StyleGAN Playground

ANLY 590 Final Project Presentation
12/06/2022

Group Members:

Dian Zhi

Ruobing Yan

Muwen You

Zihang Weng

Hanshen Jing

Overview

- Introduction
- GAN Model
  - Traditional GAN
  - StyleGAN
- Data
- Results

Introduction

What's our goal?

- Understand, implement and train the StyleGAN model to create a new image by mixing photos from two different styles.

Fig 1. StyleGAN Mixing Example

Introduction

StyleGAN

- A generative adversarial network (GAN) introduced by Nvidia researchers in December 2018.
- In February 2019, Uber engineer Philip Wang used the StyleGAN architecture to create thispersondoesnotexist.com, which displayed a new face on each web page reload.

Fig 2. Person Does Not Exist

Generative Adversarial Networks (GANs)

- GAN is a combination of two networks: a generator and a discriminator.
- The **generator** learns to generate plausible data. It learns to make the discriminator classify its output as real.
- The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.

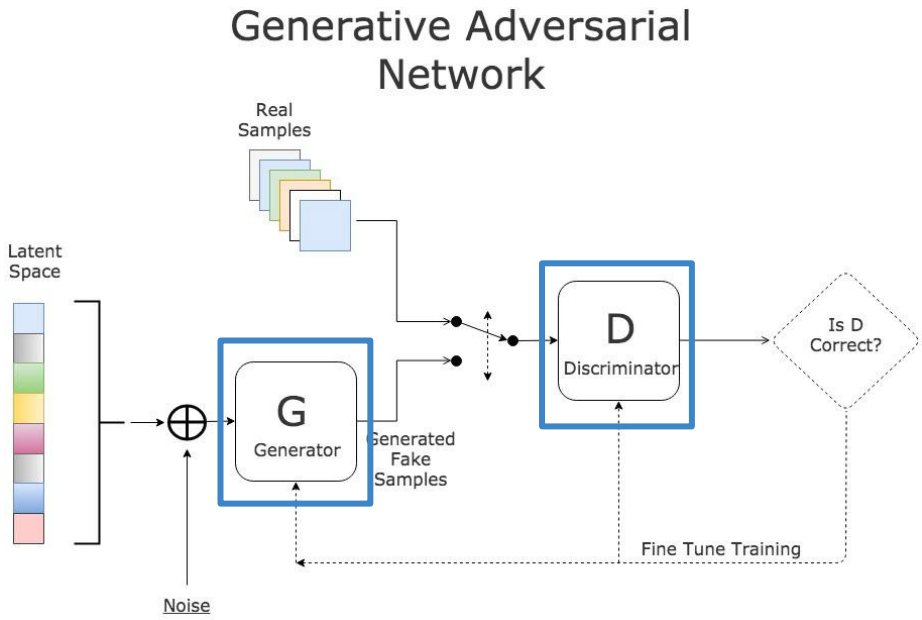


Fig 3. GANs Structure

More About GANs

Issues:

- **Non-convergence**: the models do not converge and worse they become unstable.
- **Mode collapse**: the generator produces limited modes.
- **Vanishing Gradients:** the generator training can fail due to vanishing gradients.

Different Types:

- Conditional GAN (CGAN)
- CycleGAN
- Deep Convolutional GAN (DCGAN)
- **StyleGAN**
- ……

StyleGAN

- The **fully connected network** in acts as a nonlinear mapping function.
- The **style module AdaIN** transfers the features of the w vector so that it can be used as input in the synthesis network.
- **Noise** is added into the network to generate stochastic details. It is unrelated Gaussian noise.
- **Constant** is used as fixed input to the first convolutional layer.

$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i},$$

Fig 5. AdaIN formula

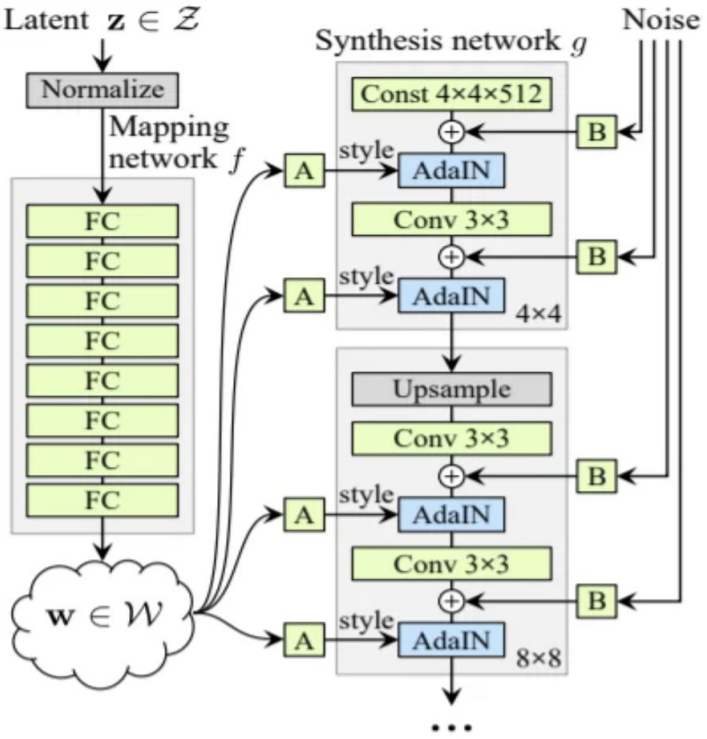


Fig 4. StyleGAN Structure

Image Data Source

Fig 6. The cover of Arcane

Arcane: league of legends (S1):

- Random screenshots of over 7000 images.
- Manually selected around 1000 images.

Fig 7. The Cover of CelebA

Celebrity Faces (Kaggle):

- Over 200k images of celebrities with 40 binary attribute annotations.
- Manually selected around 1000 images.

Data Access Link: https://drive.google.com/drive/folders/1PnlRTeZTBMRbPQcjPZIiQEwgNpmL_UT0?usp=share_link

Data Processing

Images Pre-processing:

- Face detection using Haar cascade frontal face classifier.
- Crop a sub-image from the image with face detection boxes.
- Go through all the images and manually selected 1000 images from each dataset.

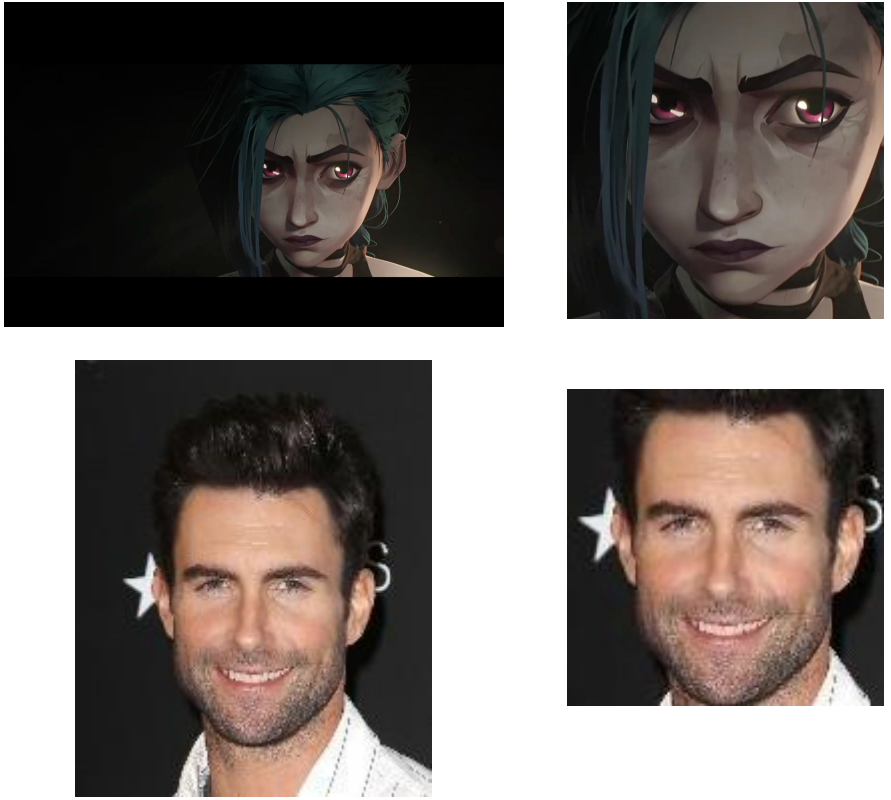


Fig 8. Cropped images comparison

Code Access Link: https://drive.google.com/drive/folders/1PnlRTeZTBMRbPQcjPZliQEwgNpmL_UT0?usp=share_link

Result Evaluation

- Mapping network
  - convert image to feature vector w in feature space
- Generator
  - use feature vector w to recover the original image
- Discriminator
  - discriminate the generated image vs original image

Result Evaluation

w_new = (feature vector w1 + feature vector w2) / 2

When feed w_new into generator, the generated image should have features from both w1 and w2

Feature/Style mixing

Feature Mixing

(w1 + w2) / 2

w1

w2

Fig 9. Left: generated celebrity face 1; right: generated celebrity face 2; middle: transition from face 1 to face 2

Style Mixing

(w1 + w2) / 2

w1

w2

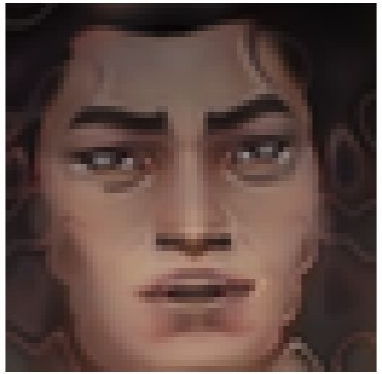


Fig 10. Left: generated celebrity face; right: generated arcane style face; middle: transition from celebrity face to arcane style face

Result Evaluation

StyleGAN truncates the intermediate vector w, forcing it to stay close to the “average” intermediate vector

w_new = w_avg + ψ(w - w_avg)

ψ: truncation_psi

between 0 (w_new = w_avg) and 1 (no truncation)

Leftmost column:
Celebrity faces

Topmost row:
Arcane character faces

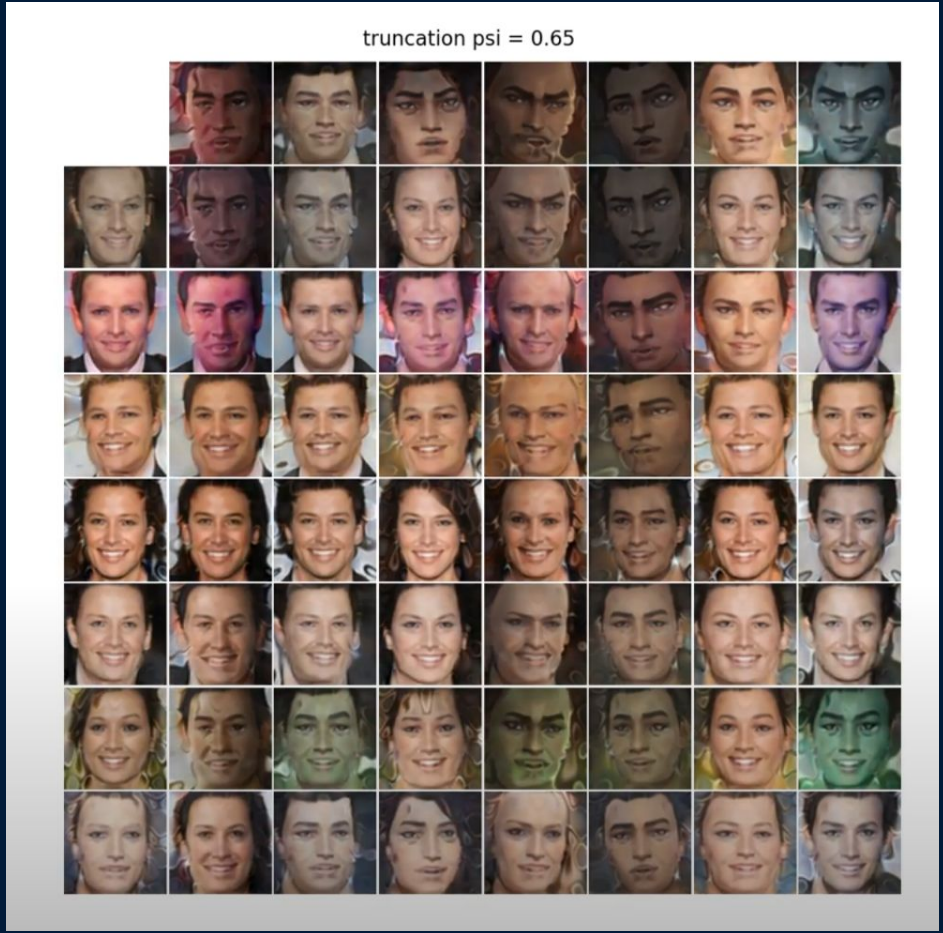
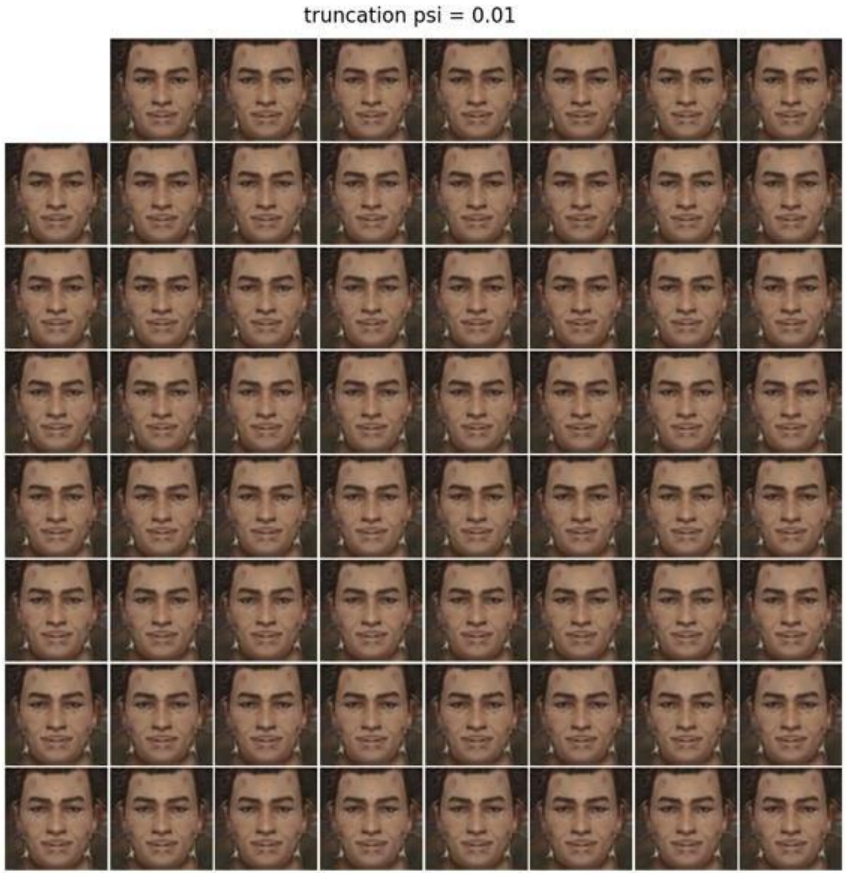


Fig 11. Style mixing matrix. Images in the matrix were generated with combined feature vectors from row and column images

Leftmost column: Celebrity faces

Topmost row: Arcane character faces



Thanks for Watching!
Q & A

Reference

Fig 1: https://www.geeksforgeeks.org/stylegan-style-generative-adversarial-networks/

Fig 2: thispersondoesnotexist.com

Fig 3:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fpaperswithcode.com%2Fmethod%2Fgan&psig=AOvVaw1JFJifod8v1P6xcQFV3hDm&ust=1670361670460000&source=images&cd=vfe&ved=0CA8QjRxqFwoTCODN7KG04_sCFQAAAAAdAAAAABAE

Fig 4 & 5: https://cv-tricks.com/how-to/understanding-stylegan-for-image-generation-using-deep-learning/

Fig 6: https://www.imdb.com/title/tt11126994/

Fig 7: https://www.kaggle.com/datasets/jessicali9530/celeba-dataset

Fig 8: https://drive.google.com/drive/folders/1PnlRTeZTBMRbPQcjPZliQEwgNpmL_UT0?usp=share_link