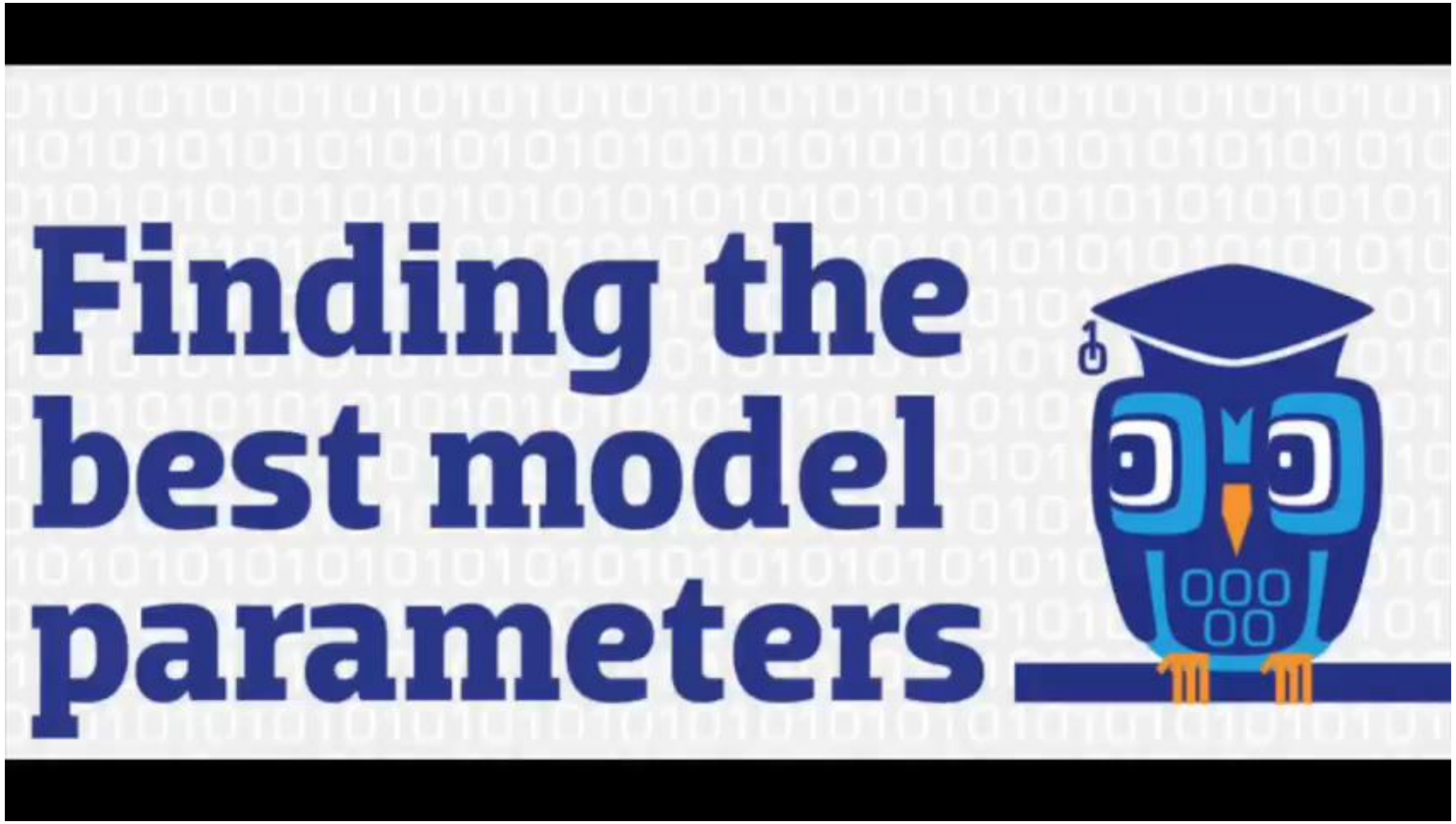
Model 1. Model 2. Model 3 Model 4 Model5

- Batch size.

- Nosie Array.

- Generator and Discriminator network.

MODEL ENHANCEMENT JOURNEY



MODEL COMPARISONS

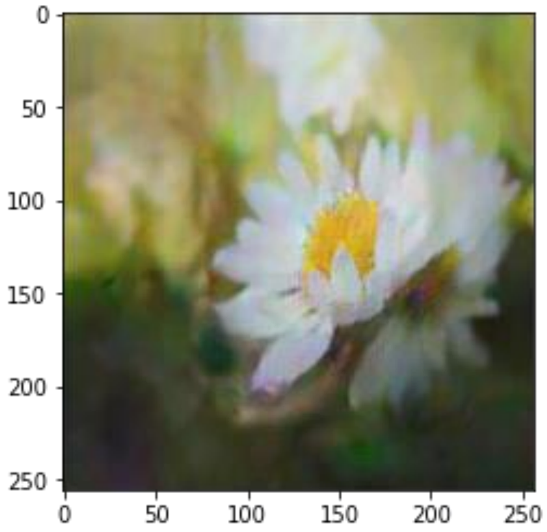
S.No.	Model Name	Batch size	Noise Array	Generator	Discriminator	LR_D	LR_G	Comments
1	Model 1	64	100	Initial Shape : 8,8,512 Layers : 6 Conv2D Transpose Layers	Layers: 5 Conv2D Layers	0.0001	0.0001	
2	Model 2	64	100	Initial Shape : 64, 64,256 Layers : 3 Conv2D Transpose Layer	Layers: 2 Conv2D Layers	0.0001	0.0001	
3	Model 3	64	1000	Initial Shape : 8,8,512 Layers : 6 Conv2D Transpose Layers	Layers: 5 Conv2D Layers	0.0001	0.0001	
4	Model 4	64	100	Initial Shape : 64, 64,256 Layers : 3 Conv2D Transpose Layer	Layers: 2 Conv2D Layers	0.0002	0.0002	Use batchnorm in both the generator and the discriminator. Use ReLU activation in generator for all layers except for the output, which uses Tanh. Use LeakyReLU activation in the discriminator for all layers.
5	Model 5	128	100	Initial Shape : 16, 16,1024 Layers : 4 Conv2D Transpose Layer	Layers: 3 Conv2D Layers	0.0002	0.0002	
6	Model 6	64	100	Initial Shape : 32, 32,512 Layers : 3 Conv2D Transpose Layer	Layers: 2 Conv2D Layers	0.0002	0.0002	

HOW TO EVALUATE GAN'S

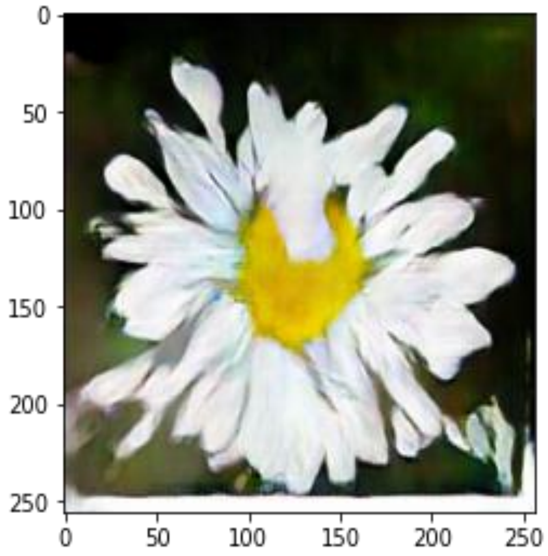
GAN's lack an objective function, which makes it difficult to compare performance of different models.

Options:

- Manual inspection of generated images.
- Qualitative evaluation.
- Quantitative evaluation.



Which one is better?



FUTURE WORK

- Train with different species of flowers
- Translation of style
  - generate cross species flowers
  - day to night
  - black and white pictures to color pictures
- Measuring the accuracy with Quantitative Evaluation methods combined with Qualitative Evaluation instead of Manual Inspection

RECOMMENDATIONS

Recommended platform for running GAN code/dataset/checkpoints.

- Google Colab Pro
  - Connected for up to 24h instead 12h
  - Priority to high-memory VMs with twice CPUs
  - Faster GPUs
  - Problem with Session connection for long training
- Colab Pro with Google Drive for storing checkpoints
  - The easiest setup
- Google Cloud AI Platform Notebooks with GPU
  - Slower than Colab Pro
- Colab Pro with Google Clould Bucket for storage was the fastest and more efficient method. (Best combination)

PROJECT DETAILS

- Project on GitHub:
  - https://github.com/ravasconcelos/flowers_dcgan
- YouTube:
  - https://www.youtube.com/watch?v=0uTfwXIWI40

REFERENCES

- Deep Convolutional Generative Adversarial Network
  - https://www.tensorflow.org/tutorials/generative/dcgan
- Flowers Recognition Kaggle Dataset
  - https://www.kaggle.com/alxmamaev/flowers-recognition
- How to Evaluate Generative Adversarial Networks
  - https://machinelearningmastery.com/how-to-evaluate-generative-adversarial-networks/
- Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks
  - https://arxiv.org/pdf/1511.06434.pdf%C3%AF%C2%BC%E2%80%B0