Develop Journal

Face Restoration Github Link

(https://github.com/SaintWatson/Face-Renovation-Project)

Environment

- OS: Arch Linux
- Python: 3.9.5
- CUDA: 11.3
- Python Package: see *requirements.txt*

Environment Building

```
# Create a virtual environment
python3 -m venv env

# Activate your environment
source env/bin/activate

# If your pip is overtime, upgrade it.
python3 -m pip install --upgrade pip

# Install the requirement package
pip install -r options/requirements.txt
```

Prepare Dataset

Download Dataset

We are using FFHQ (https://github.com/NVlabs/ffhq-dataset) here:

```
cd dataset
git clone git@github.com:NVlabs/ffhq-dataset.git
python3 ffhq-dataset/download_ffhq.py -i
```

Sample Data

Because the original dataset is too large, we can use `dataset/DataSampler.py` to sample data like this:

```
python3 DataSampler.py <train_size> <valid_size>
```

or calling function like this:

```
from dataset.DataSampler import sample
sample(train_size, valid_size)
```

Make Degraded Data

Since we need (original_image, degraded_image) pairs for our dataset, we produce the degraded data like this:

```
python3 Degrader.py
```

The degradation includes 4 methods:

1. Down sampling *(0.25x~0.125x)*

2. Add Noise *(Gaussian/Laplacian)*
3. Blurring *(Gaussian/median)*
4. JPEG compression *(Quality 50~85)*

Dataloader

Now, the dataloader can be use like this:

```
from options.Config import *
from dataset.FaceLoader import getFaceLoader

train_loader = getFaceLoader(DataLoaderConfig(), 'train')
valid_loader = getFaceLoader(DataLoaderConfig(), 'valid')
```

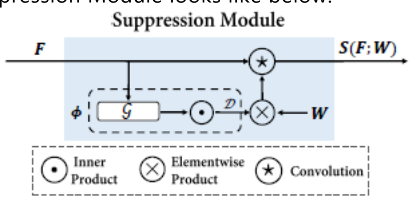
```
Below has not been fully completed, so I'll change the form to analysis report.
```

Network

Model

In this paper, generator is a multi-stage architecture formed by **suppression module** and **replenishment module**.

- Supression Module looks like below:

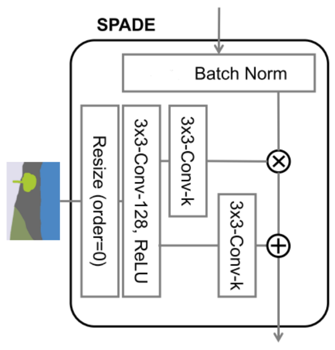


Source: https://arxiv.org/abs/2005.05005

(https://arxiv.org/abs/2005.05005))

  - $\mathcal{G}$ is a function carries the raw input vector into specific dimension. I implement it by a simple MLP (recommanded by the author).
  - $\mathcal{D}$ is an activate function, I implement it by relu.
  - ***W*** is the kernal for convolution, I have the most problem on this. I can't find a suitable shape for it, number of channel of ***W*** will affect the number of output channel of the whole module, but it is hard for the function ϕ to output a tensor that can match the shape of ***W*** simply use a MLP. I 've tried many method to fit this problem like revising the architecture of $\mathcal{G}$, but they still can't do elementwise product. That's why my implamantation is stopped here.

- Replenishment Module:



Source: https://arxiv.org/abs/1903.07291

(https://arxiv.org/abs/1903.07291)

My implementation of replenishment module is referred to this paper (https://arxiv.org/abs/1903.07291). SPADE module is suitable for conditional GAN. It has two input, one is the forwarding data, and another is the semantic-guide produce by the suppression module.

Loss function

General face restoration models just use MSE for loss function. But here, we focus on higher semantic fidelity and visual realism. Thus, we not calculate the loss just by the loss that the discriminator provide. We will introduce other two loss: (1) multi-scale feature matching loss

$\mathcal{L}_{FM}$ and (2) perceptual loss $\mathcal{L}_{prec}$

- $\mathcal{L}_{FM}$: It's referred to the loss function use in pix2pix (https://arxiv.org/abs/1711.11585).
- $\mathcal{L}_{prec}$: This part can use a pretrained VGG-19 network.
- The final loss function is the linear combination of the original loss and the two above.

Reflection

Although I still failed to implement this project, I learned a lot in the procedure.
This paper actually provides little information about its implementation, which make me have to refer many other paper to know the detail. At first, I went to see the github repository of this paper, but I found the author even not use the model he propose, and just use pix2pix model. (He didn't complete his model as well, his code of model and just copy from NVIDIA) So, I had to do it by myself. After trying for days, I still can't find a satisfying solution. The models I designed is either too large (billions of parameters) or can't use (dimensions unmatch). It's frustrated that the methods I spent so much time to make failed again and again, especially I'm still unfamiliar with this framework.

Maybe I'm not doing well in the end, I still gained something from it. I did the degradation functions well, learned more techniques of conditional GAN and attempted to read more papers. I believe it will help me a lot on my following researches.

Reference

- HiFaceGAN-Paper (https://arxiv.org/abs/2005.05005)
- HiFaceGAN-GitHub (https://github.com/Lotayou/Face-Renovation)
- pix2pix (https://arxiv.org/abs/1711.11585)
- SPADE (https://arxiv.org/abs/1903.07291)