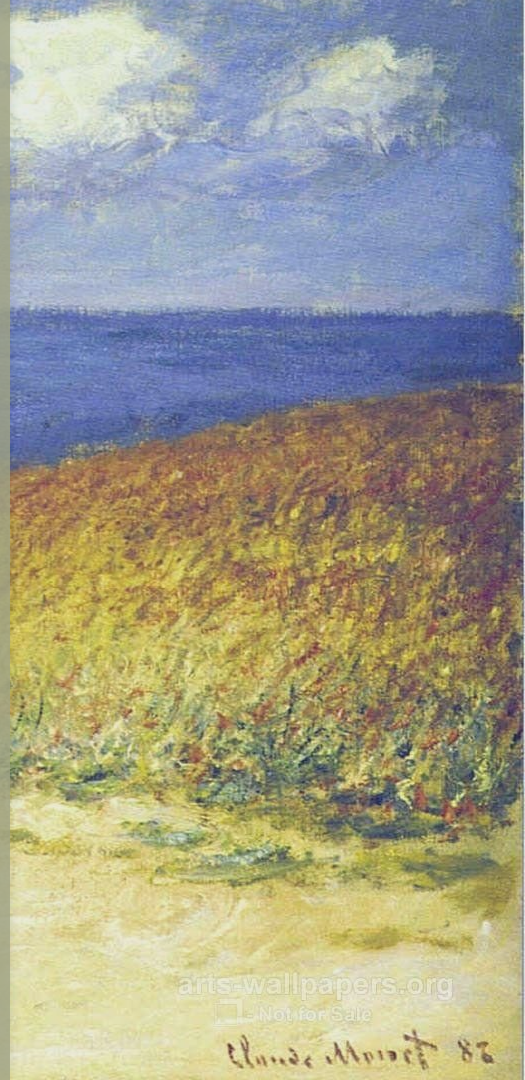
Generative Art

Muntaha Jaber - Elaf Alsaedi



AGENDA

01 INTRODUCTION

02 DATA ET ALGORITHM

03 MODEL ET RESULTS

04 CONCLUSION

01

INTRODUCTION

What is Generative Art, Backstory and Problem statement

What is Generative Art?

Any work of art created with the assistance of artificial intelligence.

Art doesn't have a concrete definition.
Some people might see it as a pure human activity and it is a crime to automate art.
Others might see computers as tools to make art just like brushes and colors.
We see it as an interesting opportunity to explore how far computers can reach by understanding human culture and communications.

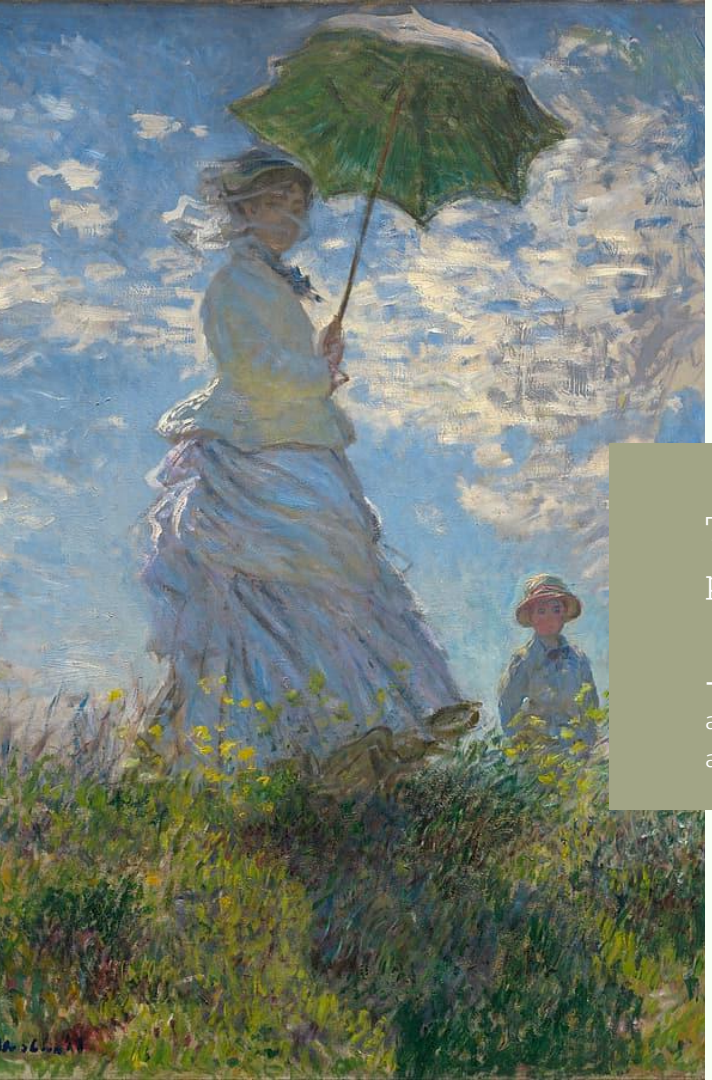
Problem Statement

Generate aesthetic images with the aid of Neural Networks.

02

DATA and Algorithm

More about the data



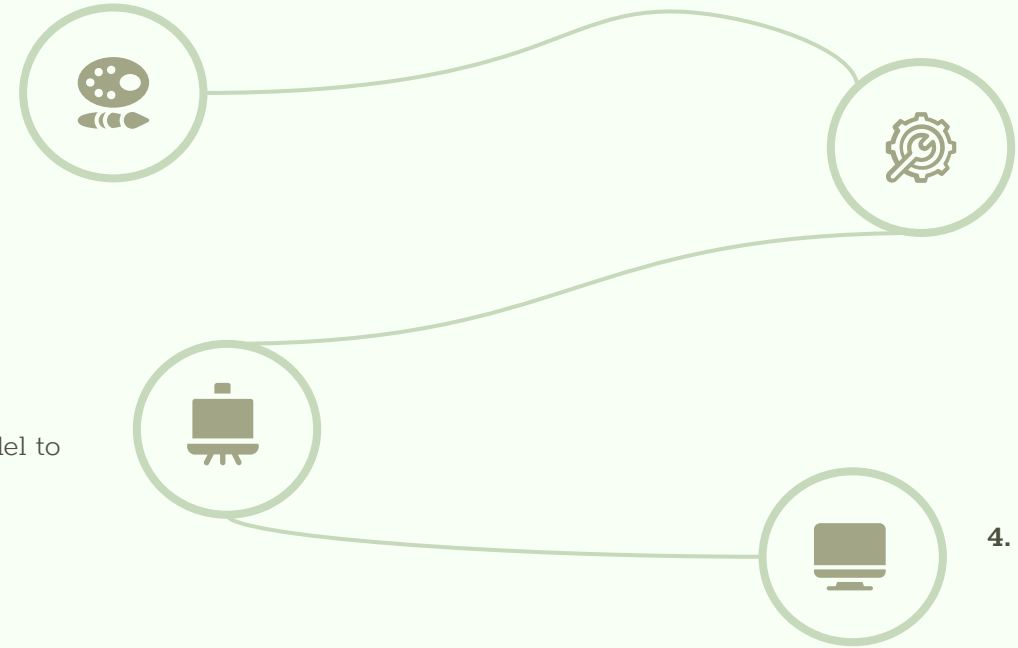
The data consists of 1,193 Images of Monet paintings, it can be found in Kaggel.

-Claude Monet was a famous French painter whose work gave a name to the art movement Impressionism with capturing light and natural forms.

Workflow

1. Presproseseng

- Resize images
- CenterCrop
- Change image to Tensor
- Normalizing



2. Build model
GAN discriminator & Generator layers

3. Train model
Finetune the model to find best result.

4. Present the results

- Make a short movie
- Monet Gallery VR experience

Generative adversarial network (GAN)

GAN is a machine learning (**ML**) model in which two neural networks compete with each other to become more accurate in their predictions.

The two neural networks are **generator** and **discriminator**.

Generative adversarial network (GAN)

Real Images

Discriminator Network

Predict label

Real & Generated

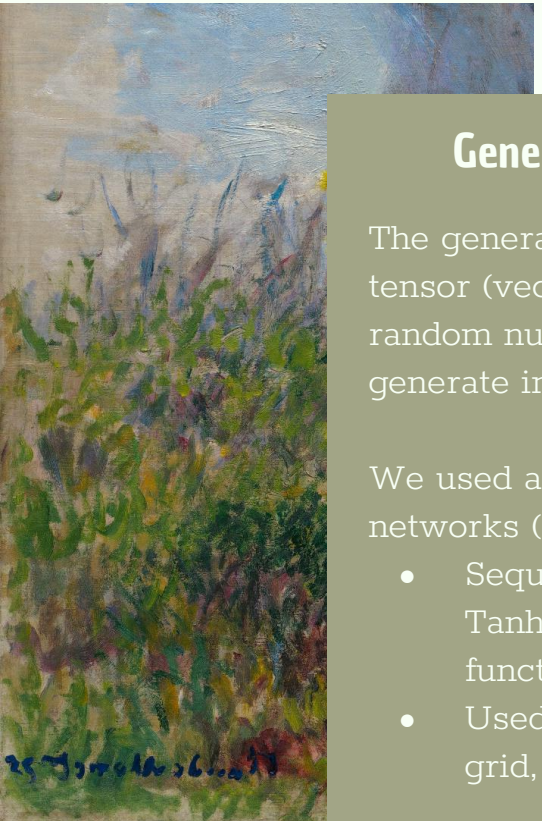
latent tensor

vector or a matrix of random numbers

Generator Network

Generated Images

Discriminator and Generator



Generator Network

The generator takes latent tensor (vector or a matrix of random numbers) as input, and generate images.

We used a convolutional neural networks (CNN)

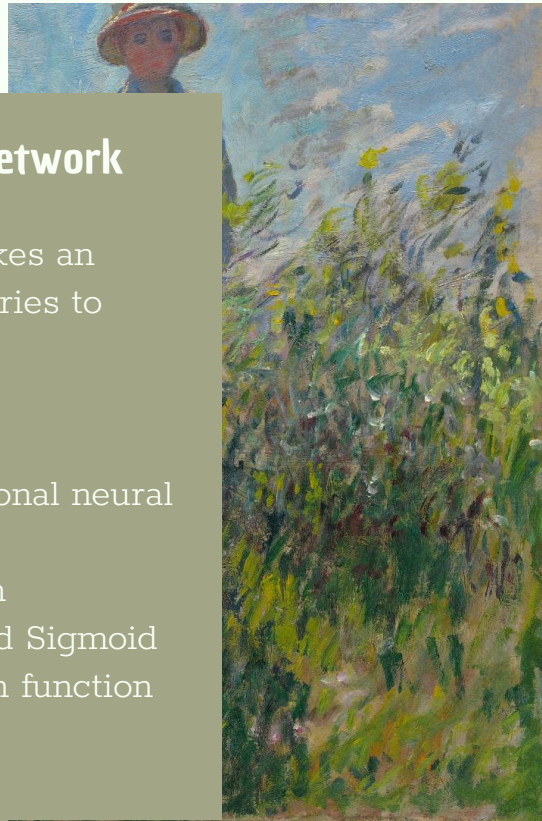
- Sequential with ReLU and Tanh as an activation function
- Used PyTorch for creating grid, save images.

Discriminator Network

The discriminator takes an image as input, and tries to classify it as "real" or "generated".

We used a convolutional neural networks (CNN)

- Sequential with LeakyReLU and Sigmoid as an activation function



Binary cross entropy

$$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i})\right]$$

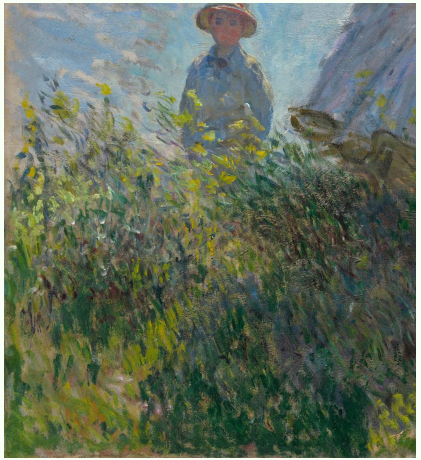
- Creates a criterion that measures the Binary Cross Entropy between the generated and the real image probabilities.
- Treat it as a binary classification problem with labels 0 and 1 for generated and real images respectively.
- Best Log Loss value is zero.

Source

Challenges

1. Resources

GPU and its memory was small for training the model with high pixels

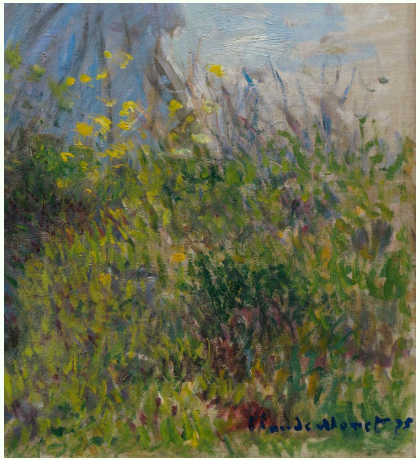


2. Model Complexity

Understanding the model and its best practices was challenging and took a lot of time.

3. Training Time

The model took a lot of time to train and experiment.



03

MODEL and RESULTS

What are the used technique? Results?

PRE-PROCESSING STEPS

Resizing Images

Convert images from 256x256 pixels to 64x64 pixels to match our GPU Capacity

Center Crop

Crop the given image at the center.

ToTensor

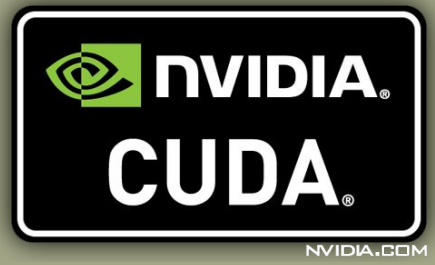
Convert the image to torch.FloatTensor

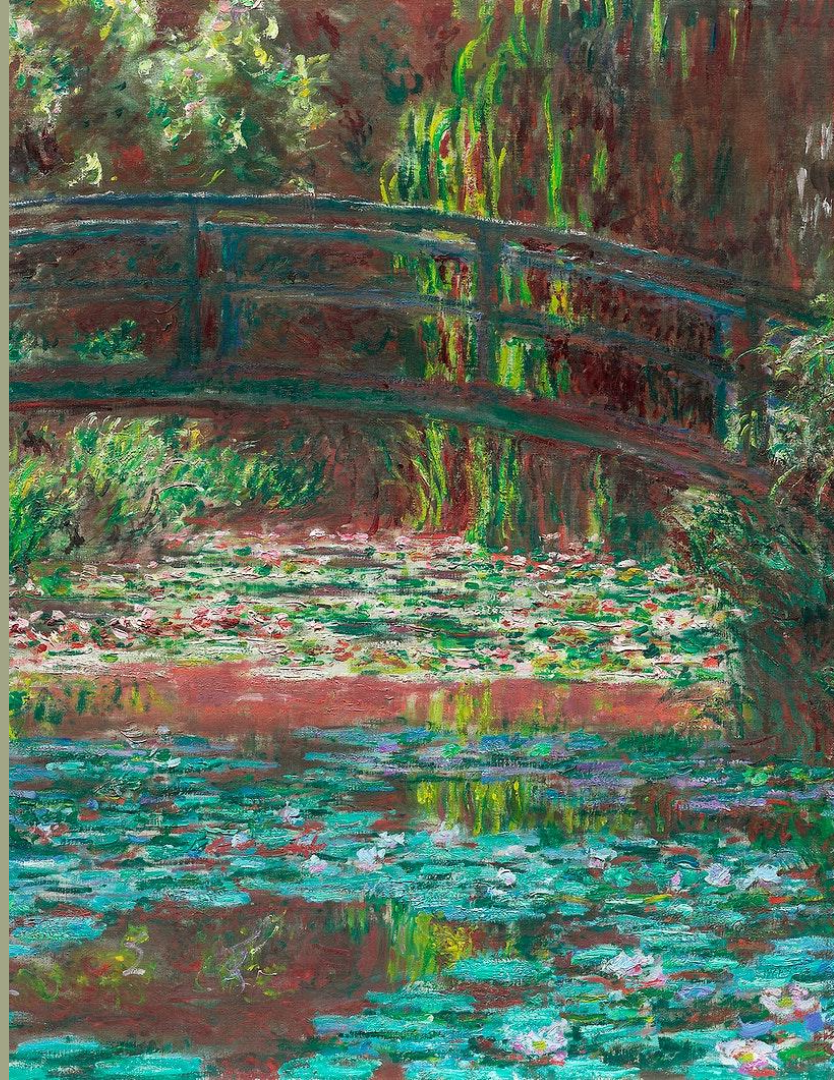
Normalization

Normalize a float tensor with mean and standard deviation

GPU Set-up

CUDA is a parallel computing platform and application programming interface allow using GPU.



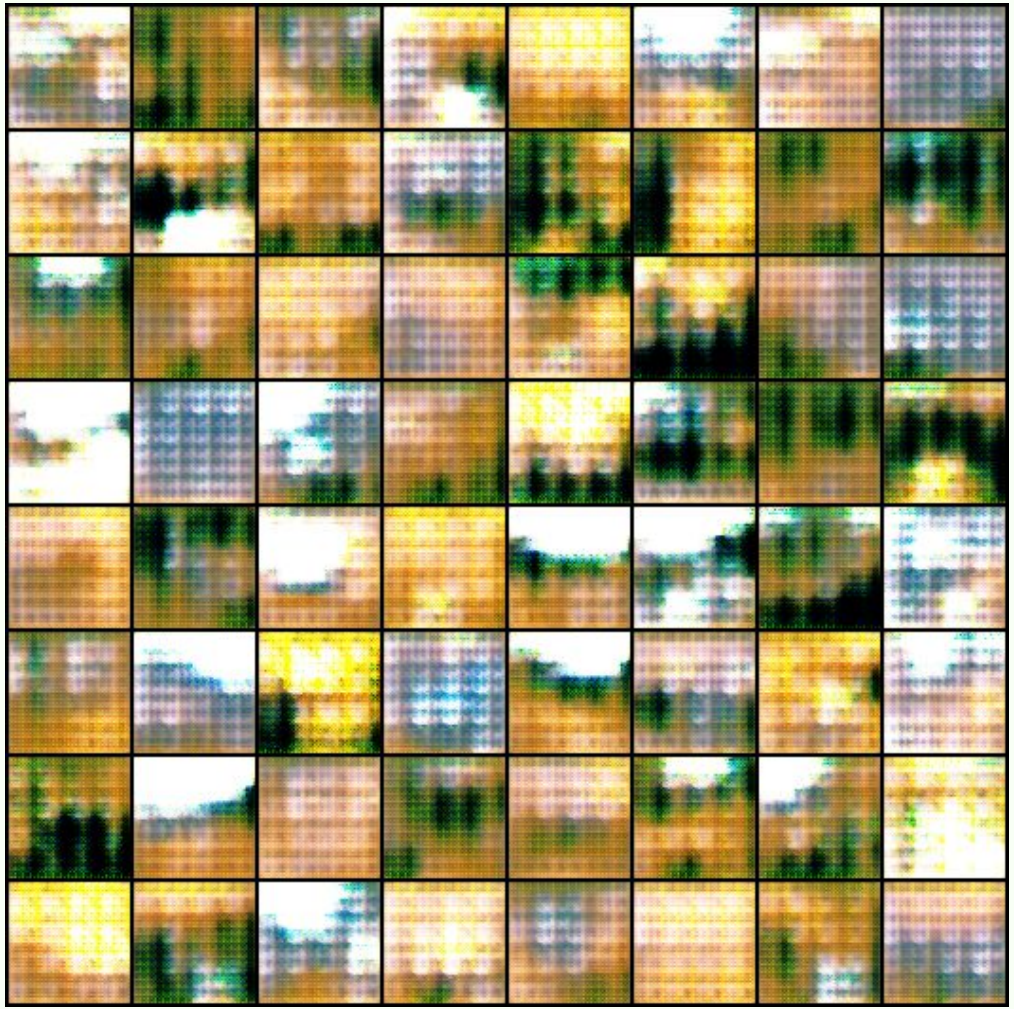


Device Specification

RAM: 16.0 GB
GPU: Nvidia GeForce RTX 2070 / 930
VRAM: 8.0 GB
Training Time: Around 8 hours

Baseline Model

- Epochs=100
- Learning rate=0.01
- Pixels = 64
- Batch size = 128
- Latent size = 128

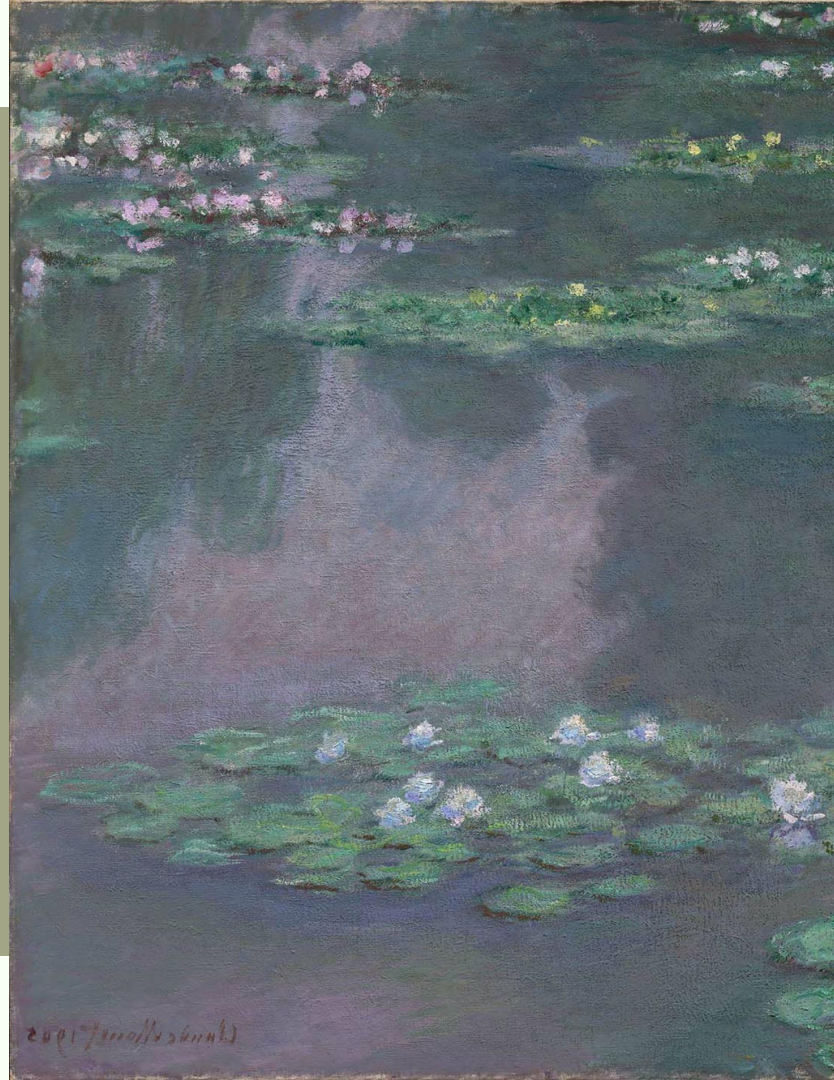


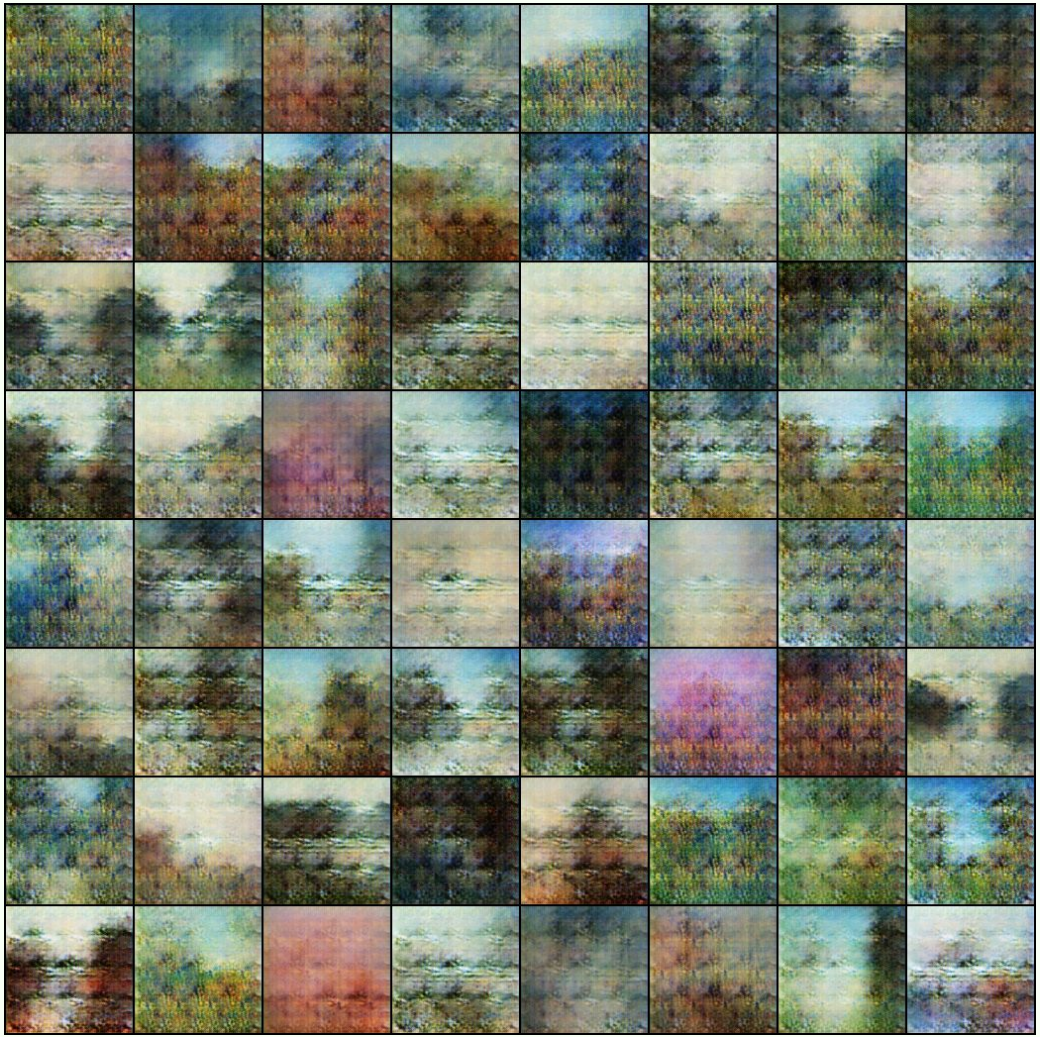
Baseline Model

- 64x64 pixels
- Epoch: 100
- High Loss

More Pixels

- 128 pixels failed to give a good results, epochs stopped at 365 with high loss.
- 256 pixels wouldn’t work on our GPU.



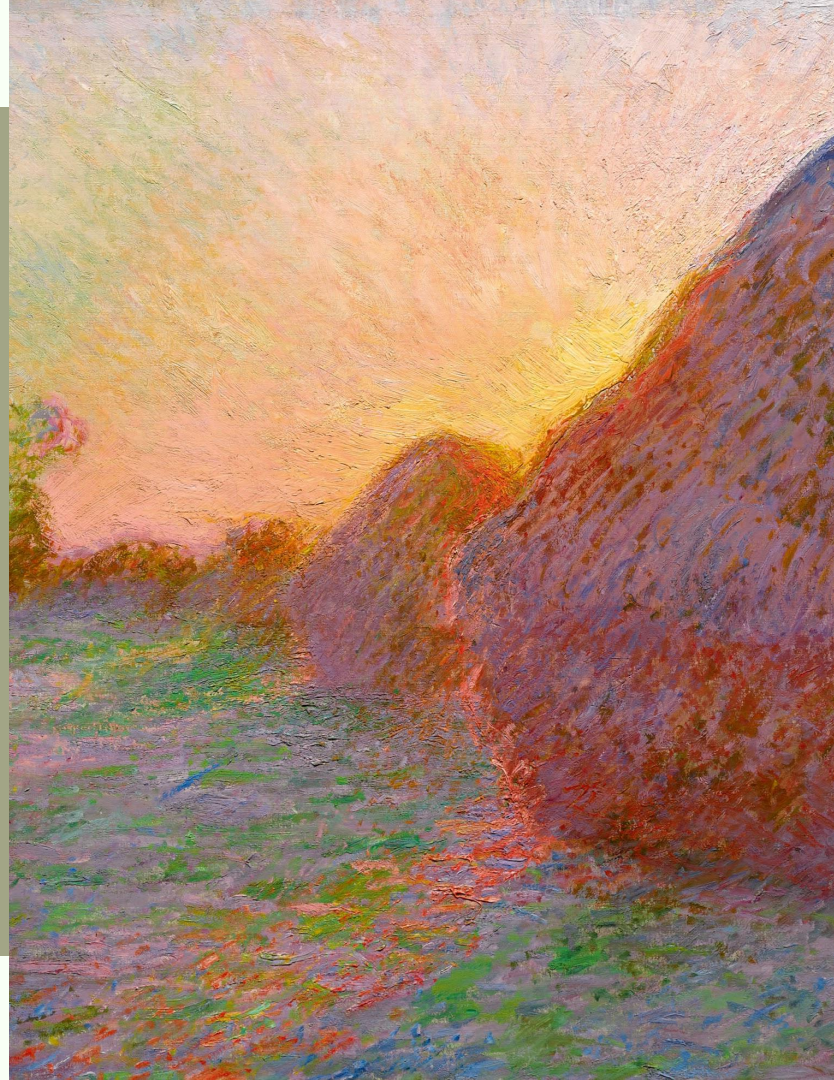


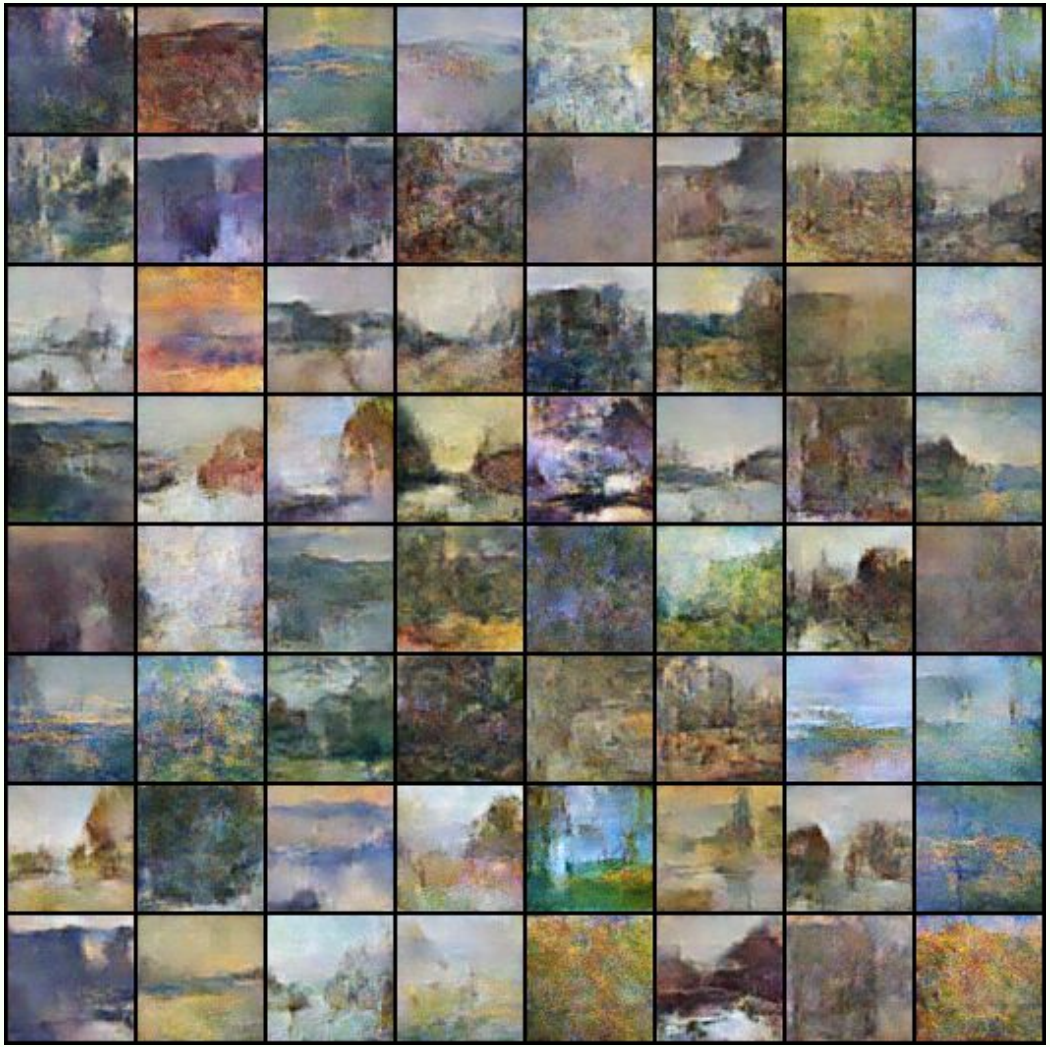
More Pixels

- 128x128 pixels
- Epoch: 365,
  loss_g: 5.2822,
  loss_d: 0.0240

Best Model

- epochs=500
- learning rate=0.0001
- Pixels = 64
- batch size = 64
- latent size = 128





More Pixels

- 128x128 pixels
- Epoch: 500
  loss_g: 5.2822,
  loss_d: 0.0240



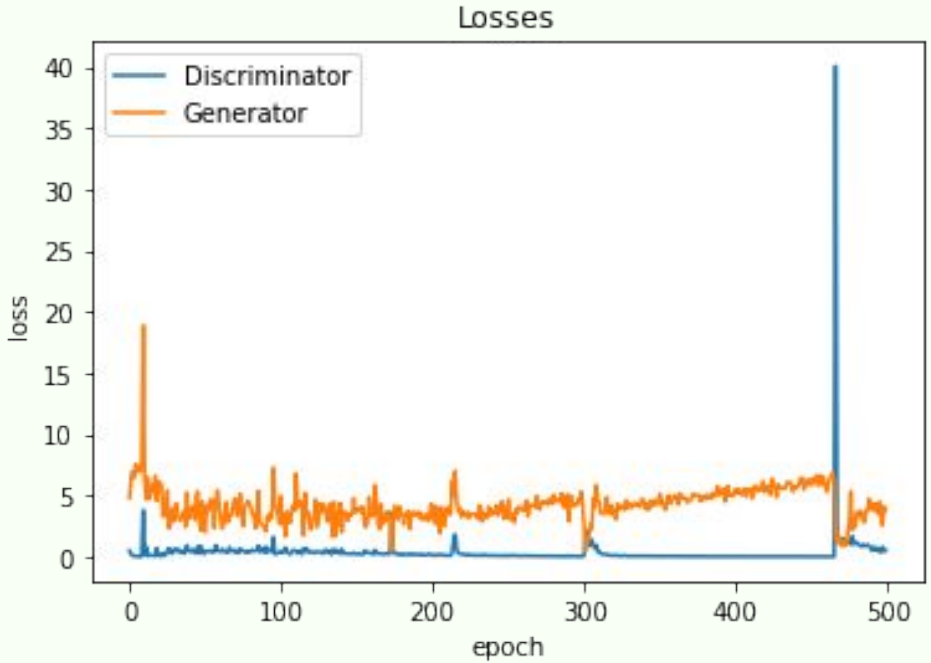



Model Score

Early Stage

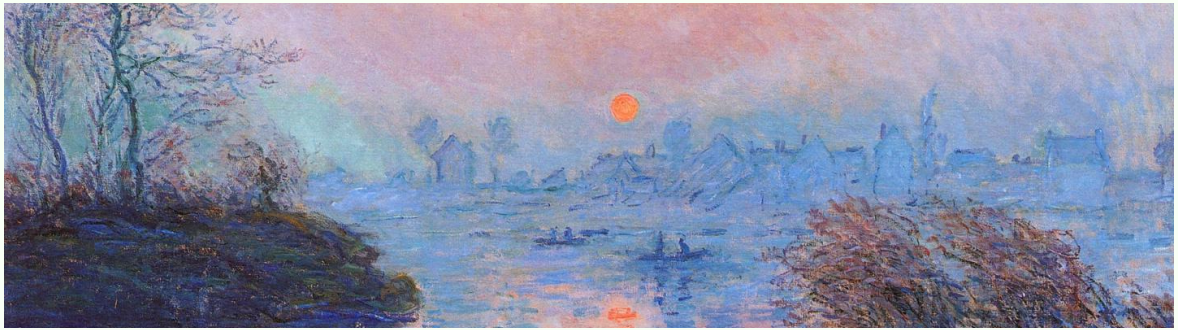
Final Model

Loss


Losses
Discriminator
Generator
loss
epoch
0
5
10
15
20
25
30
35
40
0
100
200
300
400
500

Short Film

VR Monet Gallery Experience

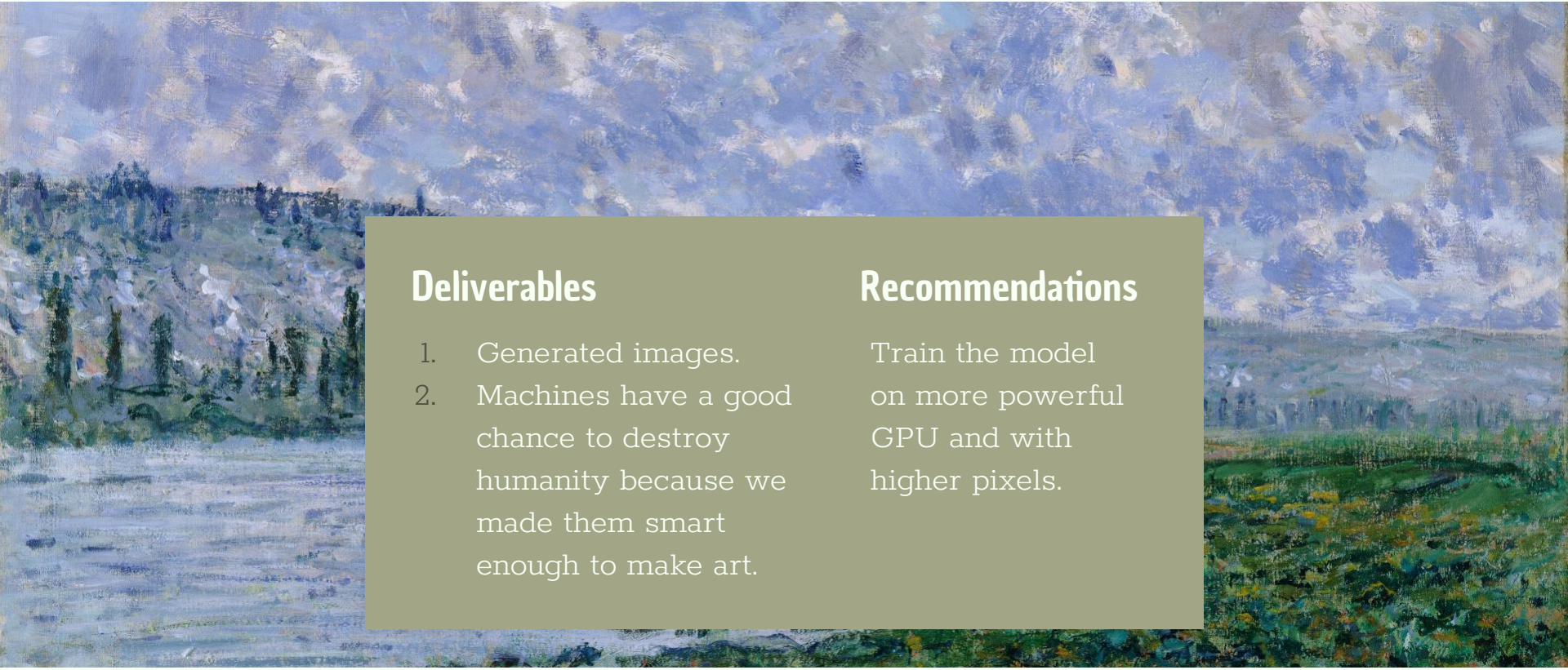


04

CONCLUSION

Deliverables, Recommendations

CONCLUSION



Deliverables

1. Generated images.
2. Machines have a good chance to destroy humanity because we made them smart enough to make art.

Recommendations

Train the model on more powerful GPU and with higher pixels.