Pokemon Generator

Soomi Lee
Seoul National University
dazzlingpalmtree@gmail.com

Adam Lee
Seoul National University
alee00@snu.ac.kr

Wonjun Choi
Seoul National University
reynie1514@snu.ac.kr

1. Overview of Project Idea

In this project, we propose to use recently released pre-trained generative models to generate pokemons based on input images of human faces. We are planning to project the input image to learned latent space using an approach from StyleGAN2 and use the vectors here to generate our target image.

2. Short Survey of Relevant Papers

There have been several relevant papers released in different aspects of training a generative model we plan to do.

First of all regarding the deep neural network model, we will be utilizing StyleGAN2 [3] which was released last year. StyleGAN2 builds on its predecessor StyleGAN with several new features. A new normalization technique called weight demodulation replaces the original adaptive instance normalization. An original training process is also introduced which starts off training with a focus on low-resolution images then gradually shifting towards high-resolution images.

Several techniques have been proposed for effectively fine-tuning pre-trained generative models. FreezeD [4] suggests to freeze the early layers of the discriminator for effective transfer learning. And in order to prevent the discriminator of a generative adversarial network from overfitting when trained with a small amount of data, a new technique called adaptive discriminator augmentation mechanism [2] has been proposed to significantly stabilize the training. This is interesting research that helps mitigate overfitting in GAN. The approach here is to apply a wide range of augmentations to prevent the discriminator from overfitting.

We will also be using a recently released data augmentation technique called Waifu2x which transforms noisy and small images to super-resolution images for Anime-style art using convolutional neural networks.

Existing GAN models are capable of generating various high quality images. However they have little direct control over image content. In order to have more interpretable control over content, we will be using techniques introduced in GANSpace [1]. This technique identifies important latent directions by applying PCA either in feature space or latent space.

Another interesting technique that utilizes information from latent space is an algorithm called SeFa (Semantic Factorization) [5]. SeFa discovers latent semantic meaningful dimensions learned by GANs. This approach provides a way to create much more versatile concepts across various GAN models.

3. Potential Data sets for Experiments

For transfer learning we will use the pokemon dataset from Kaggle named Pokemon Images Dataset, which is made by user kvpratama. The dataset contains 819 Pokemons, unlabeled. Each Pokemon is 256*256 size jpg file with white background color.

References

[1] Erik Hrknen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls, 2020. 1

[2] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020. 1

[3] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020. 1

[4] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans, 2020. 1

[5] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans, 2021. 1

Premodern Japanese Art Style Face Generator

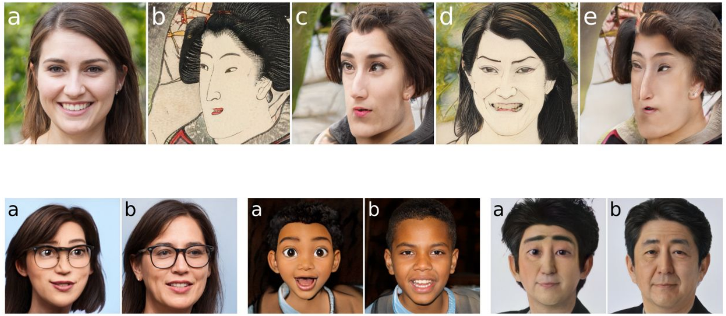
Soomi Lee, Adam Lee, Wonjun Choi

Contents

Soomi Lee, Adam Lee, Wonjun Choi

Motivation

- **Generative Adversarial Networks (GANs)** were released in 2014 by Ian Goodfellow and have since become the state-of-the-art architecture for generating high-quality images.
- **StyleGAN2-ADA (2020)** is one of several iterations of the GAN network with improvements in both the generator and discriminator network.
- Our goal is to fine-tune StyleGAN2-ADA with kaokore images to generate pre-modern japanese art style faces with animations.



Example of Cartoon Generation with GAN

StyleGAN2

- StyleGAN2 builds on its predecessor StyleGAN with several new features.
    - A new normalization technique called *weight demodulation* replaces the original adaptive instance normalization.
    - An original training process is also introduced which starts off training with a focus on low-resolution images then gradually shifting towards high-resolution images.

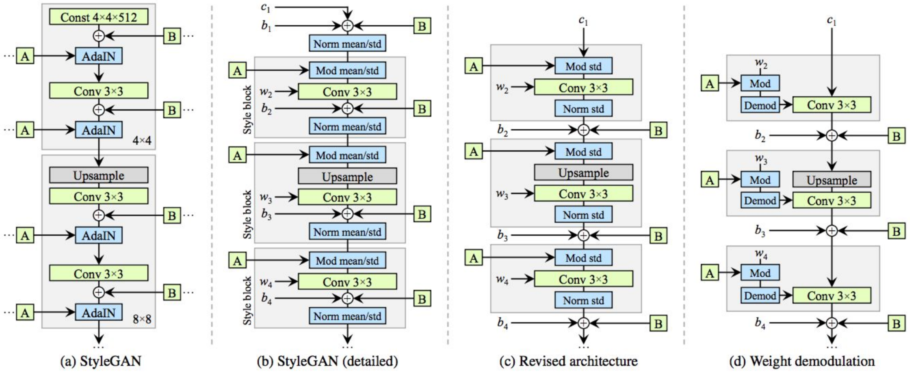


Fig 1. Architecture of StyleGAN2

Fig 2. Sample Images

StyleGAN2-ADA

- As the second iteration of the StyleGAN architecture, StyleGAN2-ADA builds on top of the previous version, StyleGAN2, adding an adaptive discriminator augmentation mechanism which stabilizes the training by deterring the discriminator from overfitting when trained with limited data.

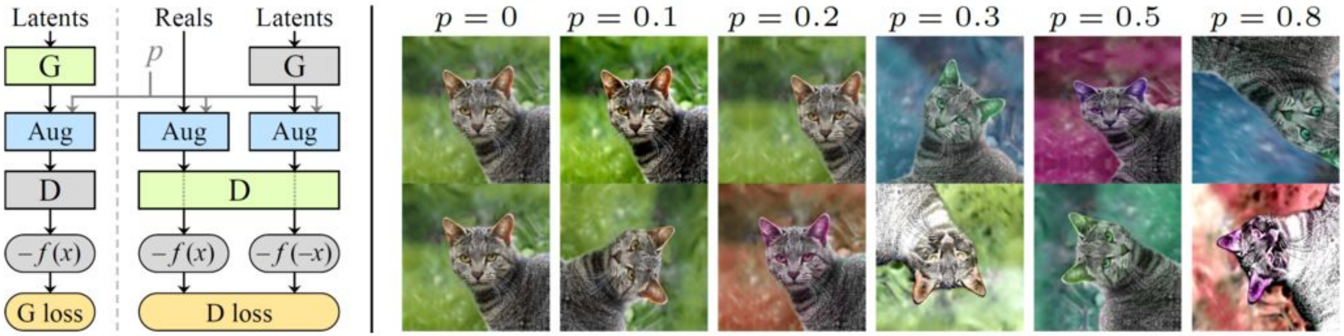
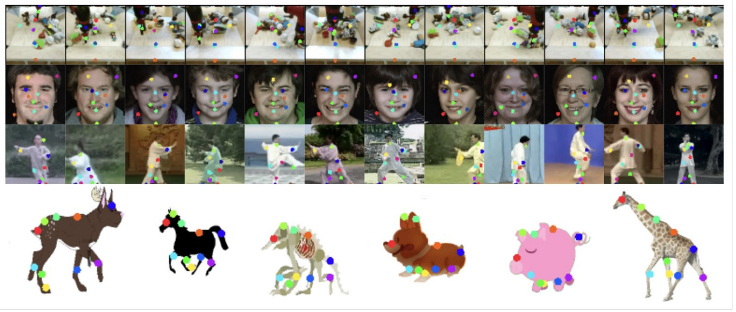


Fig 3. Architecture of StyleGAN2-ADA and Sample Images

First-Order Model

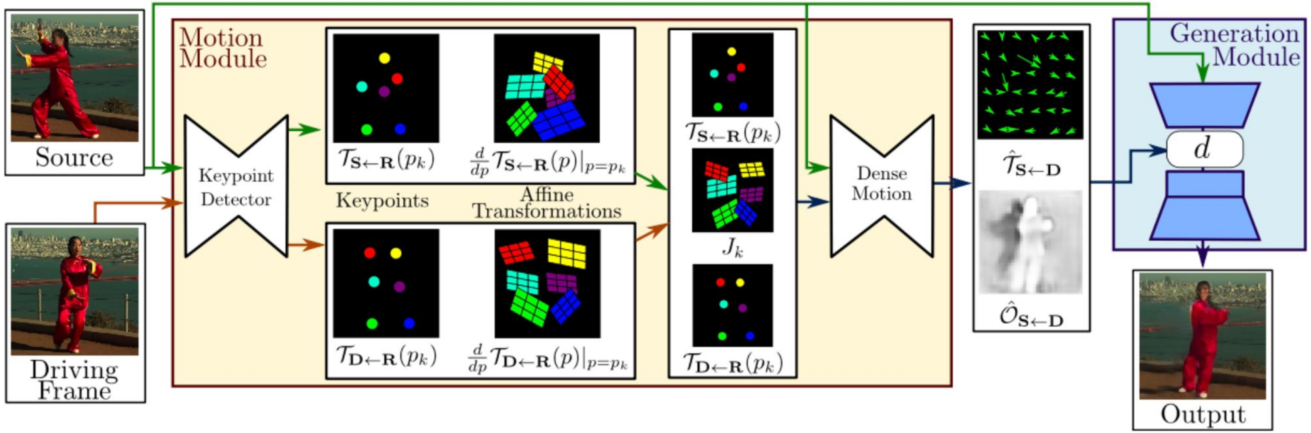
- Image animation : Task of automatically synthesizing videos by combining appearance extracted from source image with motion parts derived from driving video
  - To extract Keypoint location, GAN and VAEs have been used.
- Recently, Monkey Net is introduced, which is first object agnostic deep model.
  - But weakness is that it poorly models object appearance transformations in key point neighborhoods
- **First Order Motion Model**
  - To address that weakness, used *local affine transformation* and *key point set*.



Monkey-Net, CVPR 2019

First-Order Model

- Motion Estimation Module
- Image Generation Module
- Key Point
- Affine Transformations
- Dense Motion Field



Kaokore Dataset

KaoKore: A Pre-modern Japanese Art Facial Expression Dataset, 2020

- Faces collection of Japanese pre-modern arts
- Derived from *Collection of Facial Expressions*; Form that will be easy to use in ML tasks
- Process images and labels into ML-industry standard formats

Soomi Lee, Adam Lee, Wonjun Choi

Kaokore Dataset

- 8848 RGB images files of size 256 x 256
- There are two sets of labels
- Gender : Male, Female
- Status : Noble, Warrior, Incarnation, Commoner
- Use PyTorch

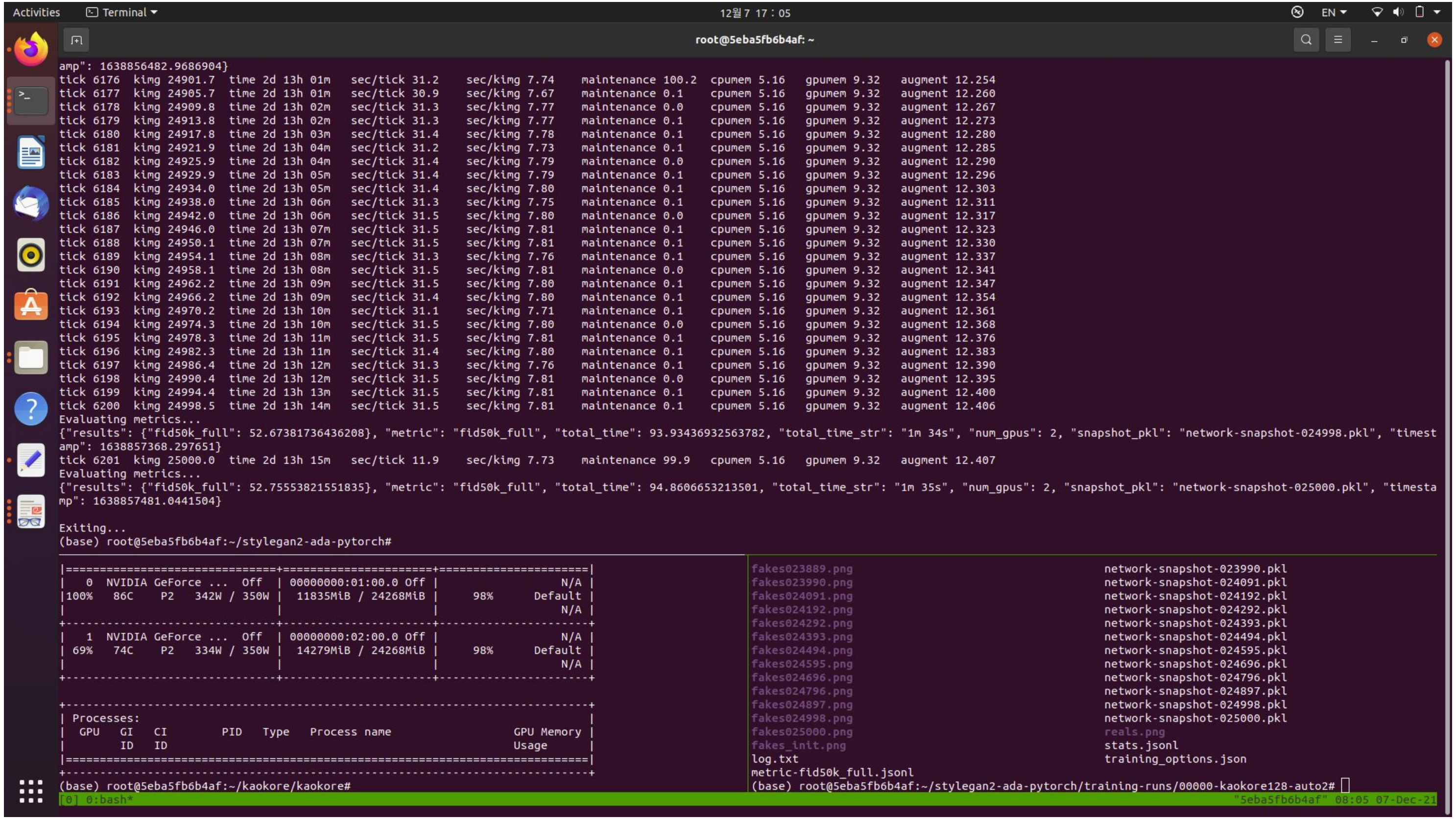
Class	*Labels*	*Examples*
gender (性別)	male (男)	
	female (女)	
status (身分)	noble (貴族)	
	warrior (武士)	
	incarnation (化身)	
	commoner (庶民)	

Kaokore Dataset

- Classification Results
- StyleGAN

Model	classifying *gender*	classifying *status*
VGG-11 [25]	92.03 %	78.74 %
AlexNet [19]	91.27 %	78.93 %
ResNet-18 [9]	92.98 %	82.16 %
ResNet-34 [9]	93.55 %	84.82 %
MobileNet-v2 [24]	95.06 %	82.35 %
DenseNet-121 [11]	94.31 %	79.70 %
Inception-v3 [27]	96.58 %	84.25 %

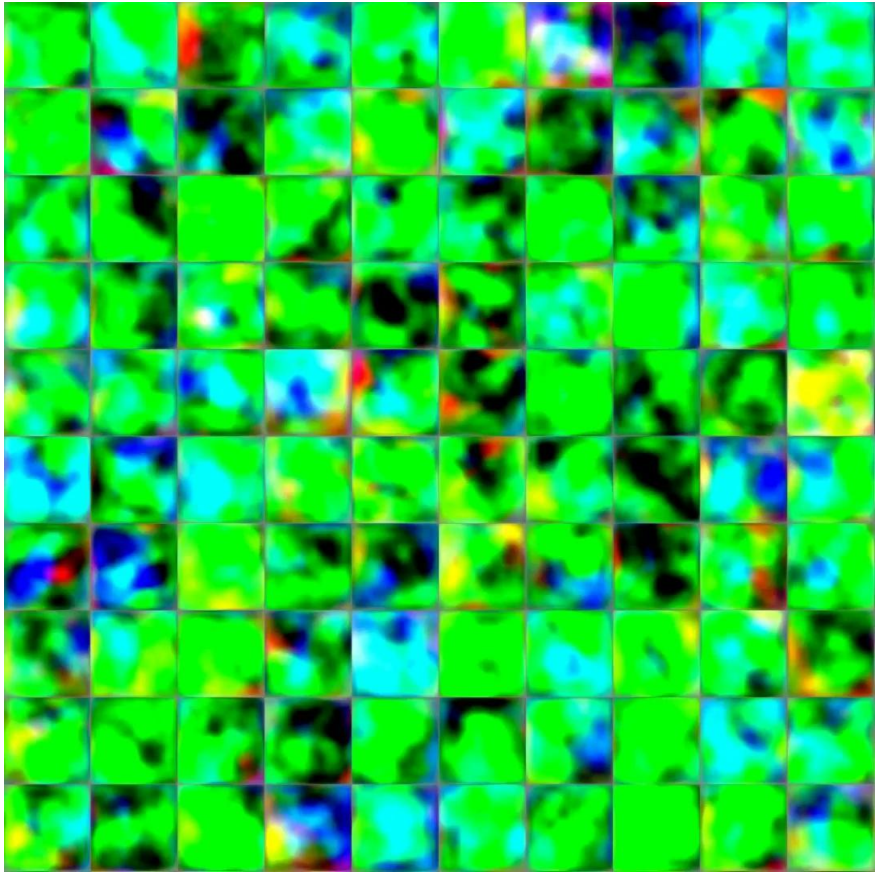
Training Process - StyleGAN2-ADA



- Resolution : 128 x 128
- GPU : 2대
- 소요 시간 : 2d 13h 14m
- thousands of real images shown to the discriminator ("kimg") : 25000kimg → 15220kimg 이후 mode collapse가 나타나 그 이전의 network 활용

Soomi Lee, Adam Lee, Wonjun Choi

Training Process - StyleGAN2-ADA



Generating face pictures using StyleGAN2-ADA

```
python generate.py --outdir=out --seeds=66,99,207,893, 173, 198, 83, 574
--network=training-runs/00000-kaokore128-auto2/network-snapshot-015120.pkl
```

Soomi Lee, Adam Lee, Wonjun Choi

Generating face pictures using StyleGAN2-ADA

```
python generate.py --outdir=out --seeds=66,99,207,893, 173, 198, 83, 574
--network=training-runs/00000-kaokore128-auto2/network-snapshot-015120.pkl
```


seed : 66


seed : 99


seed : 207


seed : 893


seed : 173


seed : 198


seed : 83


seed : 574

Projecting latent space vectors

→ Used target image from celebA dataset

1.

```
python projector.py --outdir=out --target=~/mytargetimg.png
--network=training-runs/00000-kaokore128-auto2/network-snapshot-015120.pkl
```

2.

```
python generate.py --outdir=out --projected-w=out/projected_w.npz
--network=training-runs/00000-kaokore128-auto2/network-snapshot-015120.pkl
```

or

```
python blend.py --outdir=out --projected-w=out/projected_w.npz
--network=training-runs/00000-kaokore128-auto2/network-snapshot-015120.pkl
```

→ blend.py is similar to generate.py, but it uses the weights of the ffhq model for the first few layers (generate.py is identical to the one in StyleGAN2-ada official pytorch implementation)

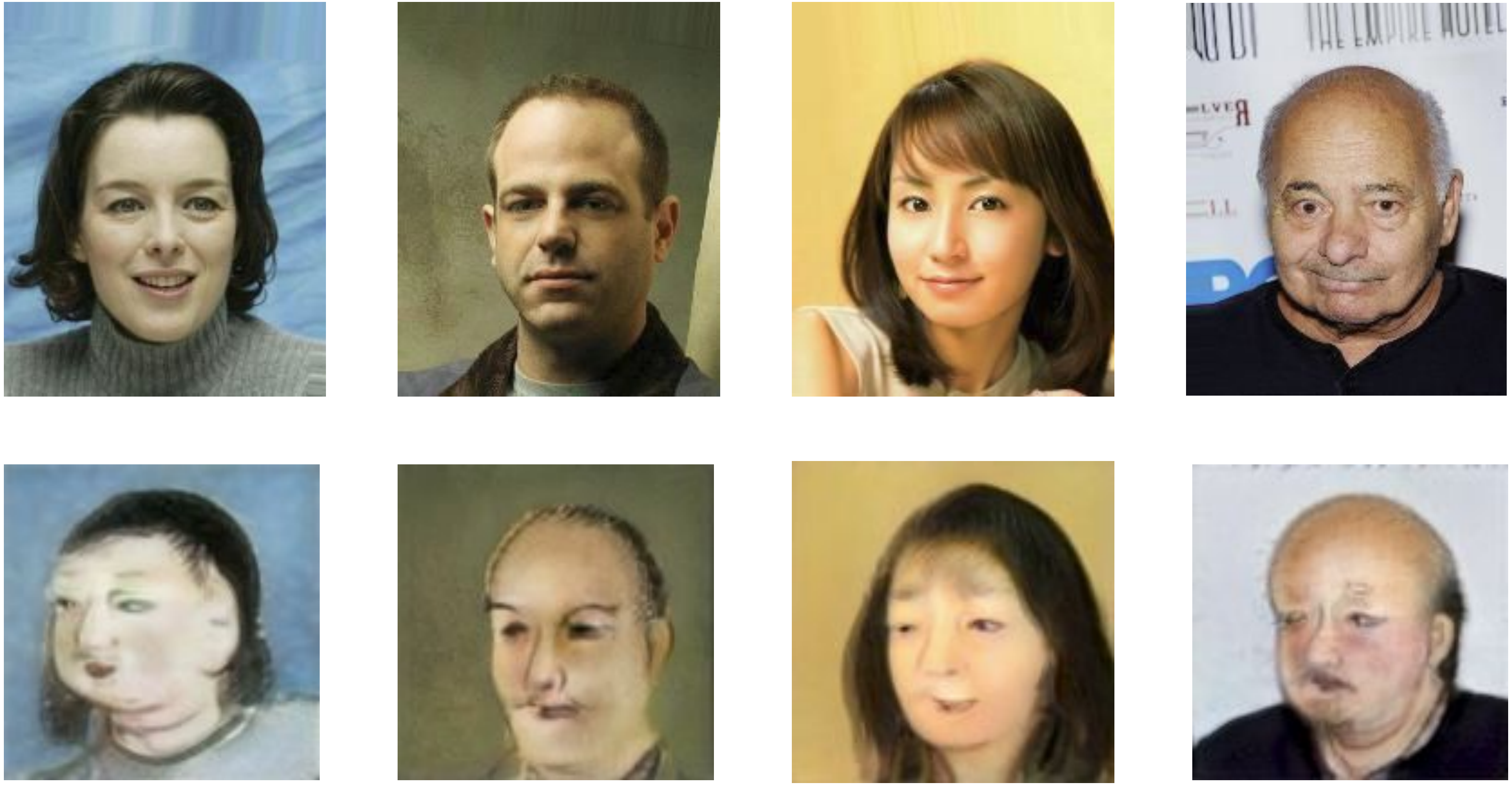
Projecting latent space vectors - StyleGAN2-ADA

Projecting latent space vectors - StyleGAN2-ADA

good

Actual

Generated



Soomi Lee, Adam Lee, Wonjun Choi

Projecting latent space vectors - StyleGAN2-ADA

bad

Actual

Generated

Adding Animation Effects - StyleGAN2-ADA + First order Model

```
python demo.py  --config config/vox-256.yaml --driving_video src_video2.mp4 --source_image seed0099.png --checkpoint vox-cpk.pth.tar --relative --adapt_scale
```

driving video

GAN generated picture to animate

Soomi Lee, Adam Lee, Wonjun Choi

Adding Animation Effects - StyleGAN2-ADA + First order Model

```
python demo.py  --config config/vox-256.yaml --driving_video src_video2.mp4 --source_image seed0099.png --checkpoint vox-cpk.pth.tar --relative --adapt_scale
```

driving video

GAN generated pictures that are animated

Adding Animation Effects - StyleGAN2-ADA + First order Model

```
python demo.py  --config config/vox-256.yaml --driving_video src_video2.mp4 --source_image seed0099.png --checkpoint vox-cpk.pth.tar --relative --adapt_scale
```

driving video

GAN generated picture that is animated

projected image that is animated

Thank You