Model 1

https://github.com/ravasconcelos/flowers_dcgan/blob/master/dcgan_flowers_1.ipynb

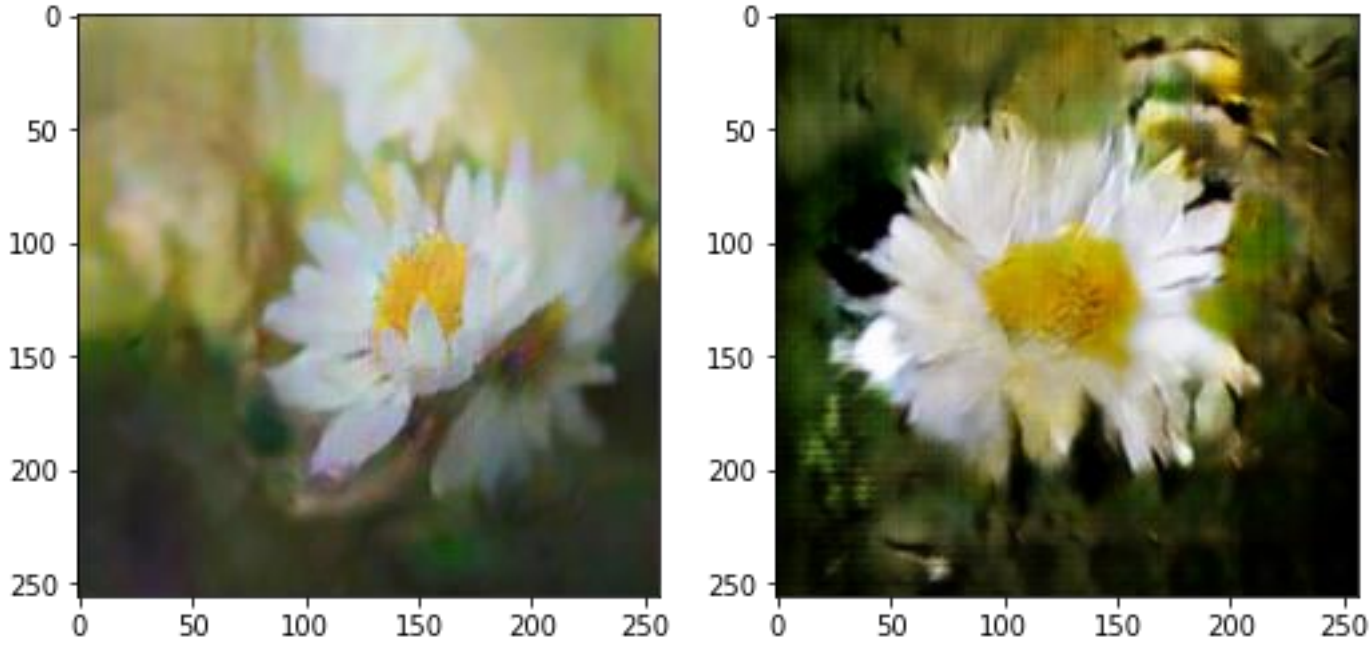
This model was based on the model of Assignment 4, but one more convolutional layer was added to the Generator and the Discriminator: https://github.com/gsurma/image_generator/blob/master/ImageGeneratorDCGAN.ipynb

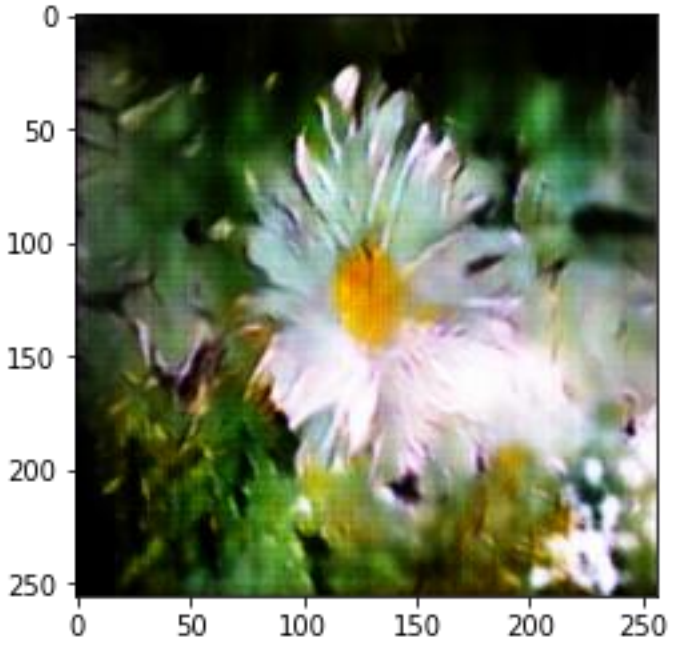
Batch Size: 64

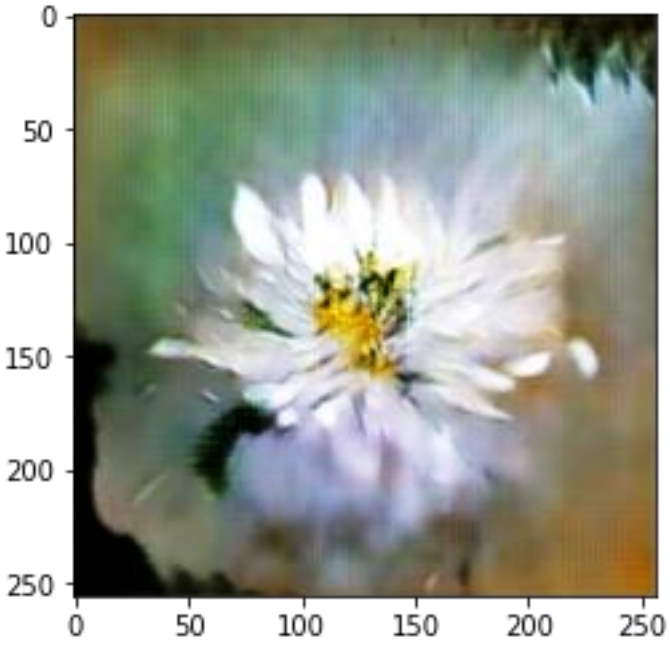
Noise Array: 100

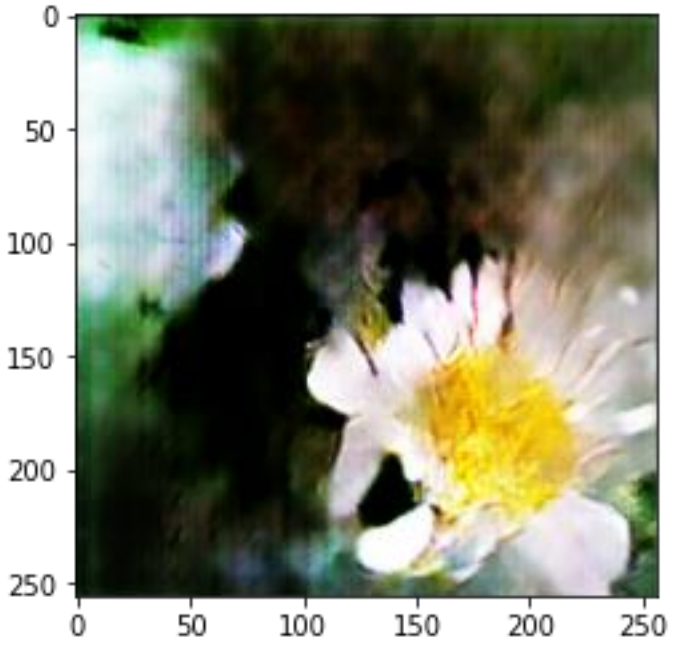
Generator: 6 Conv2DTranspose layers, initial shape 8, 8, 512

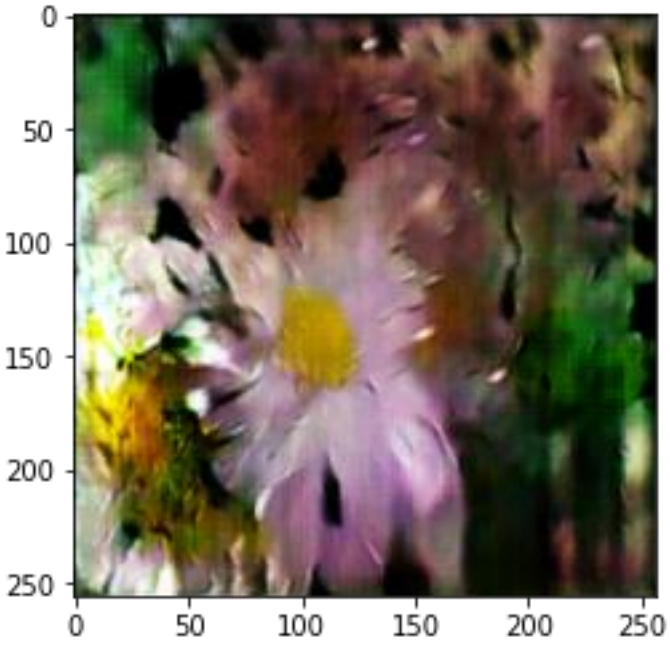
Discriminator: 5 Conv2D layers

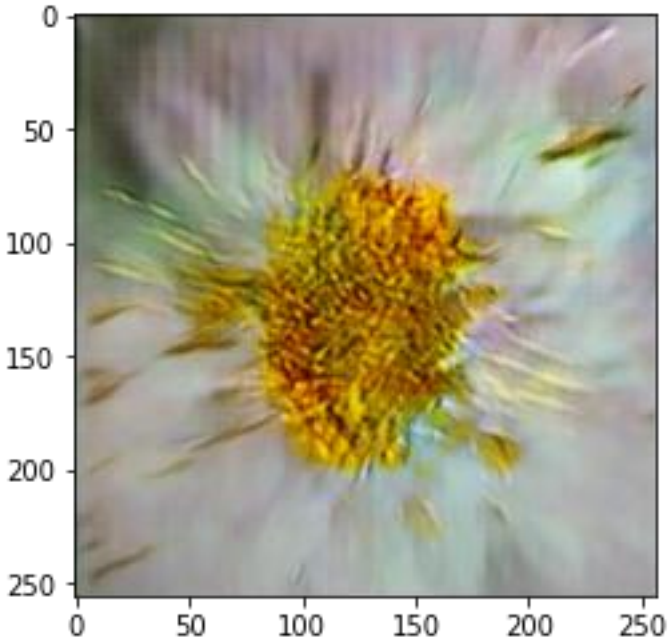
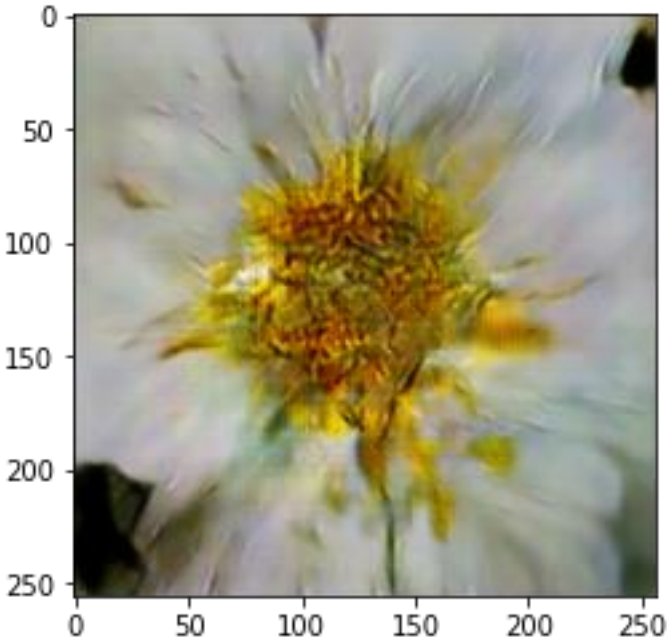



0
50
100
150
200
250
0
50
100
150
200
250


0
50
100
150
200
250
0
50
100
150
200
250


0
50
100
150
200
250
0
50
100
150
200
250


0
50
100
150
200
250
0
50
100
150
200
250


0
50
100
150
200
250
0
50
100
150
200
250

0
50
100
150
200
250
0
50
100
150
200
250

Model 2

https://github.com/ravasconcelos/flowers_dcgan/blob/master/dcgan_flowers_2.ipynb

This model was based on Tensorflow DCGAN tutorial: https://www.tensorflow.org/tutorials/generative/dcgan

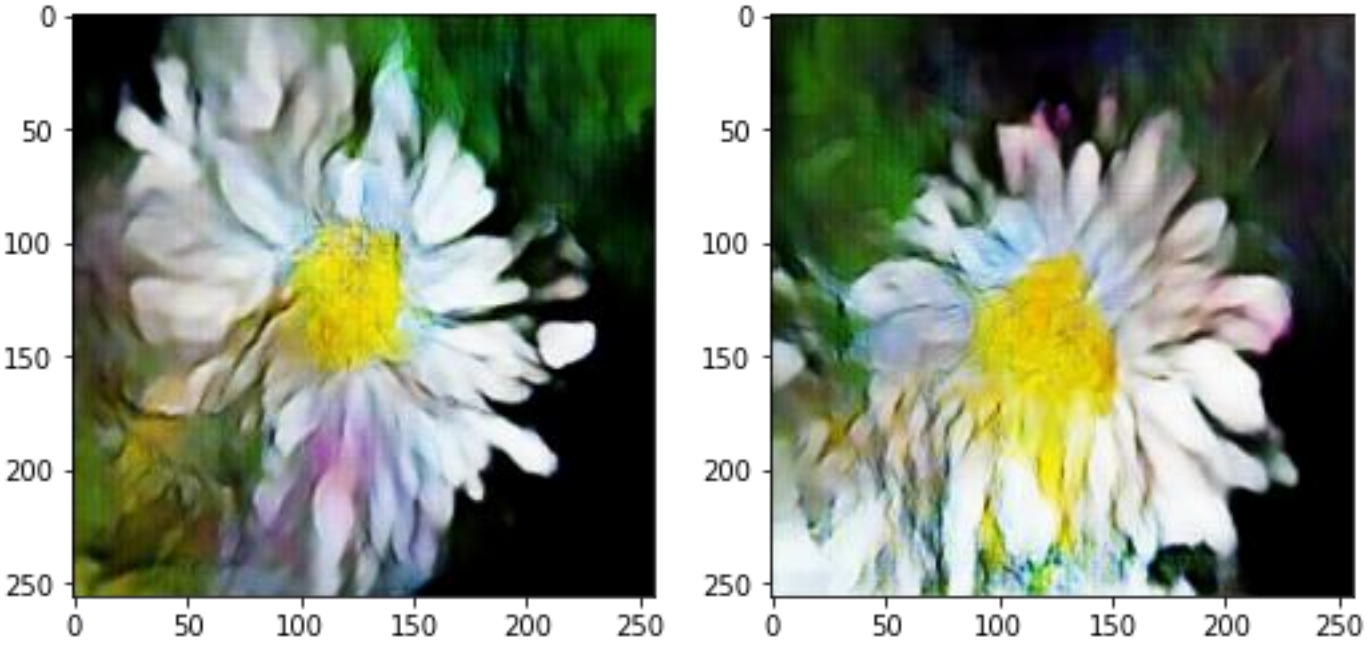
Batch Size: 64

Noise Array: 100

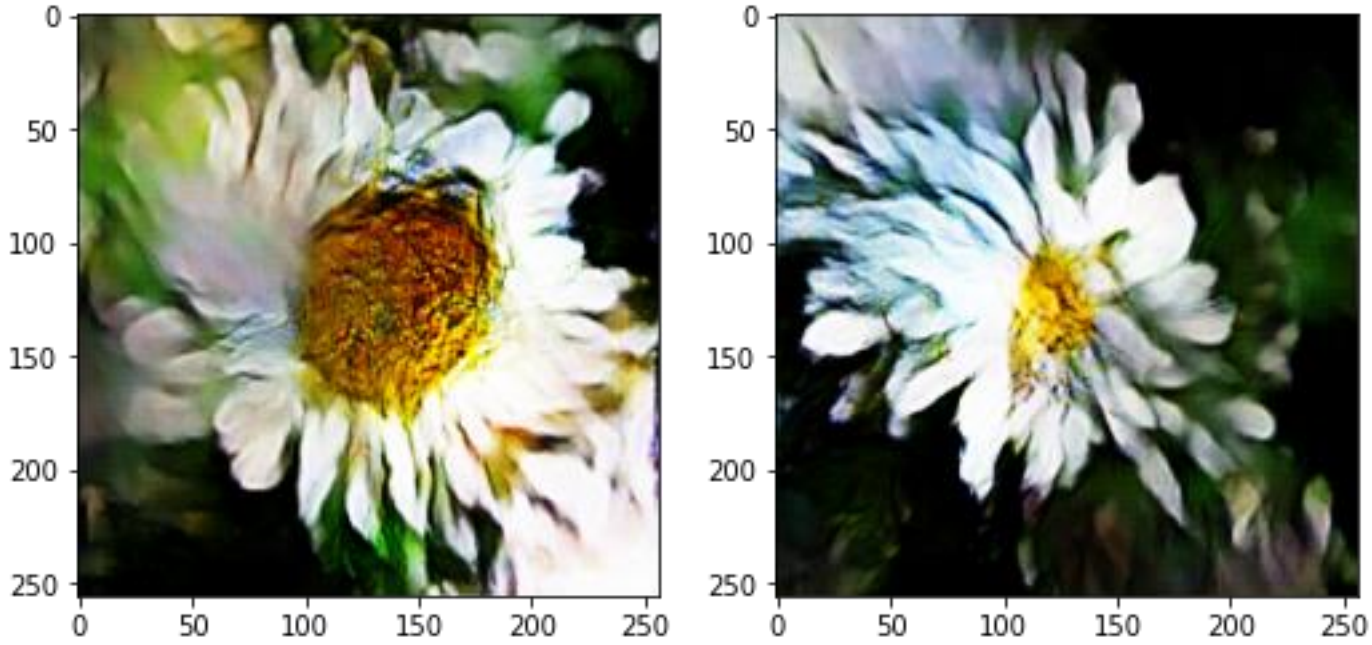
Generator: 3 Conv2DTranspose layers, initial shape 64, 64, 256

Discriminator: 2 Conv2D layers

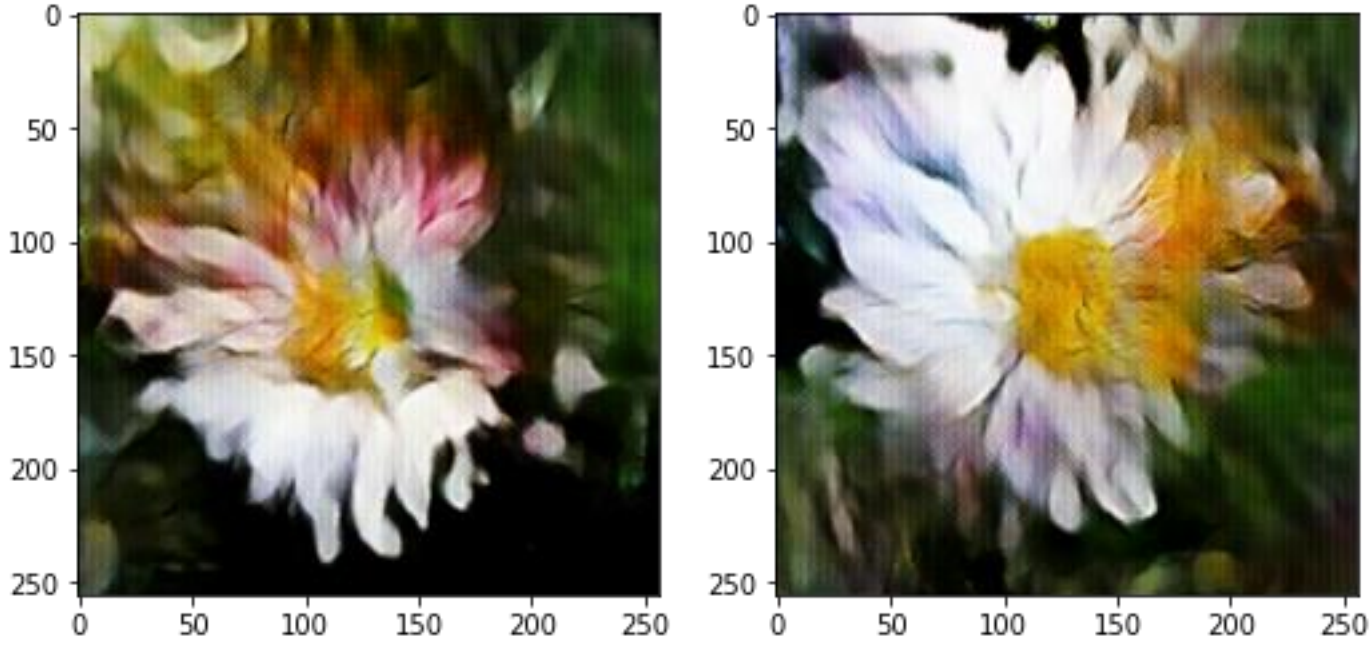
2000 epochs



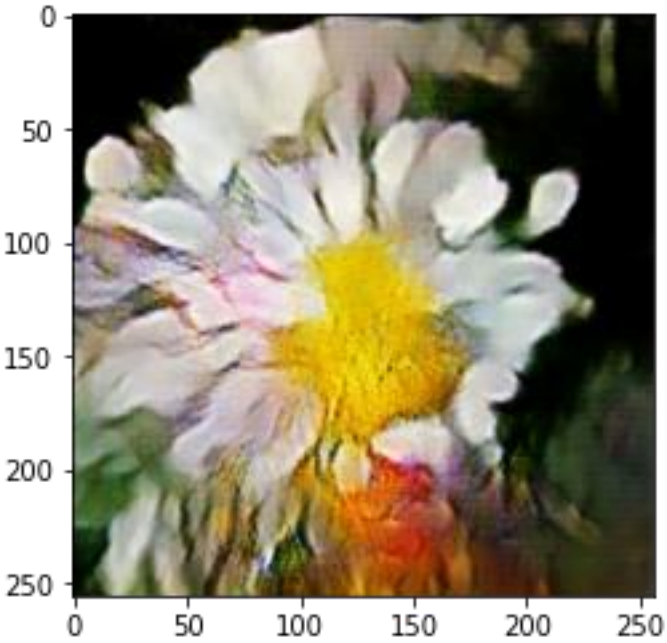
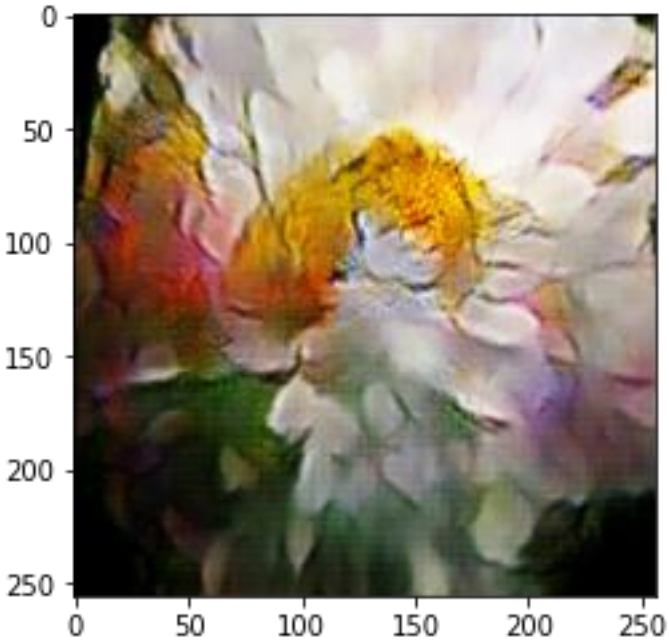
1500 epochs



1000 epochs



500 epochs


0
50
100
150
200
250
0
50
100
150
200
250

0
50
100
150
200
250
0
50
100
150
200
250

Model 3

https://github.com/ravasconcelos/flowers_dcgan/blob/master/dcgan_flowers_3.ipynb

Based on Model 1, but with a bigger noise array.

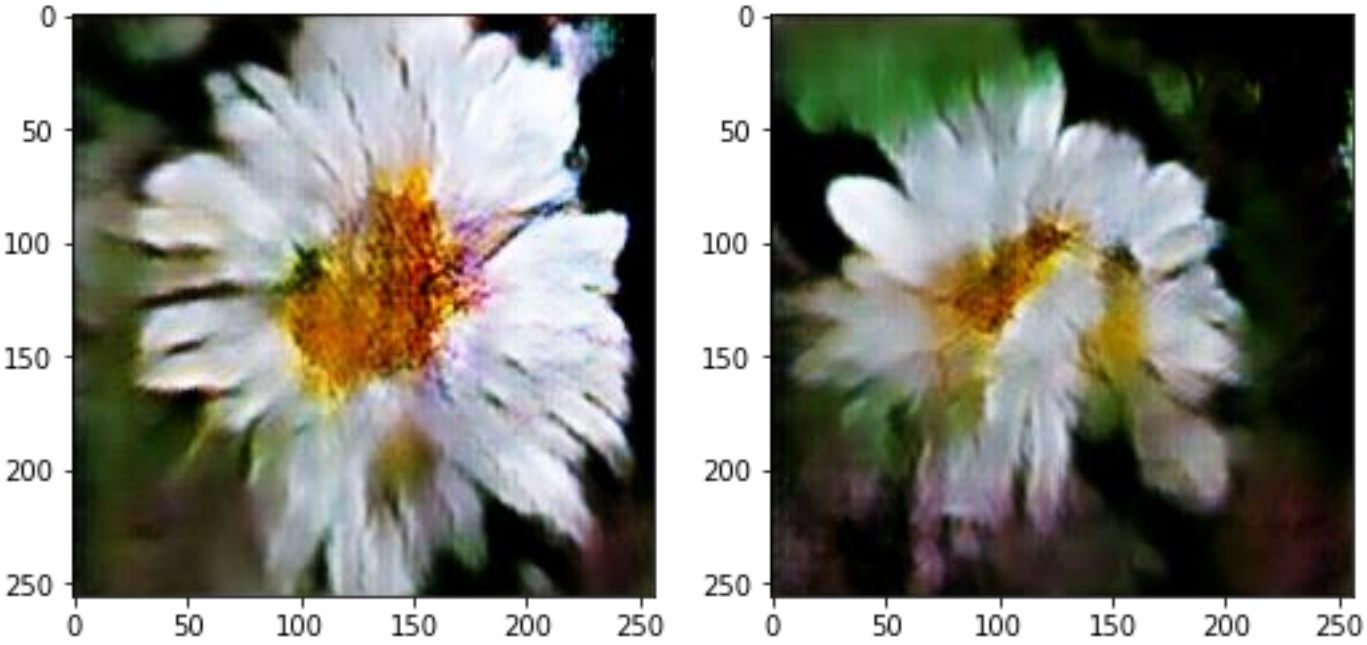
Batch Size: 64

Noise Array: 1000

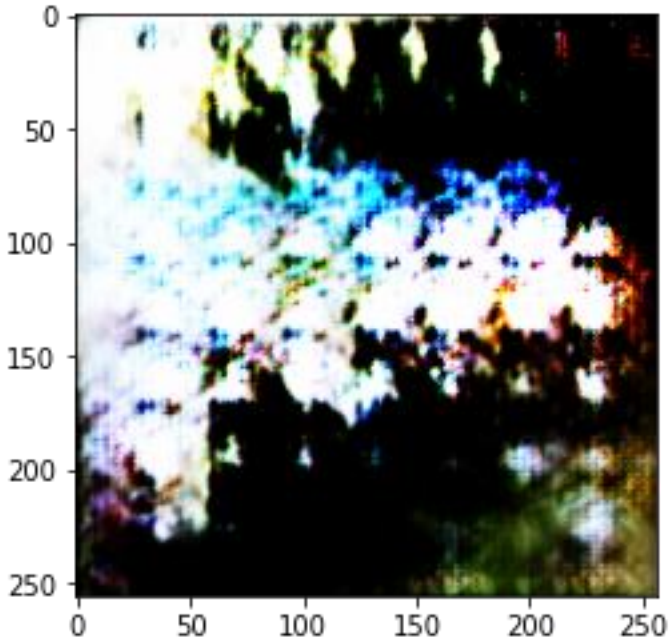
Generator: 6 Conv2DTranspose layers, initial shape 8, 8, 512

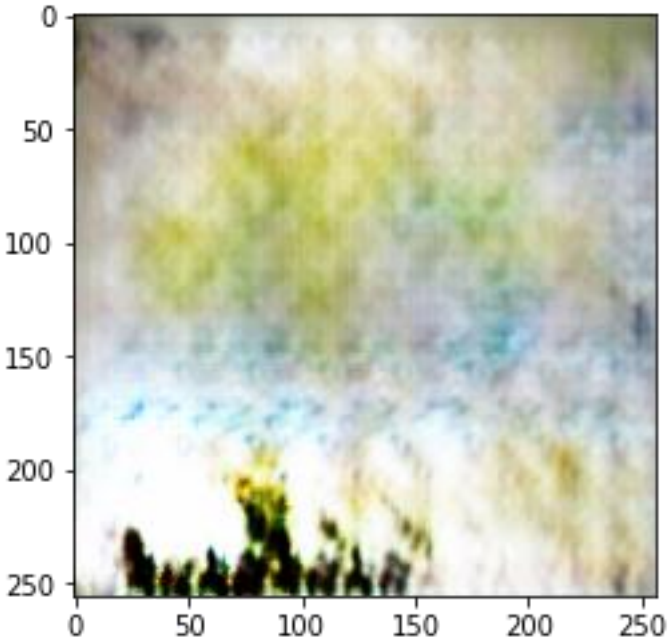
Discriminator: 5 Conv2D layers

2000 epochs



1000 epochs





Model 4

https://github.com/ravasconcelos/flowers_dcgan/blob/master/dcgan_flowers_4.ipynb

This model was based on Model 2, but with some hyperparameters from the Paper: https://arxiv.org/pdf/1511.06434.pdf%C3%AF%C2%BC%E2%80%B0

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.

Batch Size: 64

LR_D = 0.0002 # Paper

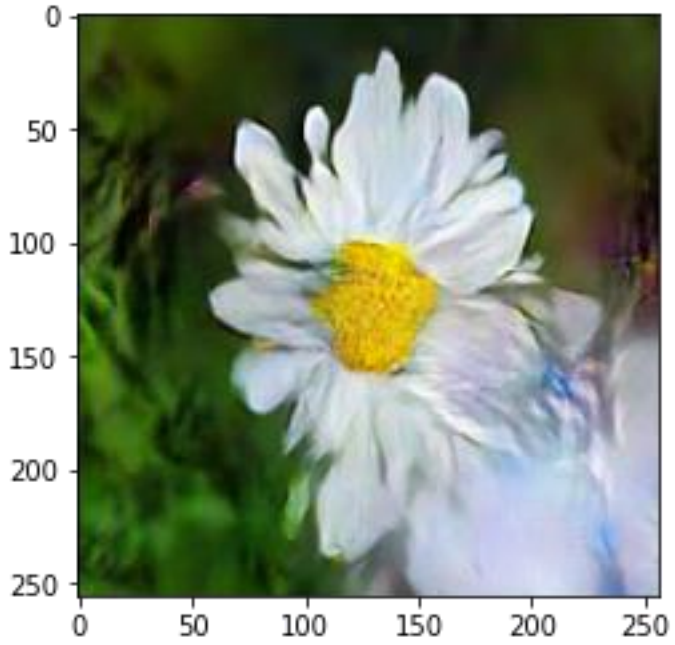
LR_G = 0.0002 # Paper

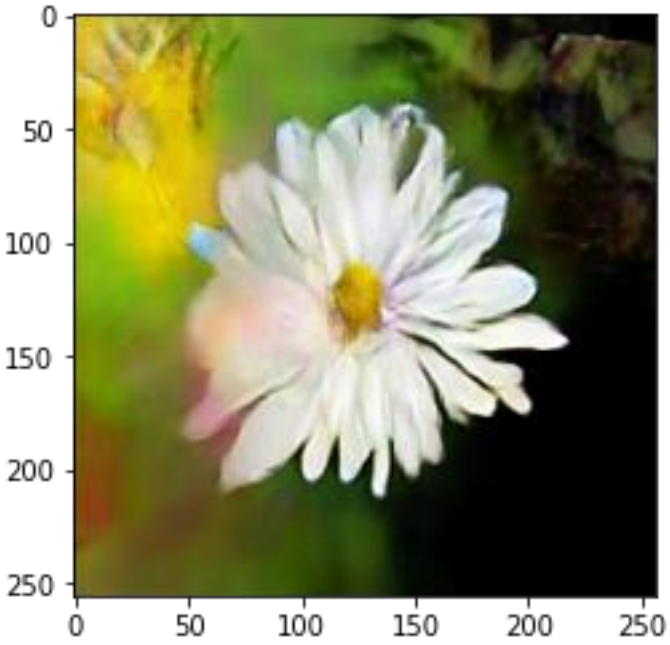
Noise Array: 100

Generator: 3 Conv2DTranspose layers, initial shape 64, 64, 256

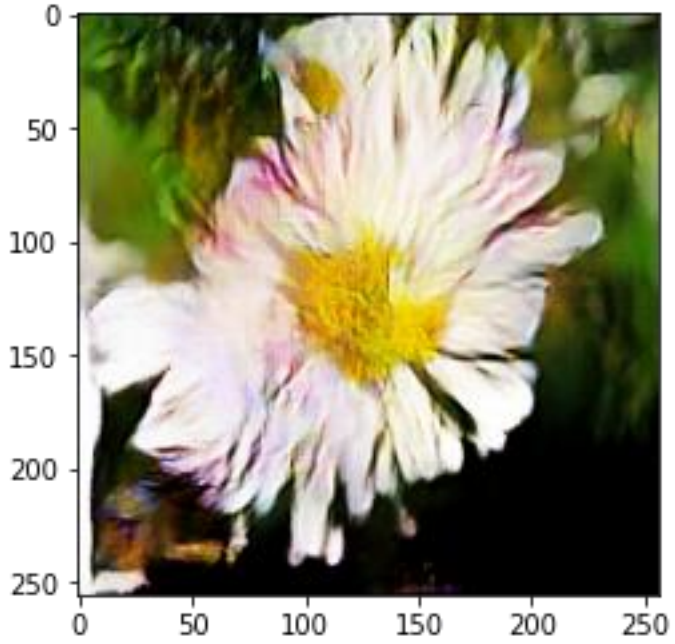
Discriminator: 2 Conv2D layers

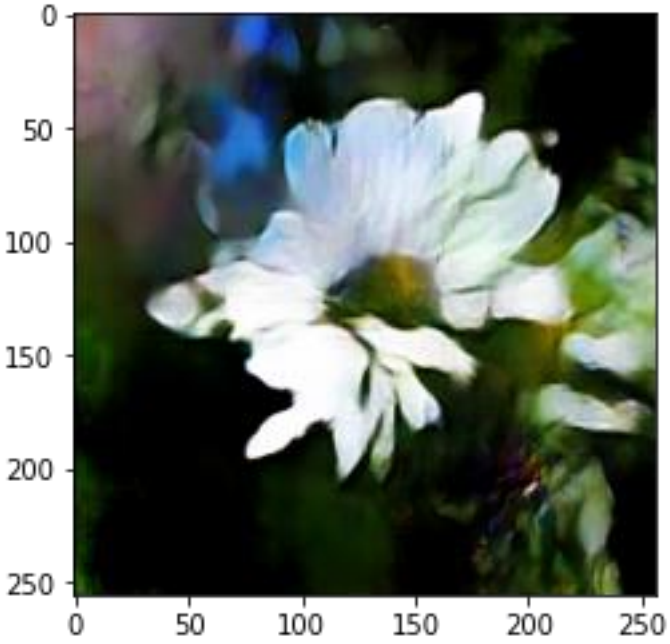
1000 epochs





500 epochs





Model 5

https://github.com/ravasconcelos/flowers_dcgan/blob/master/dcgan_flowers_5.ipynb

This model was based on the model described in the Paper.

Batch Size: 128

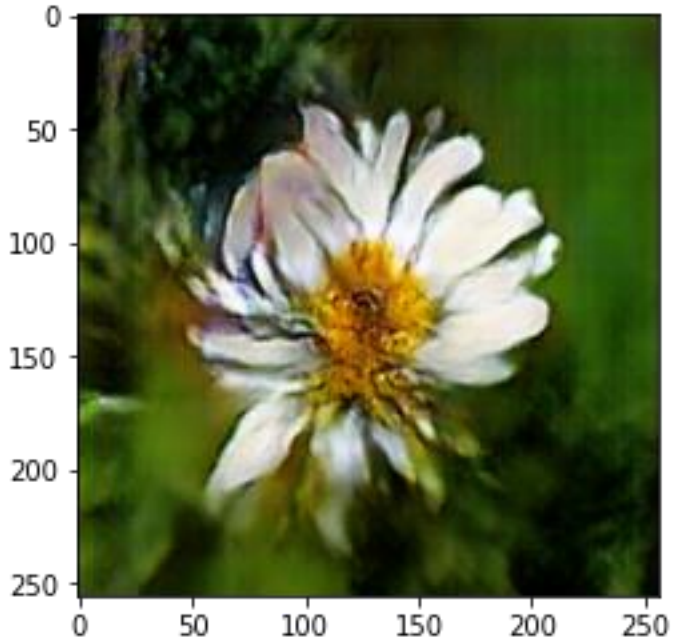
Noise Array: 100

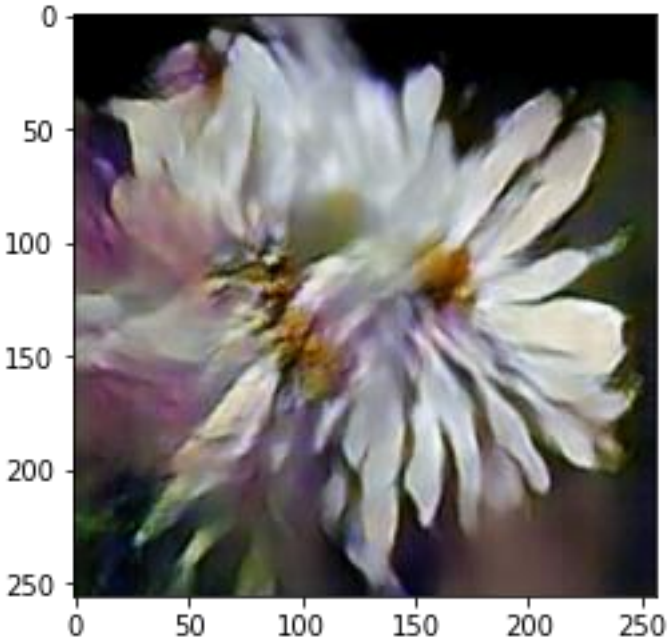
Generator: 4 Conv2DTranspose layers, initial shape 16, 16, 1024

Discriminator: 3 Conv2D layers

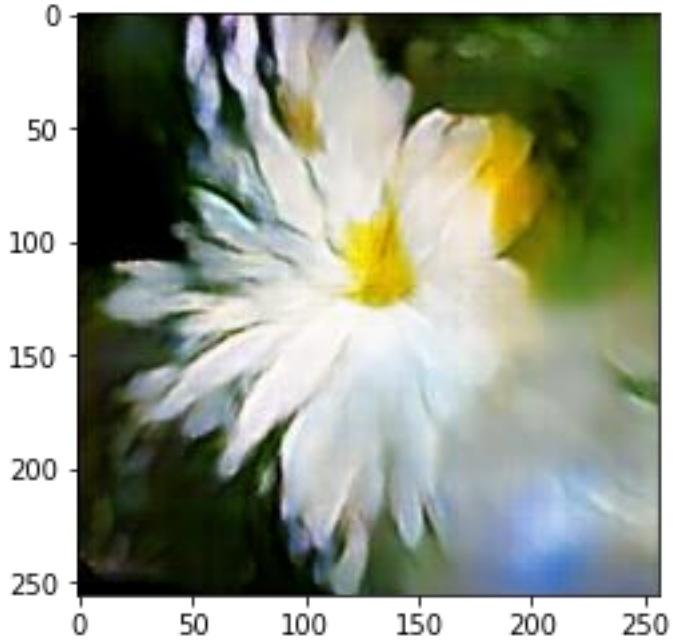
5000 epochs

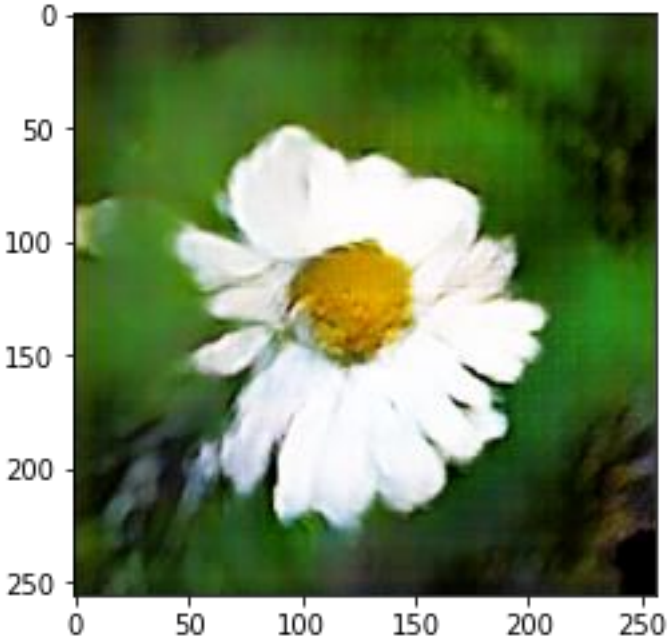
3000 epochs



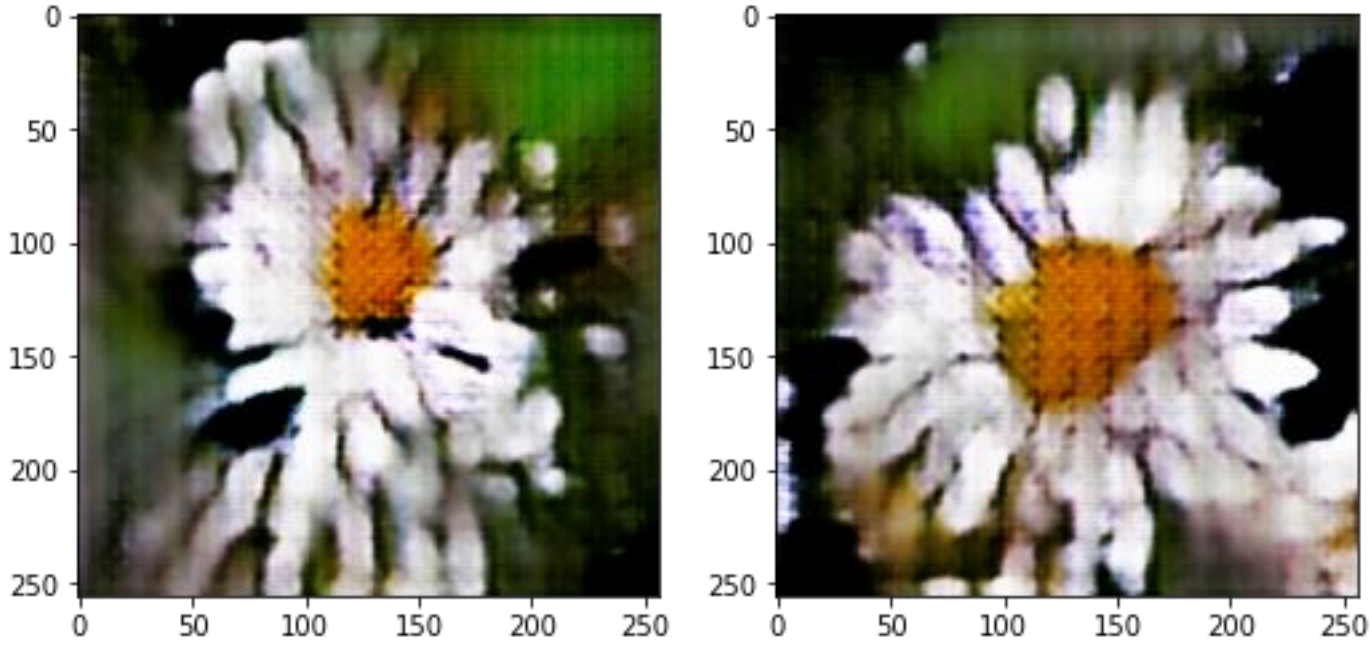


2000 epochs

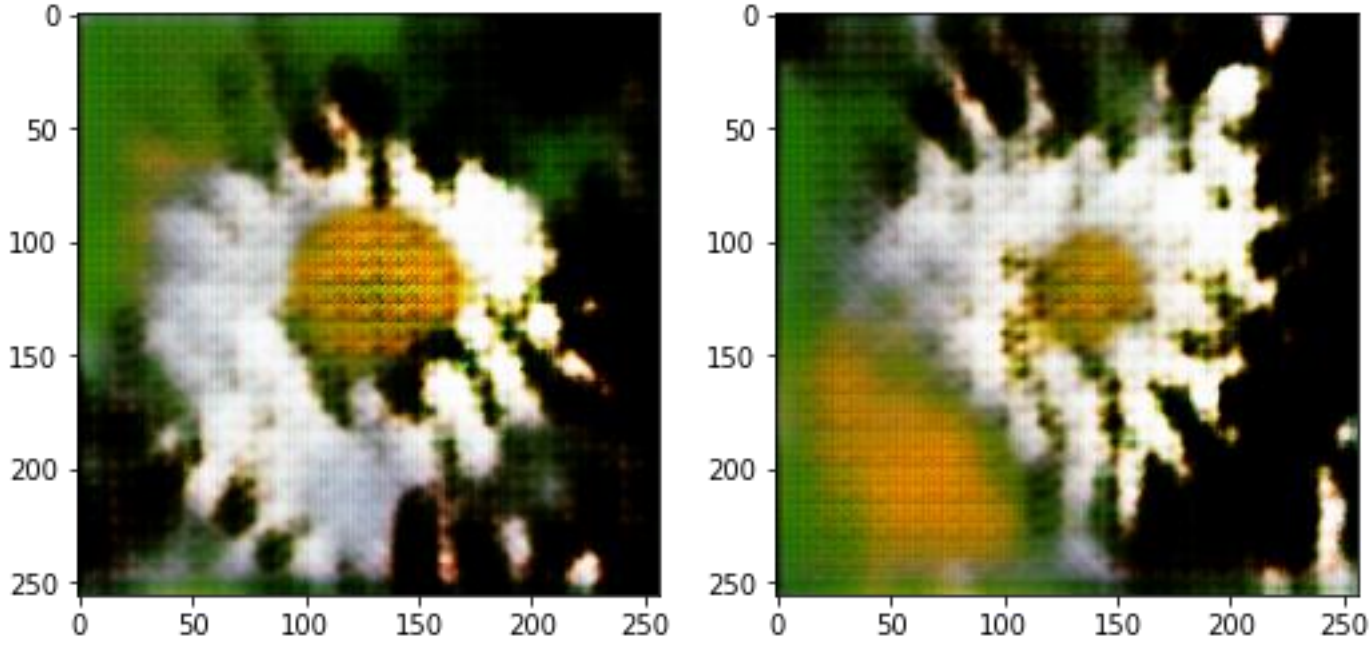




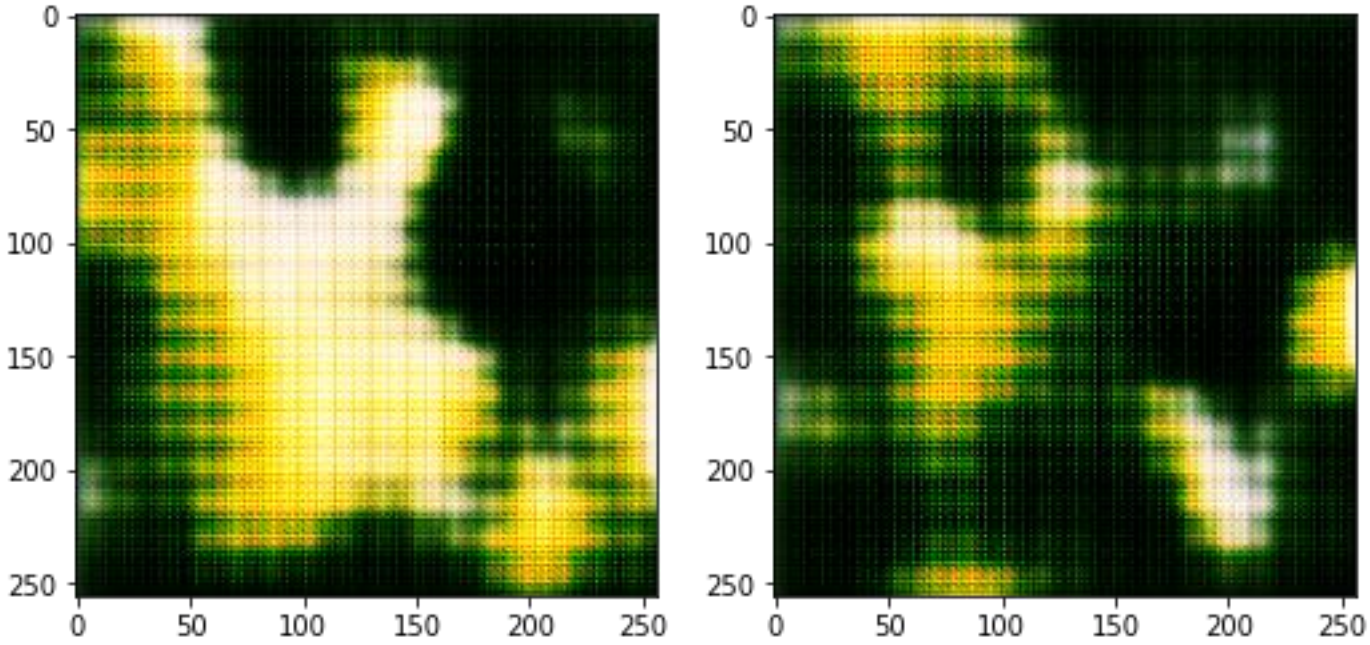
1000 epochs



500 epochs



100 epochs


0
50
100
150
200
250
0
50
100
150
200
250
0
50
100
150
200
250
0
50
100
150
200
250

Model 6

https://github.com/ravasconcelos/flowers_dcgan/blob/master/dcgan_flowers_6.ipynb

This model was based on Model 4, but with strides (2,2) in all the layers.

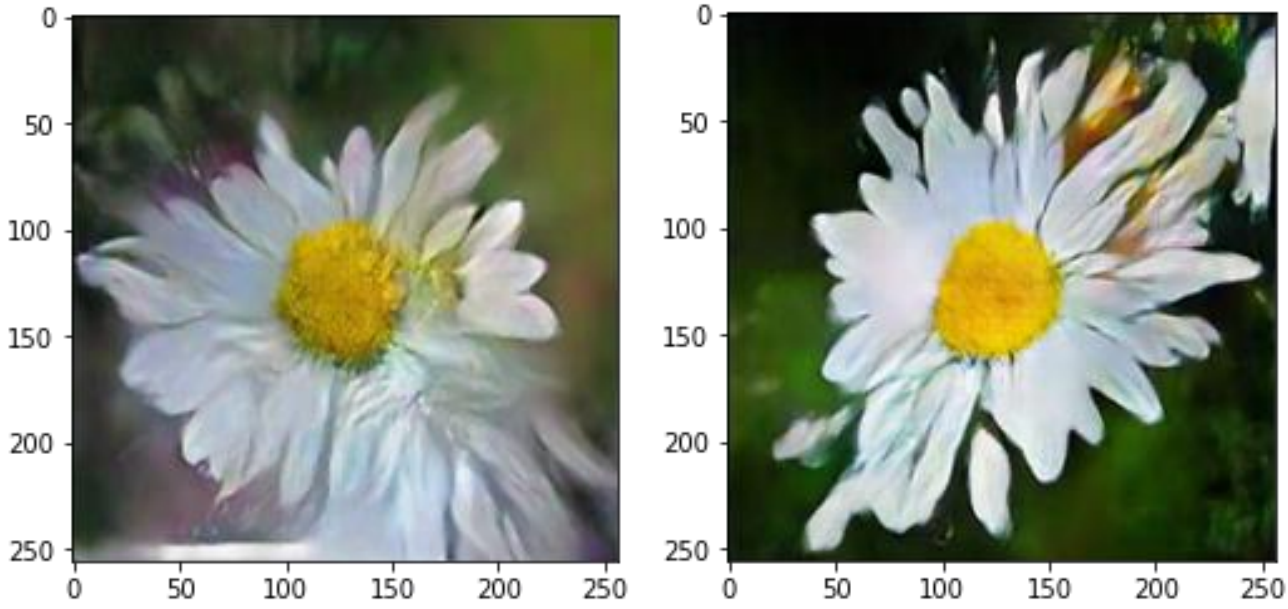
Batch Size: 64

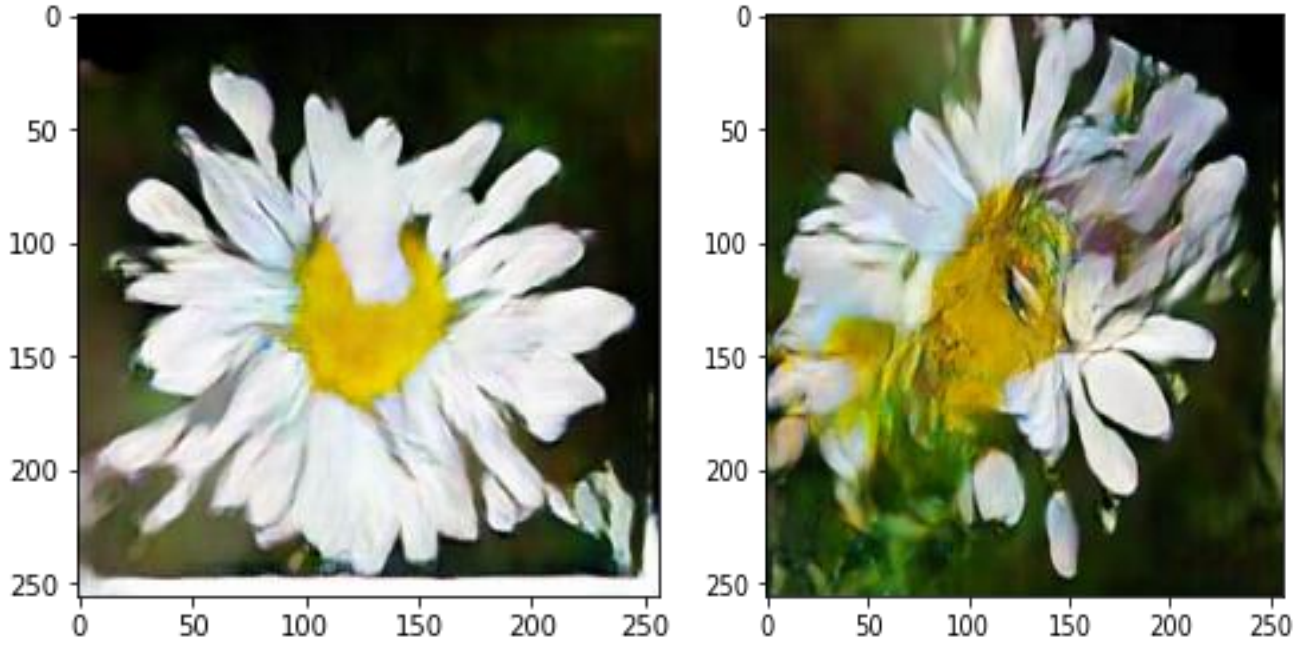
Noise Array: 100

Generator: 3 Conv2DTranspose layers, initial shape 32, 32, 512

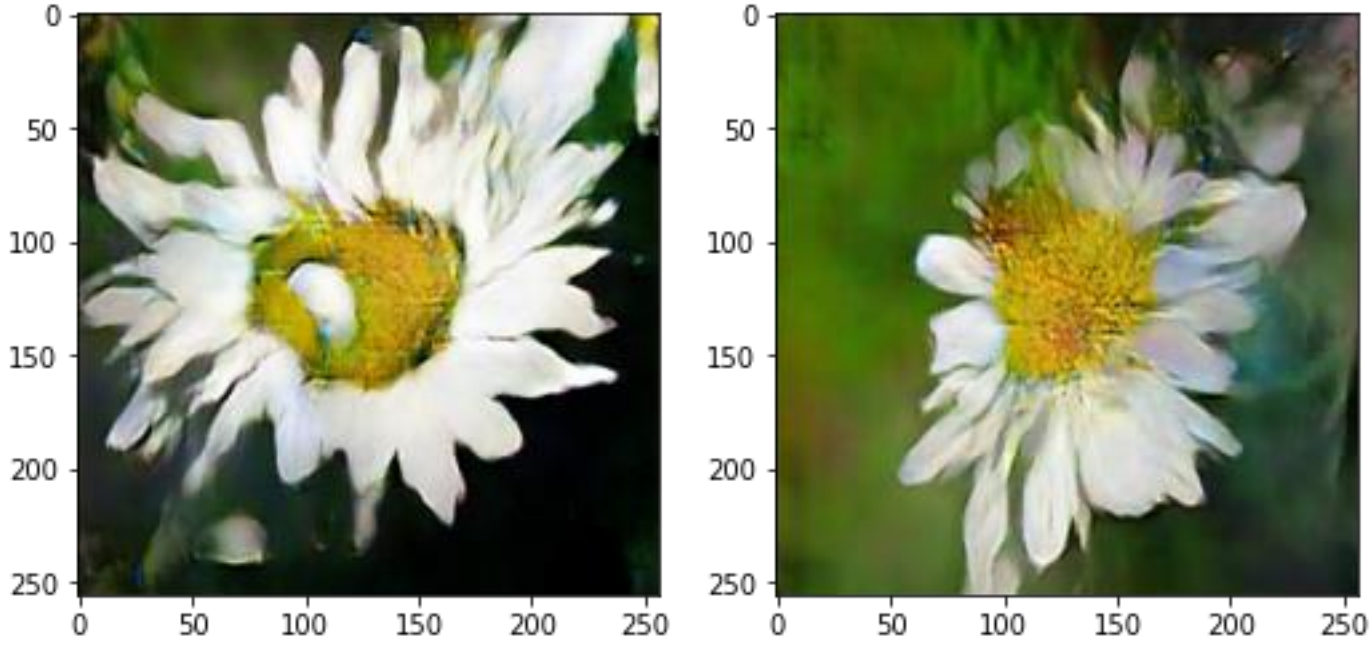
Discriminator: 2 Conv2D layers

6000 epochs

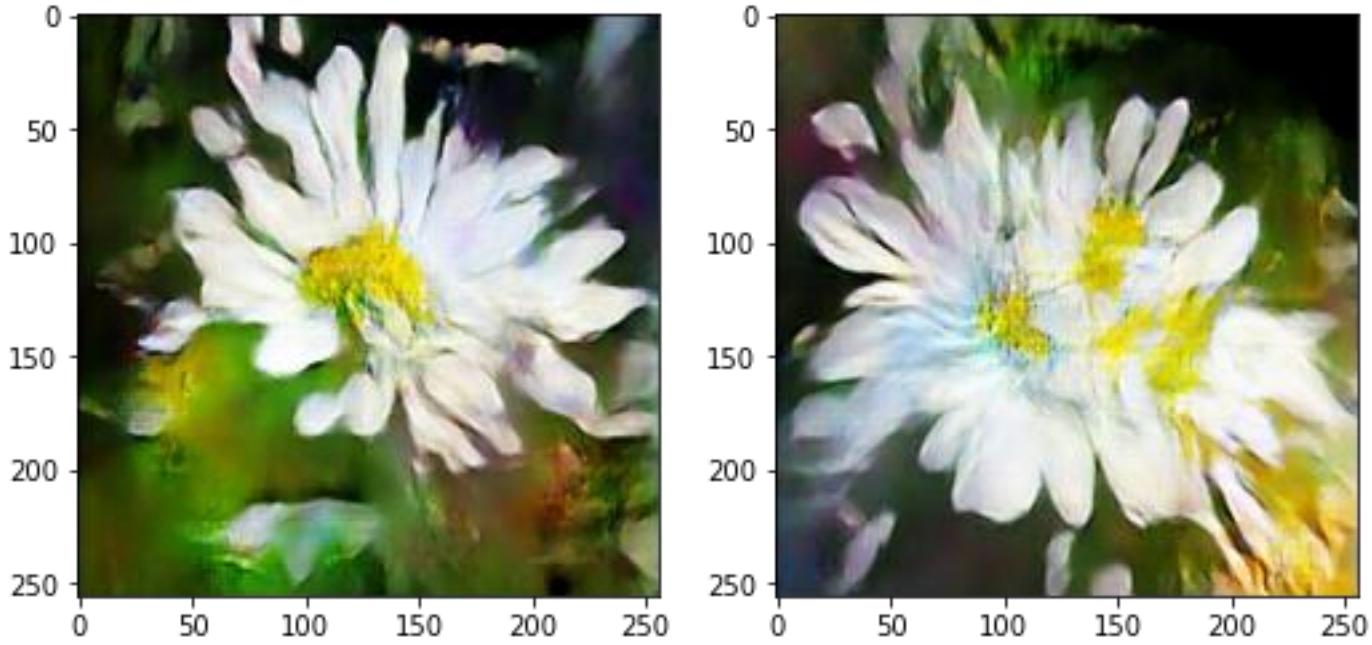




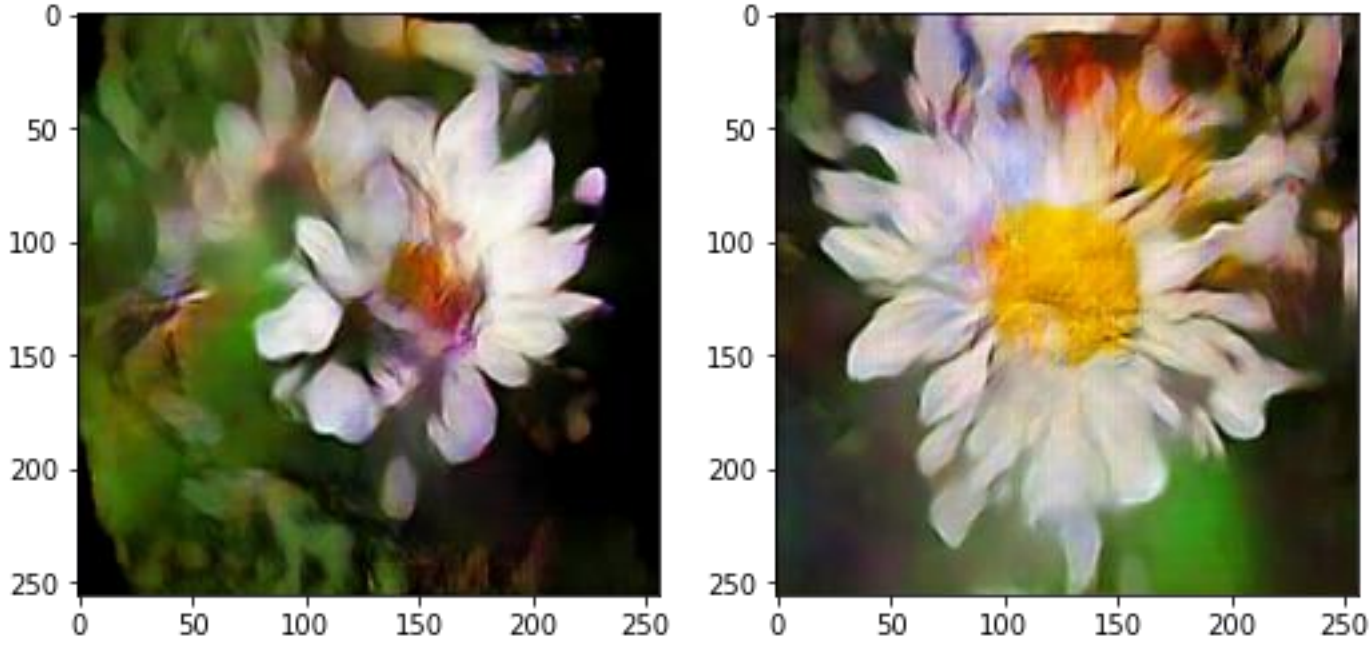
4000 epochs



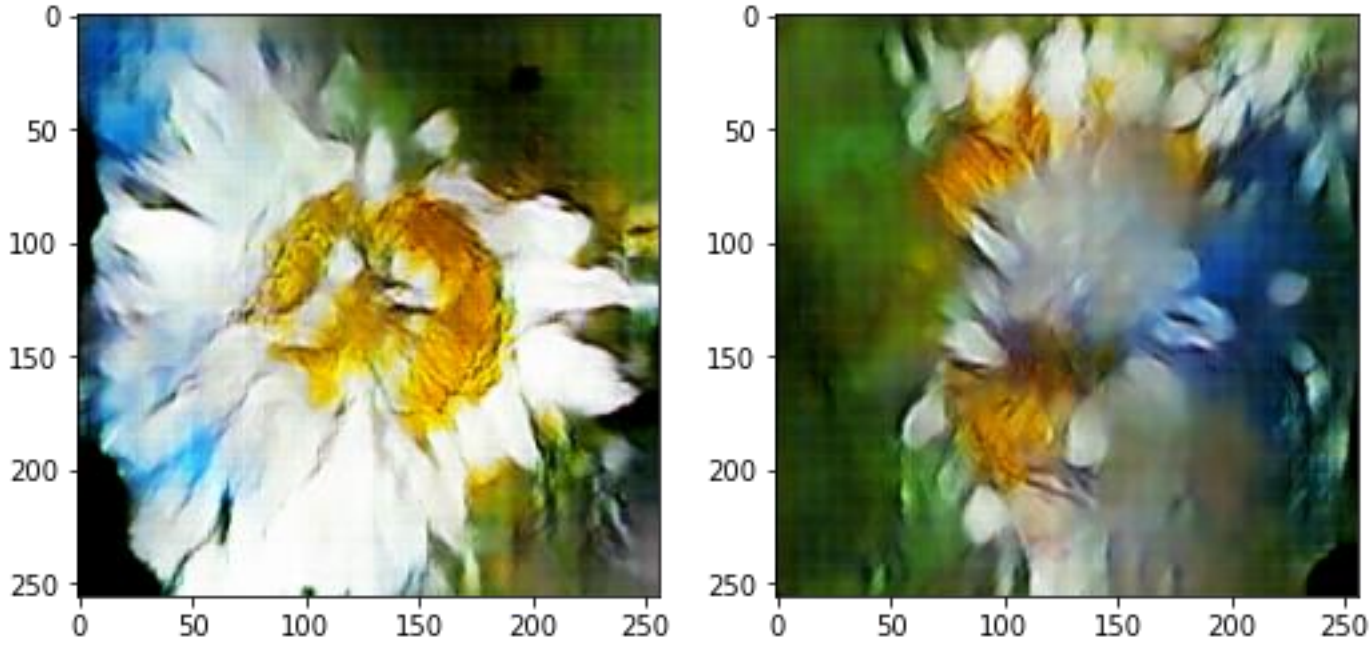
3000 epochs



2000 epochs



1000 epochs



500 epochs