NFTGan

Convert your drawing into masterful artwork

And sell it as NFT

Project Objective

Our objective is to create a web app that converts a drawing into an artwork of famous artistes so that users can sell it as NFT

We hope to:

1. Understand how GAN works
2. Productionise the models into practical application using Tensorflow.js
3. Improve existing GAN models with more suitable datasets or the model's structures

Illustration

Project Scope

1. Train, evaluate and improve 2 GAN models (Pix2Pix and CycleGAN)
2. Implement a web drawing app that translate sketch into cartoon (Pix2Pix)
3. Translate cartoon into an art piece (CycleGAN)
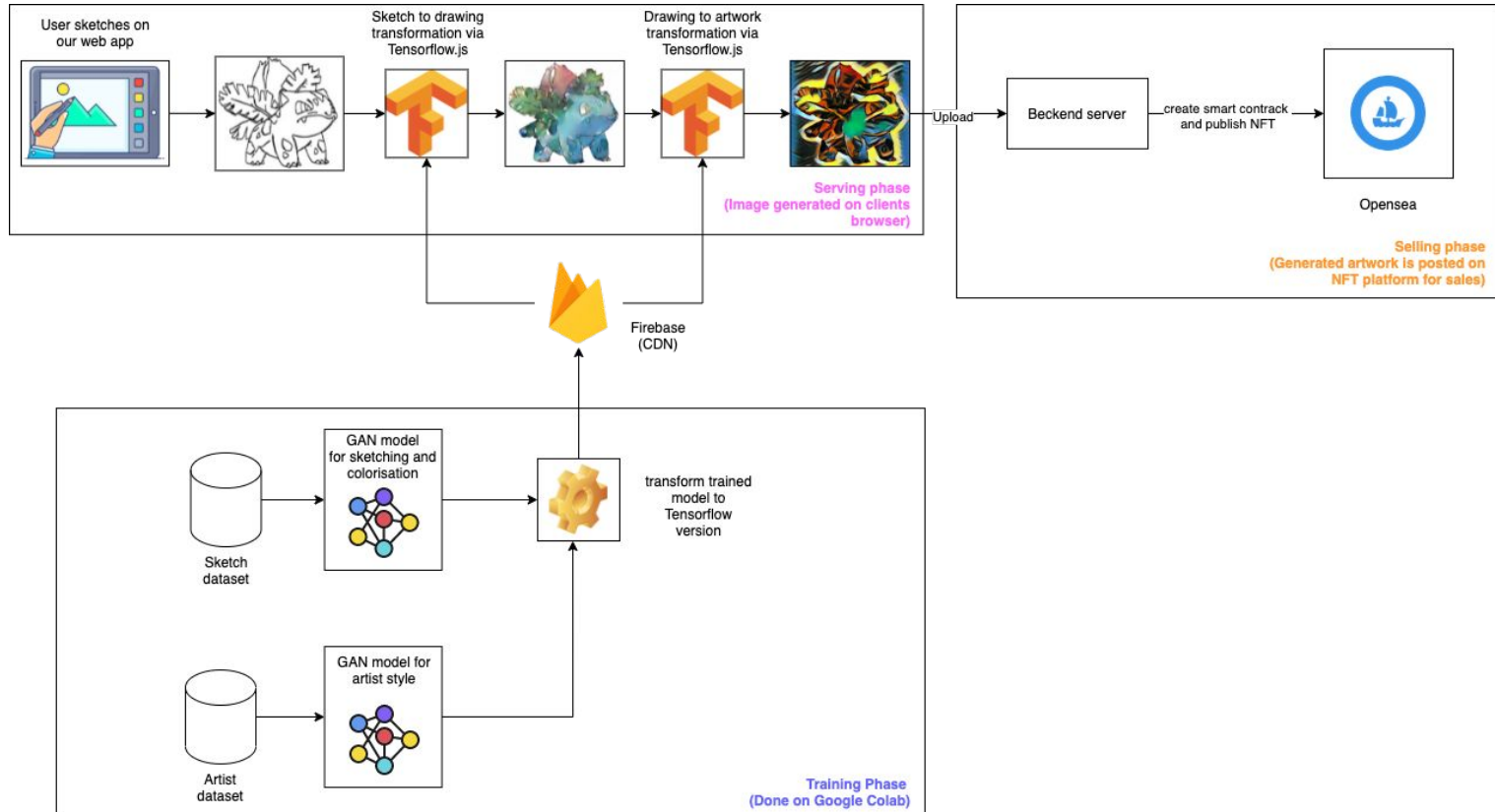4. Publish it on NFT platform like OpenSea

Data Resource Required / Available

1. Sketch to drawing dataset
   - To train Pix2Pix
2. Famous art dataset
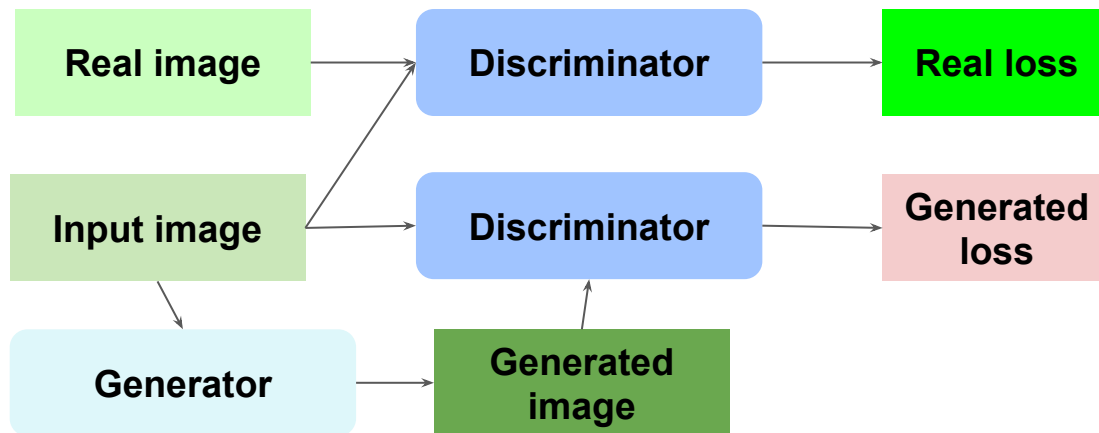   - To train CycleGAN

Ways to get data:

1. Kaggle (Best option)
2. Self sourced images (Backup)

Technical Approach (System Design)

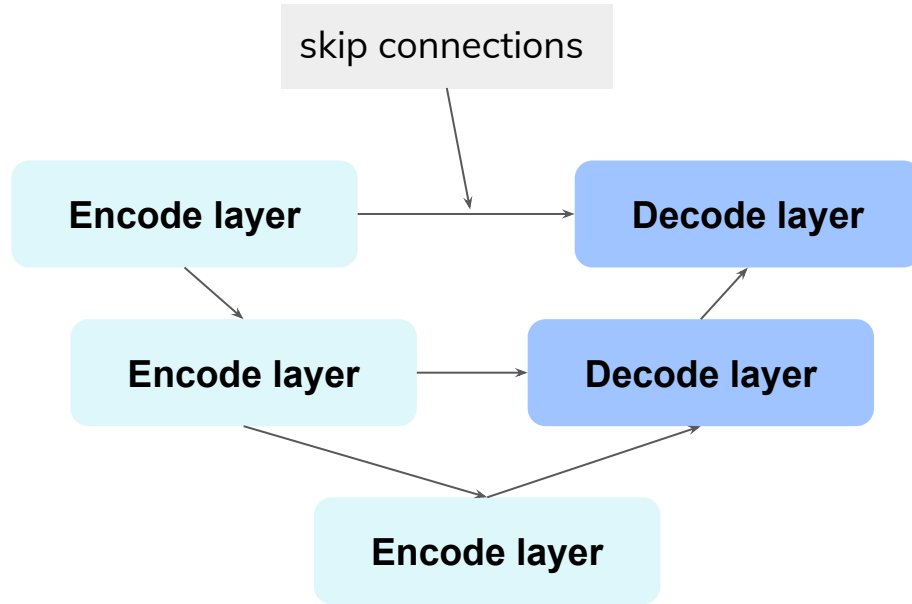# Technical Approach (Pix2Pix)

Pix2Pix(CGAN)

pix2pix reference: https://arxiv.org/abs/1611.07004

Technical Approach (Pix2Pix)

Generator(U-Net)

Encode layer: Generate smaller representation
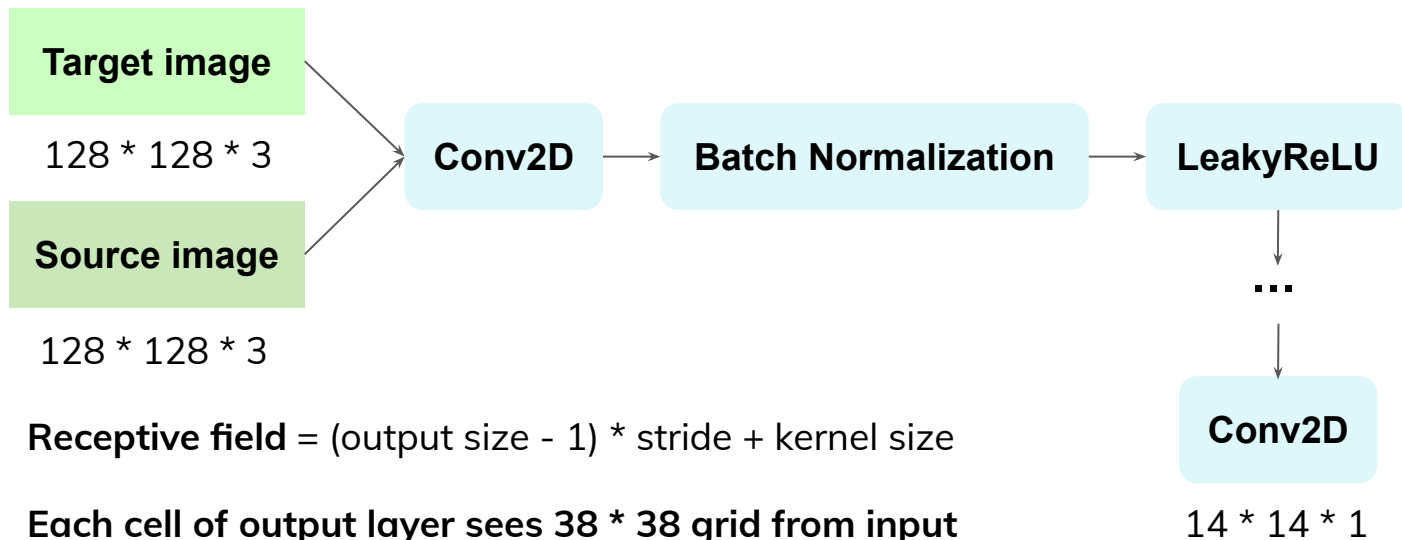
Decode layer: reverse the action of the encoder layers

skip connections

Encode layer

Encode layer

Encode layer

Decode layer

Decode layer

U-net reference: https://arxiv.org/abs/1505.04597

Technical Approach (Pix2Pix)

Discriminator(PatchGAN)

Convolutional Neural Network

Target image

128 * 128 * 3

Source image

128 * 128 * 3

Conv2D → **Batch Normalization** → **LeakyReLU**

...

Conv2D

14 * 14 * 1
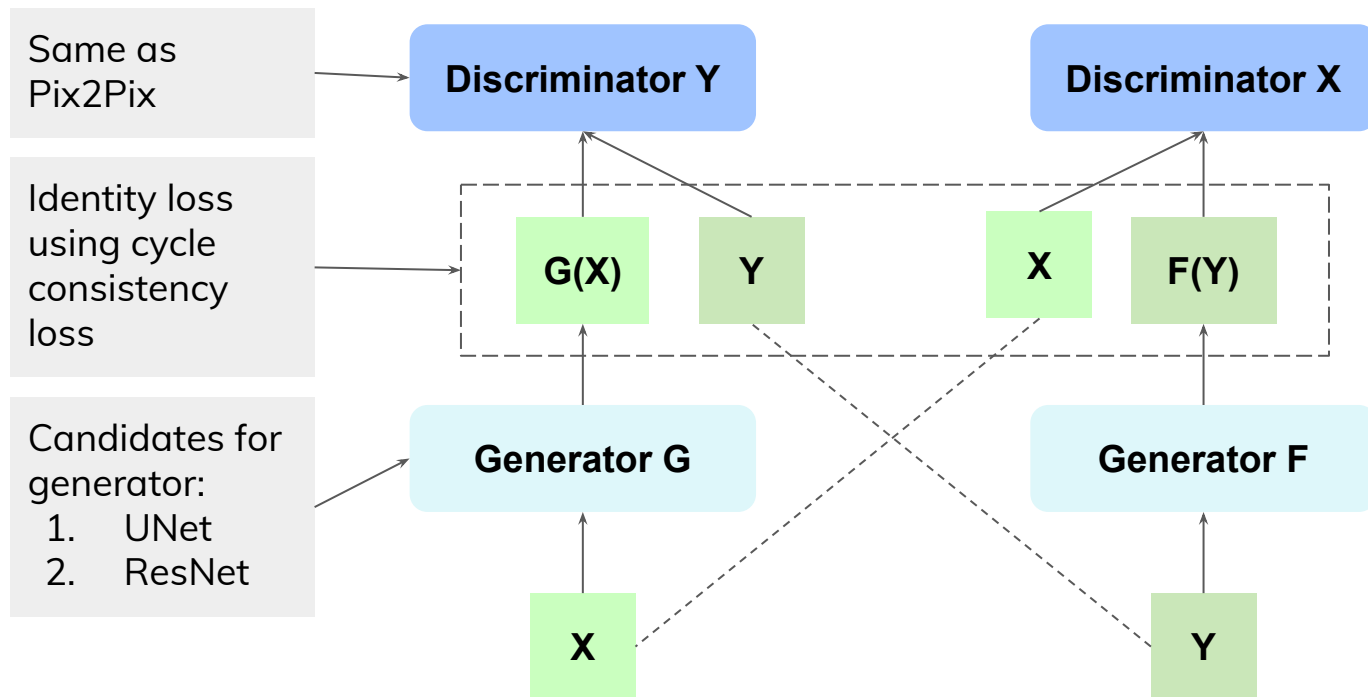
Receptive field = (output size - 1) * stride + kernel size

Each cell of output layer sees 38 * 38 grid from input layer

Technical Approach (CycleGAN)

Same as Pix2Pix

Identity loss using cycle consistency loss

Candidates for generator:
1. UNet
2. ResNet

Discriminator Y

Discriminator X

G(X) Y   X   F(Y)

Generator G   Generator F

X   Y

CycleGAN reference:
https://www.tensorflow.org/tutorials/generative/cyclegan

Things Done So Far

Tried both models to set baseline

Tested a sample flow of how our GAN models might work

Exported our initial GAN models into Tensorflow.js