USC University of Southern California

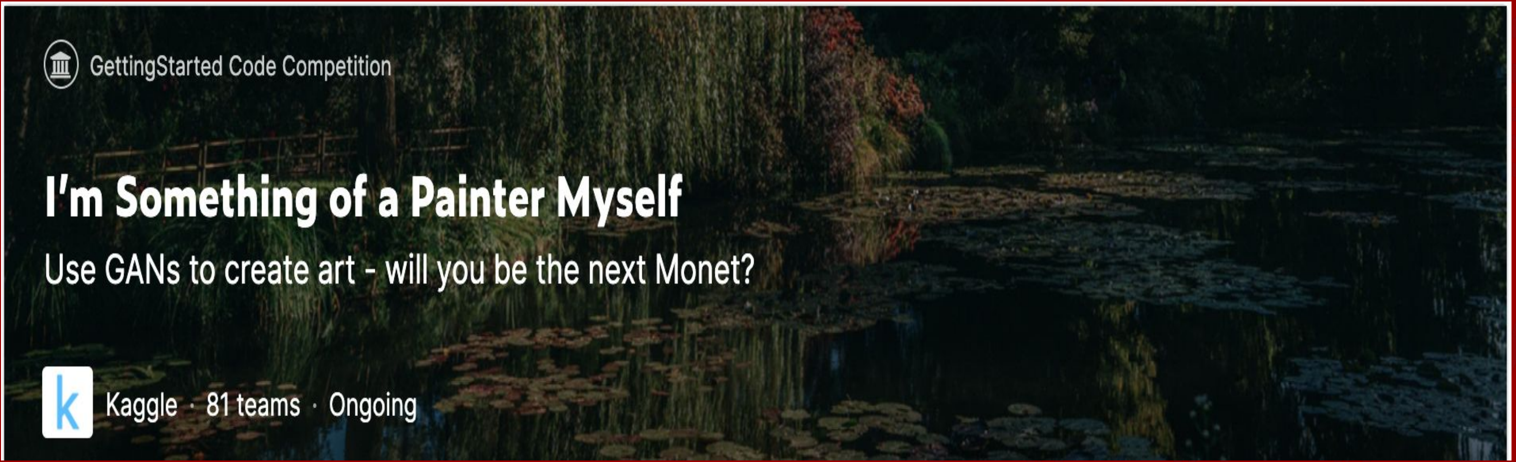
Cyclic Generative Adversarial Networks for Photograph to Painting Translation

EE 541: A Computational Introduction to Deep Learning
Prof. Brandon Franzke

By
Aditya Anulekh Mantri (USC ID: 80849574464)
Shoumik Nandi (USC ID: 3621442772)
Prithvi Dalal (USC ID: 1939114566)

USC

Competition



Objective : Convert photographs into Monet-Style images

Introduction

Aim - Create Monet-style images from photographs

Assume two domains - Monet paintings and Photographs - have some relationship and seek the relationship.

Two mappings : $G : X \rightarrow Y$ and $F : Y \rightarrow X$. G and F are inverse of each other.

Introducing Cycle Consistency loss -> $F(G(x)) \approx x$ and $G(F(y)) \approx y$

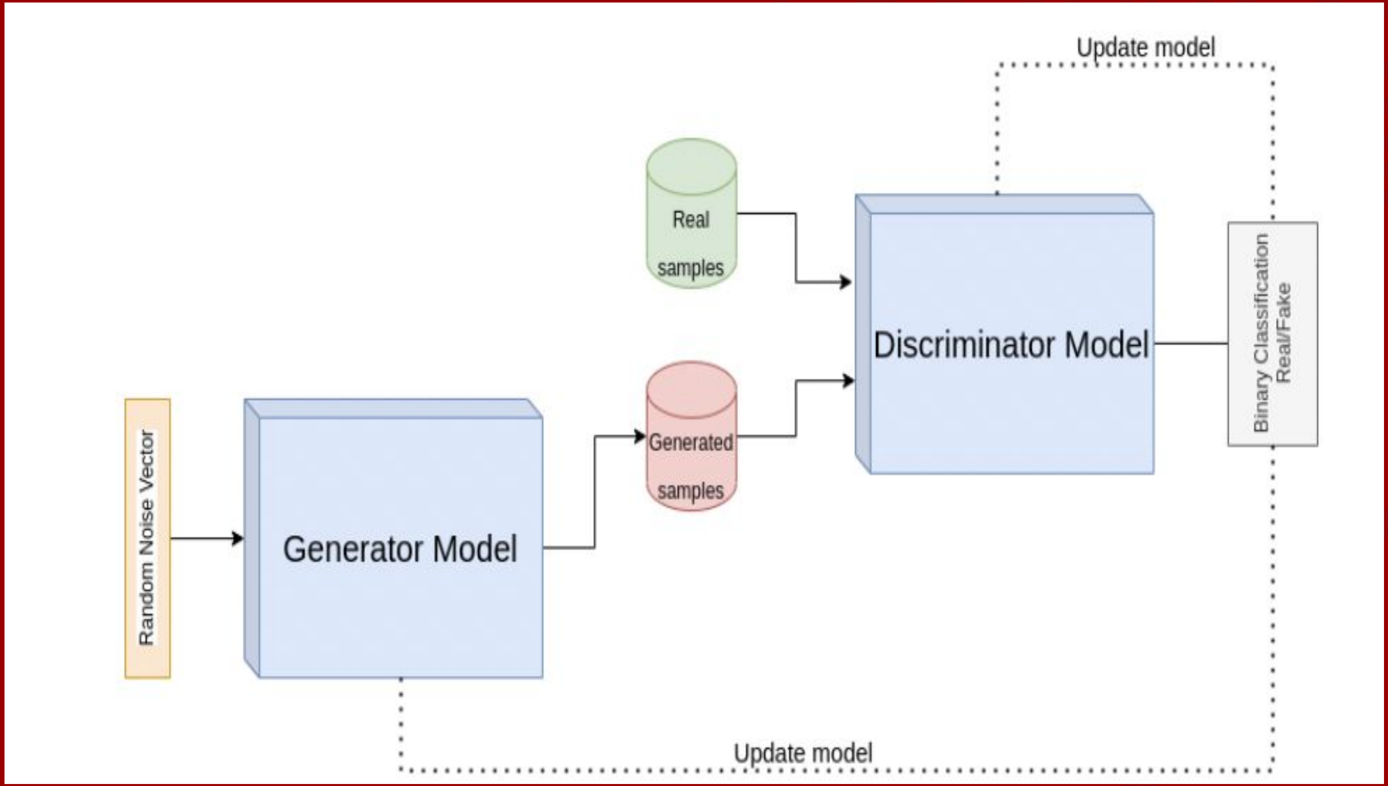
Dataset : 1. Dataset posted on the Kaggle competition

2. Dataset used by Zhu et al in original CycleGAN paper

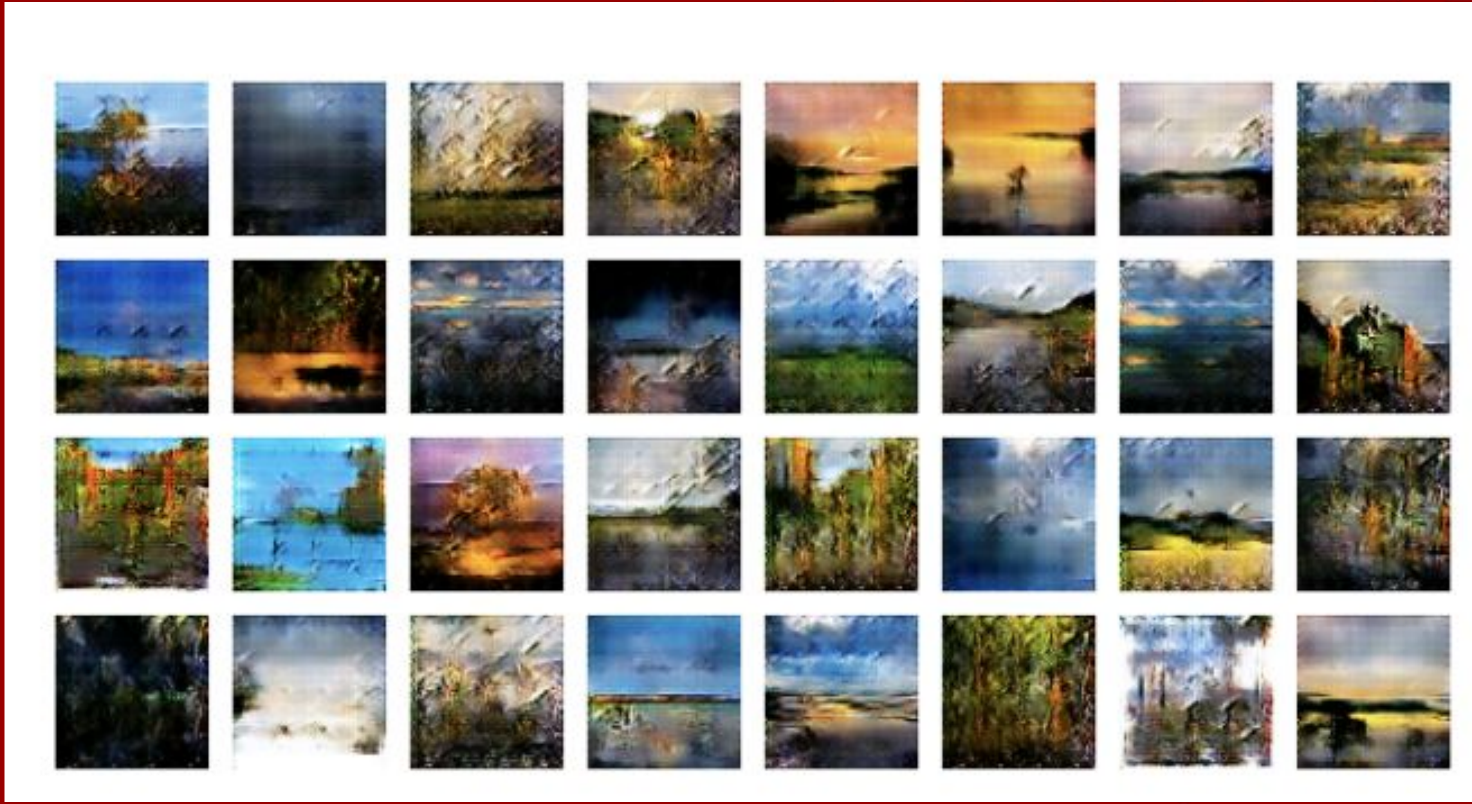
Related Work: 1. Generative Adversarial Network (GAN)

2. Deep Convolutional Generative Adversarial Network

Introduction to Generative Adversarial Network



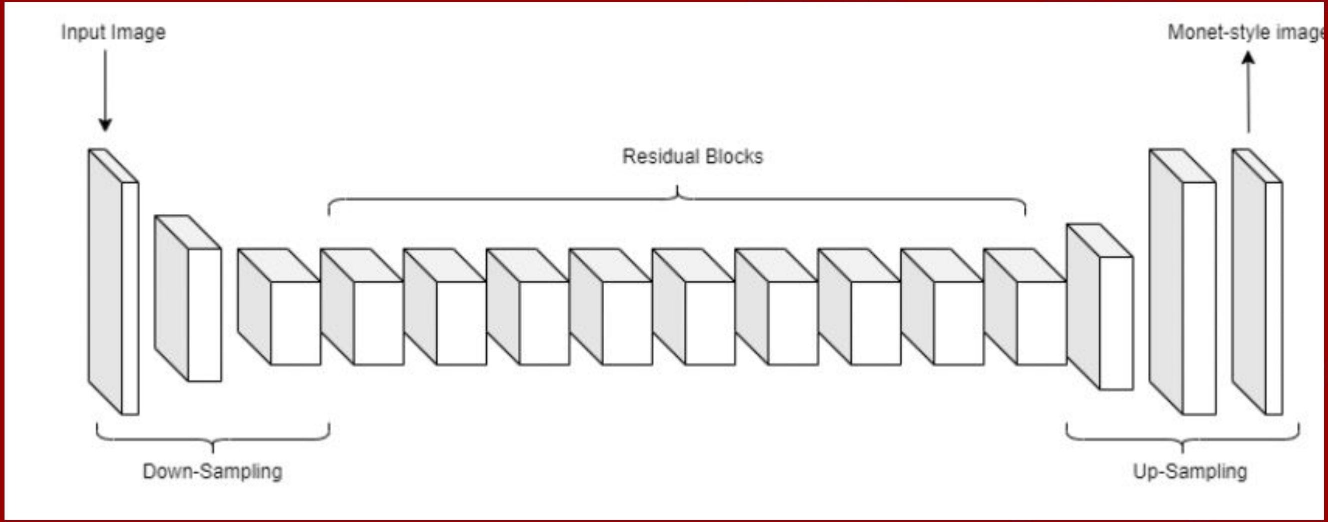
Deep Convolutional GAN - Results



Architecture - Generator

The images in the Monet dataset and the photos dataset are not paired with each other. The goal of this project is to learn a mapping between the two domains given the unpaired training samples. To achieve this we need to construct two mappings F : Photos → Monets and G : Monets → Photos.

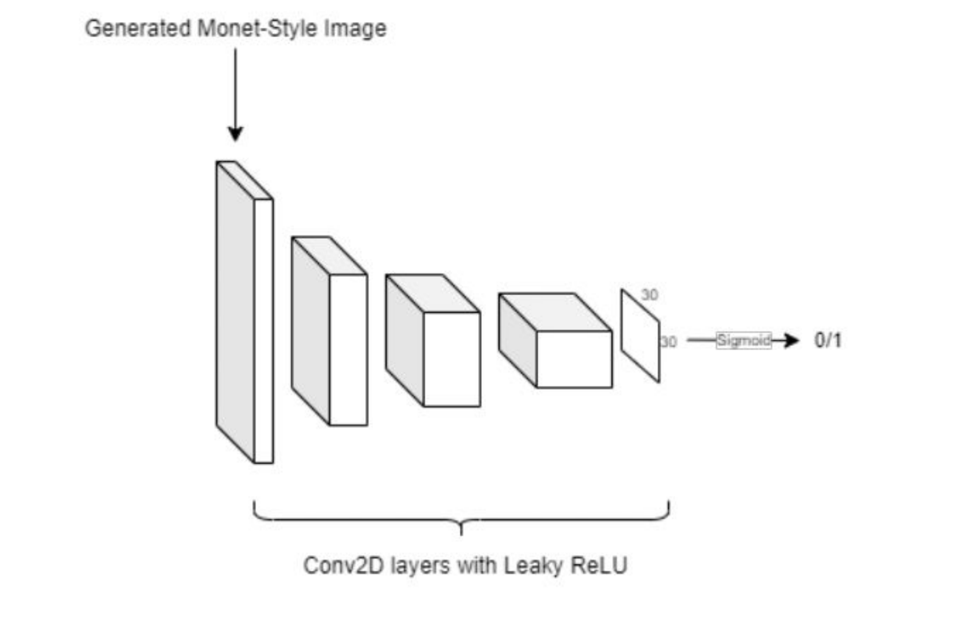
Generator Architecture: The network consists of : c7s1-64 → d-128 → d-256 → R-256 → R-256 → R-256 → R-256 → R-256 → R-256 → R-256 → R-256 → R-256 → u-128 → R-64 → cs71-3



Generator Architecture

Architecture - Discriminator

Discriminator Architecture: The discriminator architecture consists of : C-64 → C-128 → C-256 → C-512



Discriminator Architecture

Architecture

Loss Function :

- **Adversarial Loss** -------

$$\mathcal{L}^{(D)} = \mathbb{E}_{x\sim p_{data}(x)}[(D(x)-1)^2] + \mathbb{E}_{x\sim p_{model}(x)}[(D(G(y))-1)^2]$$

- **Cycle Consistency Loss** ----

$$\mathcal{L}_{cyc} = \mathbb{E}_{x\sim p_{data}(x)}[||F(G(x))-x||_1] + \mathbb{E}_{y\sim p_{data}(y)}[||G(F(y))-y||_1]$$

- **Identity Loss** : This loss helps preserve the color space between the input and output

Left to Right: Input Photo, Monet style painting using CycleGAN without Identity Loss, Monet style painting using CycleGAN with Identity Loss. Identity mapping loss helps preserve the color space of the input photograph.

Training

Parameters:

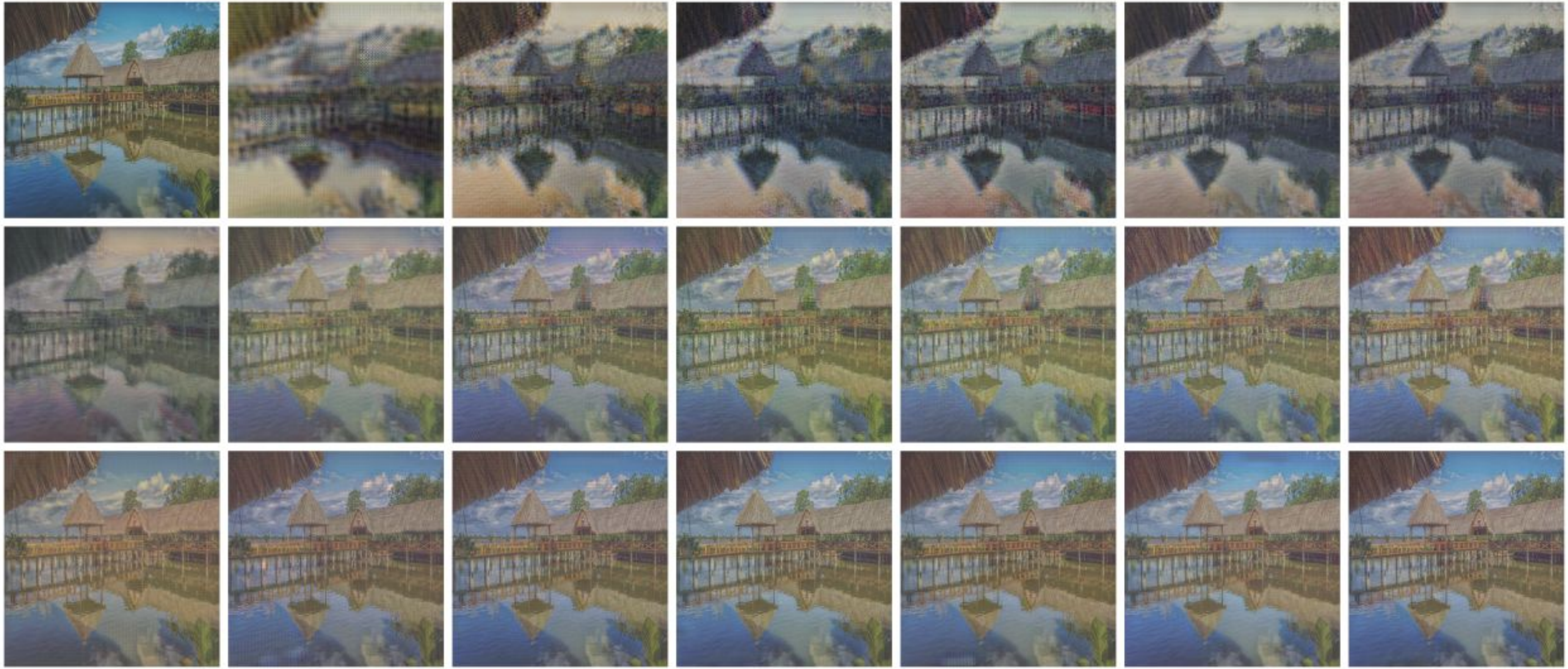
Parameter	Values
Optimizer	Adam
Learning Rate Generators	10^{-5}
Learning Rate Discriminators	10^{-5}
Batch Size	1
Betas for Adam Optimizer	$\beta_0 = 0.5 \beta_1 = 0.999$
Number of GPUs	1
Discriminator steps per generator	1
Weight Initialization	$\mathcal{N}(0, 0.02)$
Normalization	Instance Normalization [4]
Cycle Consistency Loss - λ	10
Identity Loss	$0.5 * \lambda$
Input Dimensions	$3X256X256$
Output Dimensions	$3X256X256$

Training Details:

- First 30 epochs without identity loss - to speed up process
- Next 30 epochs incorporating identity loss

Parameter	Description
GPU	Tesla V100
GPU Memory	16GB
Number of CPU cores	8
System RAM	64GB
Time per epoch (W/o Identity Loss)	15 minutes
Time per epoch (W/ Identity Loss)	20 minutes
Cost per hour	$0.8 per hour (Spot Instance pricing)
Cost for 100 epochs	$30
Time to train	34 hours

Training and Results



Thank You