Image Processing using GANs

By: Amanda Ahn

Image Processing

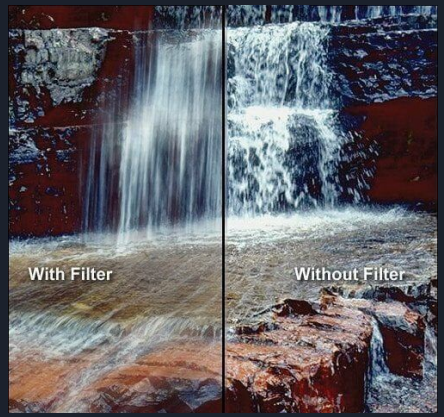
Digital image processing is only growing, and the creation and application of filters play a crucial role in enhancing and manipulating images for various purposes.

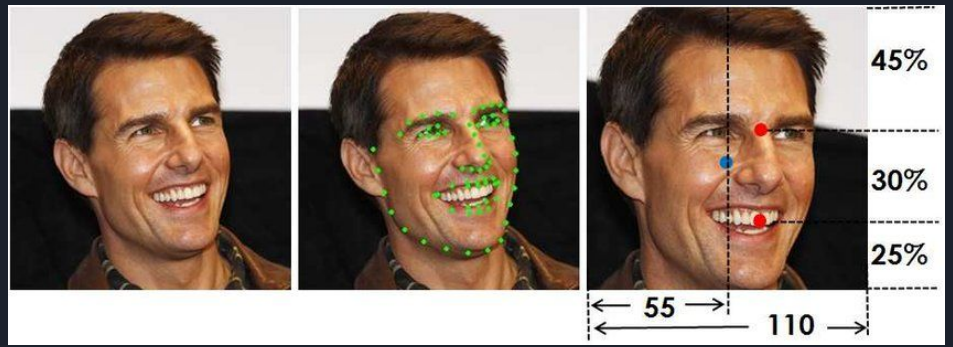
- Social media
- Video/Photo editing
- Design
- Telecommunication

Problem

Goal: Explore and develop an image processing filter that manipulates the individual components of an image, while enhancing the quality, aesthetics, and utility.

- ★ GANs
- ★ Effect of Image Processing/Manipulation





Overview

- ★ Style mapper
- ★ GAN inversion
- ★ StyleGAN

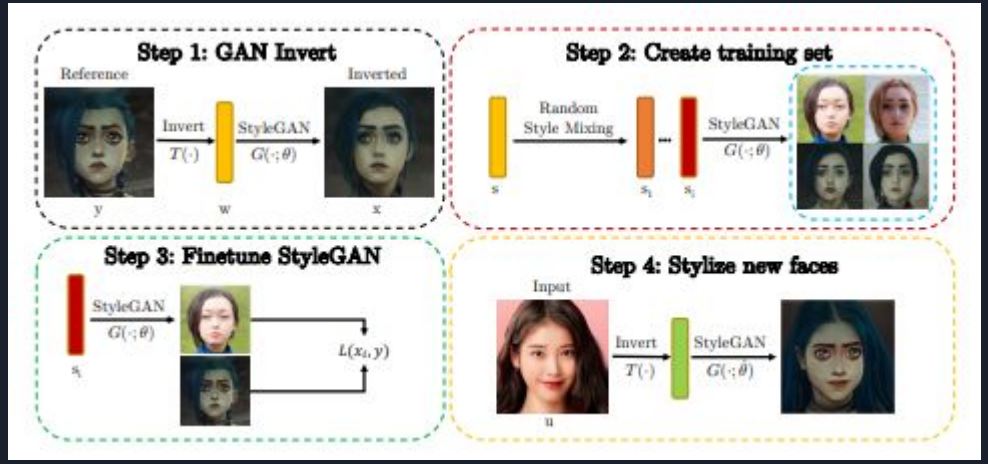
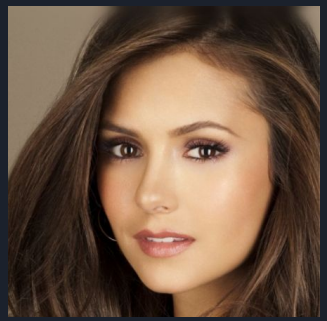


Figure taken from JoJoGAN: One Shot Face Stylization paper

Data

- ★ JoJoGAN
- ★ E4e StyleGAN
- ★ Input image
- ★ Input styling images

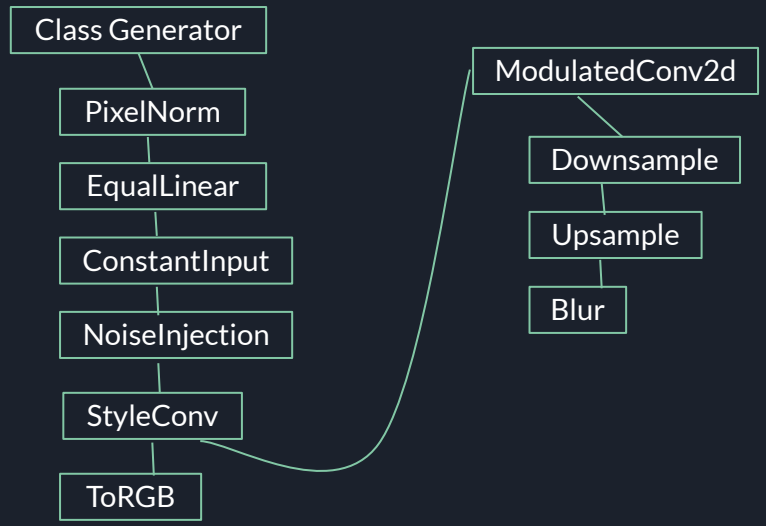


Networks

Create Generator Network:

- Class involving multiple layers to progressively create a higher resolution image
- ‘__init__’ constructor
    - (self, size, style_dim, n_mlp, channel_multiplier, blur_kernel, lr_mlp)

Function Layers:



Networks

Create Discriminator Network:

- Class that serves to distinguish between real images and the newly generated images, it takes in image inputs and produces feature maps that evaluate the realness of input images
- ‘__init__’ constructor
    - (self, size, channel_multiplier, blur_kernel)
- Functions:
    - log_size
    - ConvLayer
    - ResBlock
    - Forward method

Predictor

- Performs a style transfer on an input image using pre-trained or generator based styles.
- Pretrained:
  - Art, arcane_multi, sketch_multi, arcane_jinx, arcane_caitlyn, jojo_yasuho, jojo, disney
- Not pretrained: input 4 style images.
- Util python file

Training & Fine Tuning

Train the GAN by iteratively updating the generator and discriminator networks.

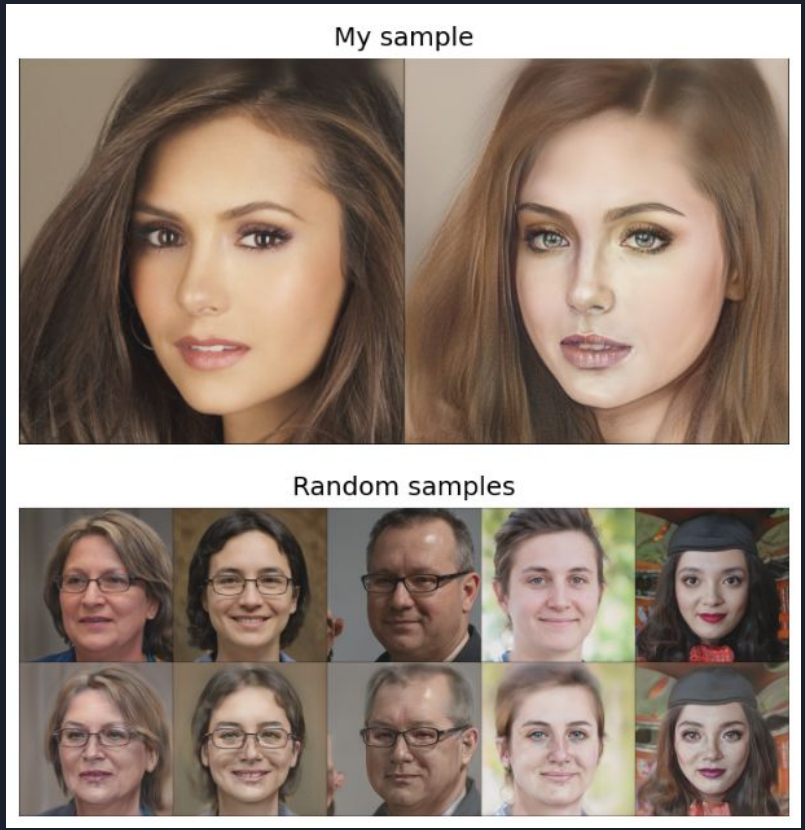
- GAN inversion
- Runtime for training: 1 min

Fine-tune StyleGAN

- Discriminator & loss function
- Progressive growing
- Runtime for fine-tuning: 12 min

Evaluation & Results

- ★ My input image
- ★ Random Samples



Future Work & References

- ★ Continue to fine-tune and improve styleGAN to counter unwanted blur effects
- ★ Create a application or website utilizing project for non-technical users

- ★ Code borrowed from JoJoGan and e4e github repositories

@article{chong2021jojogan,
title={JoJoGAN: One Shot Face Stylization}, author={Chong, Min Jin and Forsyth, David},
journal={arXiv preprint arXiv:2112.11641}, year={2021}}

Thank you for listening to my presentation ☺️