

Generative Image Creation

Yan Liang
Project: Generative Image

Abstract—Generative Image is a API with simple .

I. INTRODUCTION

Generative Adversarial Network (GAN) is a generative model based on deep learning. More generally, GAN is a model architecture used to train generative models, and the most common is to use deep learning models in this architecture. I use the start code from StarGAN v2, which contributes a new large-scale hairstyle to generate new hairstyles. Users can use APIs to upload samples and source pictures to generate their own hairstyles.

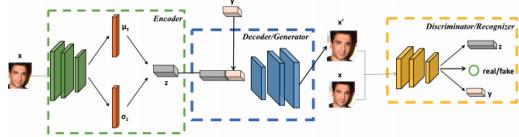


Fig. 1. Overview of Gan

II. RELATED WORK

A. Download data

Users can download the [CelebA-HQ] database and checkpoints through the shell.

B. Transform the Pictures

Transform custom pictures to match [CelebA-HQ] dataset.

C. Training networks

I using pre-trained models, users can also train their own models as needed, it's normally takes three days on a single Tesla V100 GPU.

D. Evaluation metrics

Evaluation metrics are available.

III. APPROACH

A. Upload sample hairstyles

There are some sample hairstyle picture under the reference fold, the users also can upload their own hairstyle as needed

- home page (Fig. 2). End point: (<http://127.0.0.1:8080/>)
On the home page, we can click the response button to upload files.
- Upload sample API(Fig. 3), once upload succeeded(Fig. 4). End point: (http://127.0.0.1:8080/add_sample).
- Upload source API(Fig. 5). Endpoint: (http://127.0.0.1:8080/add_src)

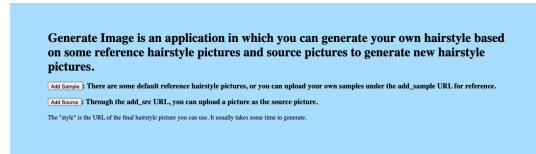


Fig. 2. Upload Sample



Fig. 3. Upload Sample

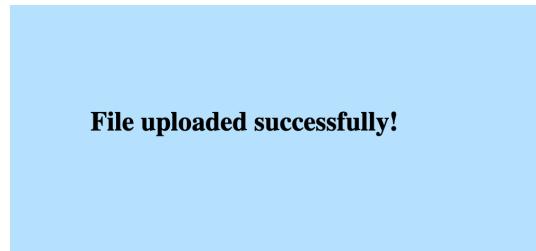


Fig. 4. Upload succeeded

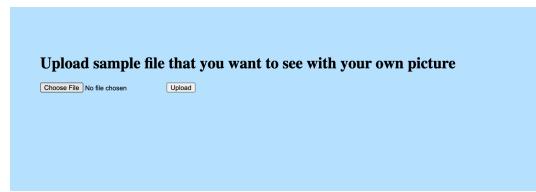


Fig. 5. Upload Source

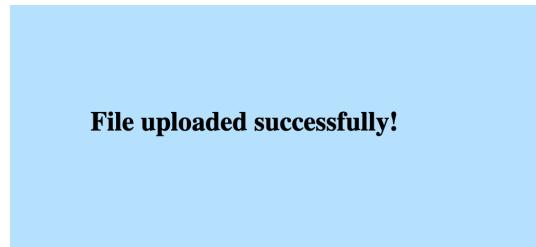


Fig. 6. Upload succeeded

Once upload succeeded(Fig. 6)

- Generate hairstyle API(Fig 6). Endpoint: (<http://127.0.0.1:8080/style>) The first row is for people who want to have new hairstyles. Except for the first row, the leftmost column is for example hairstyles, and all the rest are hairstyles generated for the source picture. It could upload more than one source picture under the same under point.

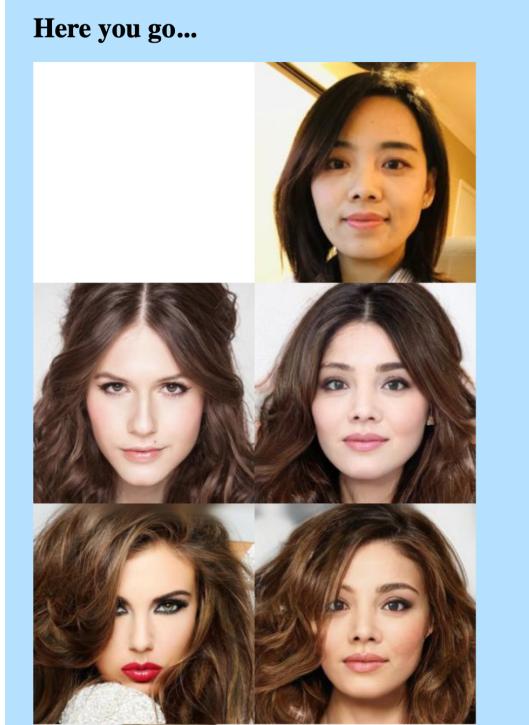


Fig. 7. Hairstyle

IV. RESULTS

A. Generate with two sources

Figure 8 corresponds with two sources three samples

B. Generate with three sources

Figure 9 corresponds with three sources and three samples.

C. Generate with more sources

Figure 10, and 11 correspond with more sources and samples.

V. FUTURE WORK

Overall, based on the result shows, the model is built on torch the qualitative results generated on CelebA looks good. The training really takes a long time.

- 1) If possible I think I would like to try more other parameters to compare the models.
- 2) The API is really flexible it can call in anywhere if needed. I consider in my free time to build another mobile app base on this project.

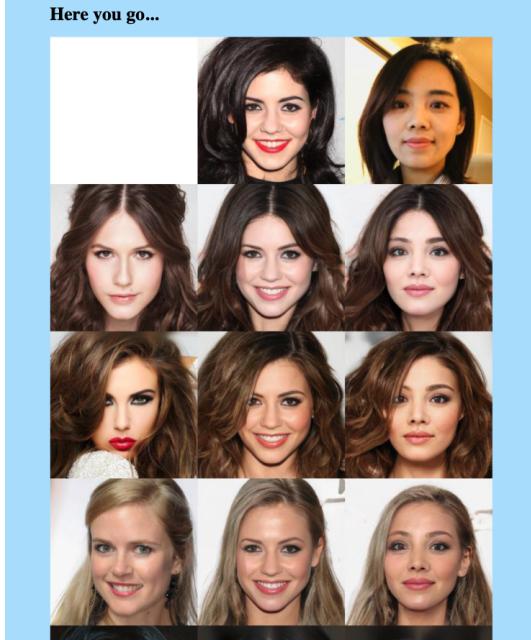


Fig. 8. result reference 1

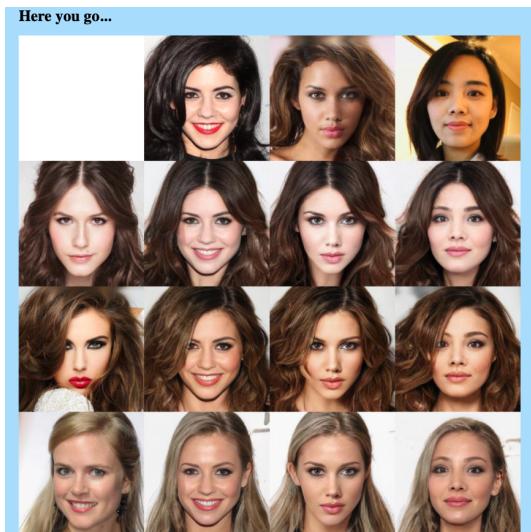


Fig. 9. result reference 2



Fig. 10. result reference 3



Fig. 11. result reference 4

VI. USAGE

- Install libraries
- Active virtual environment
- Download all the datasets
- Start server
- Call the endpoints as needed
- Generate hairstyle