

Deep Learning

Exercise 9: Facial Attribute Manipulation via StarGAN

Instructor: Manuel Günther

Email: guenther@ifi.uzh.ch

Office: AND 2.54

Friday, May 7, 2020

Outline

1

StarGAN

Outline

1

StarGAN

- StarGAN Paper
- StarGAN Repository
- Attribute Manipulation

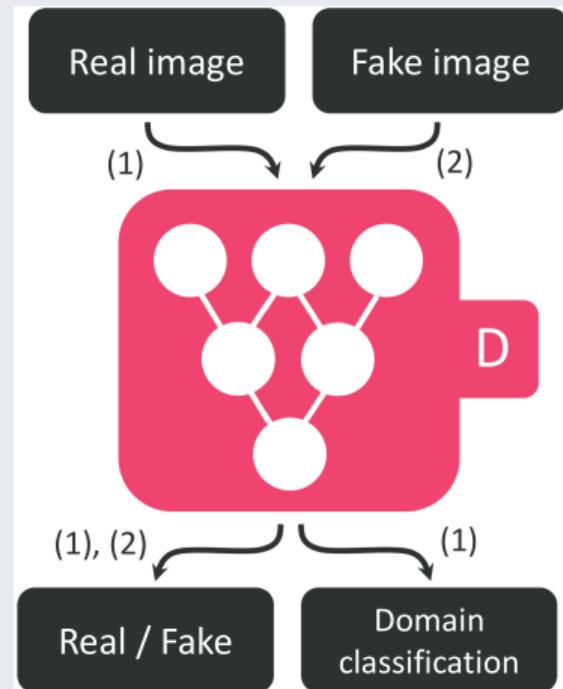
StarGAN Paper

Facial Attribute Manipulation

- Conditional GAN
 - Conditioned on facial attributes
 - ⇒ Attribute vector $\vec{t} \in \{+1, -1\}^{40}$
- Generator $\mathcal{G}(\vec{x}, \vec{t}')$ requires target \vec{t}'
- Two-fold discriminator
 - Classify real/fake
 - Classify original attribute vector \vec{t}
- Cycle consistency: $\|\vec{x} - \mathcal{G}(\mathcal{G}(\vec{x}, \vec{t}'), \vec{t})\|_1$

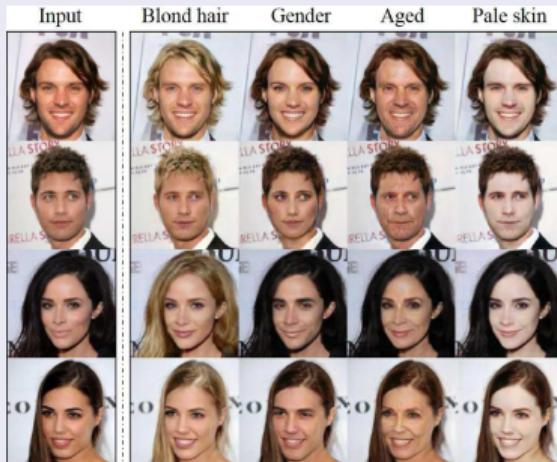
Choi, Choi, Kim, Ha, Kim, Choo: "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", CVPR 2018

Discriminator Training



StarGAN Paper

Manipulated Attributes



Attribute Manipulation

- Use Generator to modify attributes
- Can modify several attributes
- In total 40 different attributes
 - Here: only 5 attributes used
 - 'Black_Hair', 'Blond_Hair',
 - 'Brown_Hair', 'Male', 'Young'

Multiple Attributes



StarGAN Repository

StarGAN

- PyTorch implementation
- Pre-trained GANs for 5 attributes
- Code to train GAN

Download Repository

- via `git`:
`git clone https://github.com/yunjey/stargan`
- or: download ZIP and extract

Repository URL

<https://github.com/yunjey/stargan>

Preparation

- Download pre-trained GANs:

```
bash ./download.sh pretrained-celeba-256x256
```

```
bash ./download.sh pretrained-celeba-256x256
```

- Optional: Download CelebA:

```
bash ./download.sh celeba
```

- Look into `download.sh` to download manually

Attribute Manipulation

Task

- ① Replicate their experiments
 - Requires downloaded database
- ② Use images of your own
 - Crop your face manually
- ③ Load image from file
- ④ Apply image transforms
- ⑤ Create attribute vector $\in \{0, 1\}^5$
- ⑥ Load pre-trained generator
- ⑦ Call generator (image + target)
- ⑧ Save result as image

main.py

- Run their experiments
 - See online documentation
 - `--use_tensorboard false`

dataloader.py

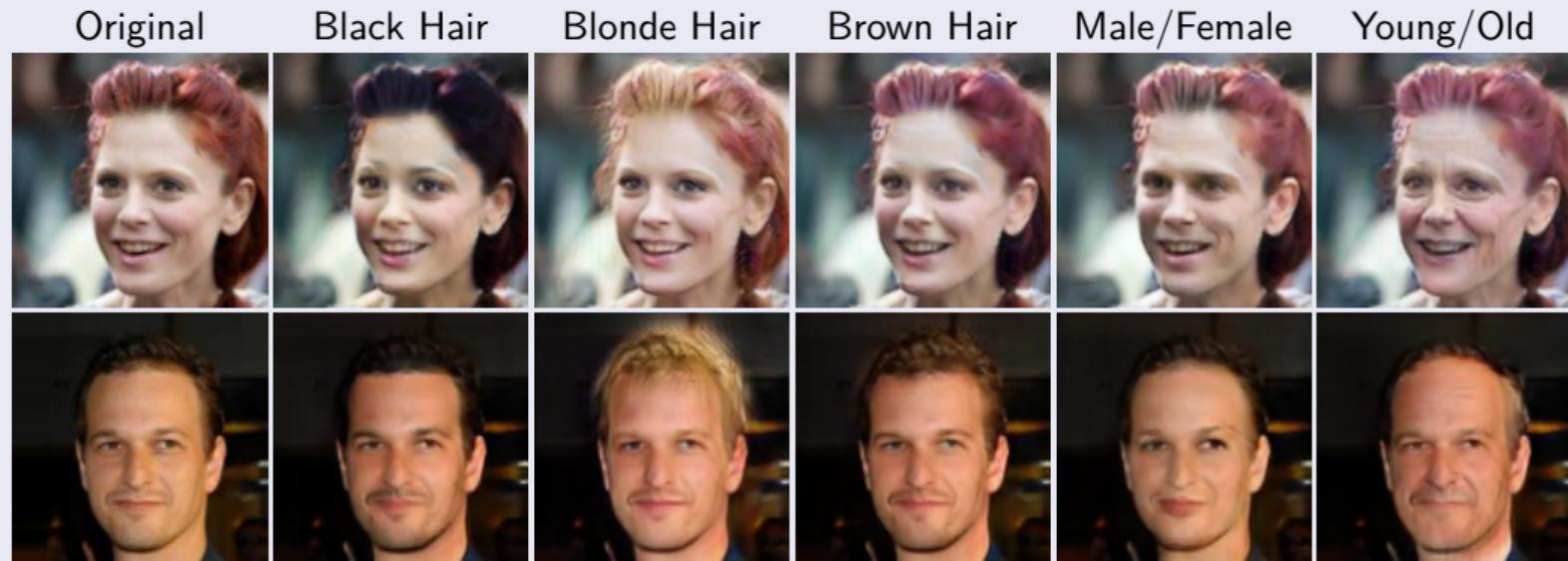
- Check function `get_loader`
 - Use transform from there

model.py

- Instantiate and use `Generator`
 - Default parameters viable
- Load pre-trained model
 - See `solver.py`, line 101

Attribute Manipulation

Results from the CelebA Dataset



Attribute Manipulation

Results from my own Images

