StyleGAN Approach to Generate Anything

by
Hakki Eren Arkangil
M.S., Computer Engineering, Bogazici University, 2020

Cmpe 587 How to Generate Anything
Term Project Submitted to the
Institute for Graduate Studies in
Science and Engineering

Bogazici University
2020

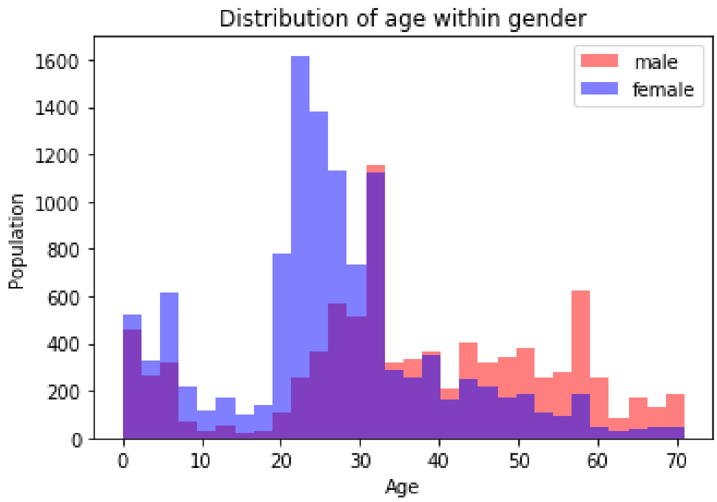
TABLE OF CONTENTS

1) INTRODUCTION

AI is not necessarily bad or terrifying as some people think. Ordinary imagination of human-ai collaboration usually end up with suffocating ''Skynet'' nightmares. However integration of artificial intelligence in the labor force is not bad as it seems. This project aims collaboration of artificial intelligence with human labor. Similar to recent mobile applications such as Snapchat or Instagram , this project is able to change photos in three direction which are gender, age and smile. Although the age direction might be so helpful in the real life, the timeline of the project is Christmas, and nobody can say no to a warm smile. The project aims to make the people smile interactively by changing their photos which are taken by any camera. It is shown in the figure 1.

Figure 1. Interactive usage of styleGAN with a camera

The projects also allows users to create their own latent spaces if they proper a dataset properly. I used the FFHQ dataset for the project which consist face photos of people in the Flickr. The further analysis of the dataset is shown in the following graphs.



The distribution shows us generator-detector model build on this dataset may be biased.

2) RELATED WORKS

Karras T, Aila T, Laine S, Lethinen J, (2018) introduced a new training methodology for generative adversarial networks [1].
Karras T, Aila T, Laine S, (2019) proposed an alternative generator for generative adverserial networks which is style-based [2].
White T, (2016) however has a different perspective to styleGAN and his methodology is widely discussed in his paper [3].

3) PROBLEM STATEMENT AND FORMULATION

Instead training a linear model I trained a non-linear model with two layers neural network for predicting age, gender and smile vectors. For a given latent vector we want to find a direction in non-linear space to become happier face images. Gradient descent is used to handle these directions. Thus, we I optimized latent vector to get happier face image as output. The generator model is explained in the following sub-topics.

Baseline Progressive GAN

The generator and discriminator models for styleGAN are trained using the progressive growing GAN training method. Both models start with small images in 4×4 pixels. The models are fit until stable, then both discriminator and generator are expanded to double the width and height (quadruple the area) to 8×8 pixels.

$$\min_{G} \max_{D} V(D, G)$$

$$V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]$$

Figure 2. Objective function of GAN

A new block is added to each model to support the larger image size, which is faded in slowly over training. Once faded-in, the models are again trained until converge and the process continues with larger image sizes until the desired target image size is met, such as 1024×1024 [1].

Bilinear Sampling

The progressive growing GAN uses nearest neighbor layers for up-sampling instead of transpose convolutional layers that are common in other generator models.The first point of deviation in the StyleGAN is that bilinear up-sampling layers are unused instead of nearest neighbor. The nearest-neighbor up/downsampling is replaced in both networks with bilinear sampling, which we implement by lowpass filtering the activations with a separable 2nd order binomial filter after each upsampling layer and before each downsampling layer.[2]

Mapping Network and AdaIN

A standalone mapping network is used that takes a randomly sampled point from the latent space as input and generates a style vector. The mapping network is comprised of eight fully connected layers. The style vector is then transformed and incorporated into each block of the generator model after the convolutional layers via an operation called adaptive instance normalization or AdaIN.

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

Figure 3. Calculation of the adaptive instance normalization (AdaIN) in the StyleGAN

The AdaIN layers involve first standardizing the output of feature map to a standard Gaussian, then adding the style vector as a bias term. The addition of the new mapping network to the architecture also results in the renaming of the generator model to a "synthesis network."

Removal of Latent Point Input

We modify the generator model so that it no longer takes a point from the latent space as input. Instead, the model has a constant 4x4x512 constant value input in order to start the image synthesis process.

Addition of Noise

The output of each convolutional layer in the synthesis network is a block of activation maps. Gaussian noise is added to each of these activation maps prior to the AdaIN operations. A different sample of noise is generated for each block and is interpreted using per-layer scaling factors.

Mixing regularization

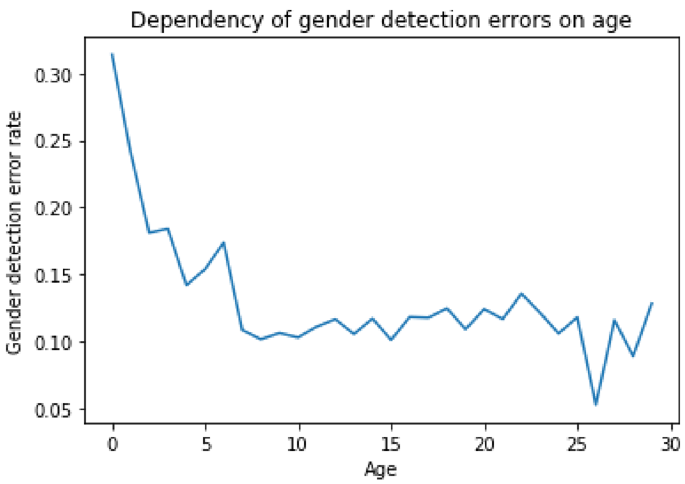
Mixing regularization involves first generating two style vectors from the mapping network. A split point in the synthesis network is chosen and all AdaIN operations prior to the split point use the first style vector and all AdaIN operations after the split point get the second style vector. This encourages the layers and blocks to localize the style to specific parts of the model and corresponding level of detail in the generated image.

4) STYLEGAN TECHNIQUE

```
StyleGAN {
Input: Human pictures
Output: New pictures with different latent representations.

    For i=1 to n
        Take picture from any source

    ->Align faces in order.

For i=1 to n
    ->Keep generating face images until finishing all aligned faces.
    ->Take a latent vector value
    ->Change image pixels accordingly.
}
```

5) PERFORMANCE EVALUATION

The model has difficulties to distinguish babies. There is drastic decrease in the gender detection error after the age 10. It is also possible that the model we used for creating ground truth produces random guesses.



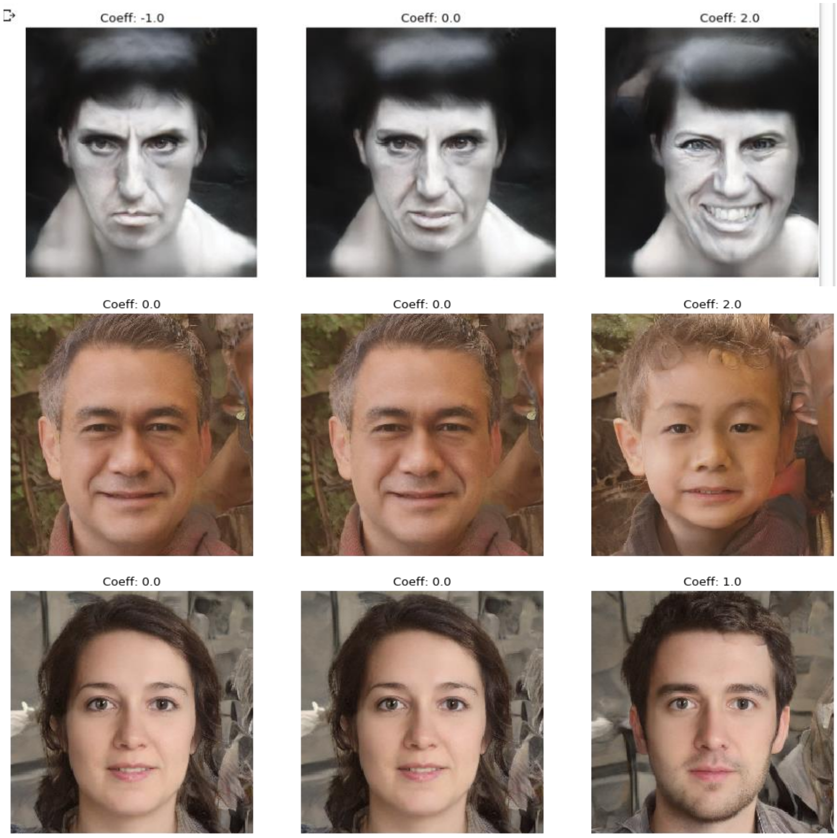
The graph below shows that 100 labeled examples is already enough to reach 80% accuracy rate, which is an eye-catching result.

The project is applied images of Bogazici University Computer Engineering Faculty members in order to perform human-AI collaboration. Some of the photos created by the model shows the quality of results. The next 3 results show images whose smile direction is changed.

The model also allows creating fun directions if it is trained with a proper data. The next example is trained with pickle dataset.

In the next result I trained the data with the high school pictures. The rightmost image is biased with neandertal pictures. The leftmost and middle one gives an idea about the limit of power of the styleGAN. The model is lacking of creativity to produce and imitate extreme cases because of the data bias.

The last three results also shows the power of styleGAN. You can change the directions even of cartoon pictures. You are not confined to smile vector, there last 2 example is a nice illustration where the directions are age and gender in order.



6) CONCLUSION AND FUTURE WORK

As it is shown in part 5, styleGAN is very successful and fulfilling tool to generate new images and making some changes on it. Even though styleGAN is so powerful, it can give bad results sometimes due to image quality or dataset. Its power limited to data. The future work aims to be able to apply styleGAN multiple faces or face images from different angles. The project also intend more easy and appliable latent-vector creator which will be researched in the later papers.

APPENDIX A

Listings of software is given to the Institute. The diskette/CD contains files such as the source code, one or more sample input and corresponding output separately.

REFERENCES

1) PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION. Karras T, Aila T, Laine S, Lethinen J. (2018). ICLR 2018.

2) A Style-Based Generator Architecture for Generative Adversarial Networks. Karras T, Aila T, Laine S (2019). NVIDIA.

3) Sampling Generative Networks. White T. (2016). Victoria University of Wellington, New Zealand.