

jiechen2358 / FaceAging-by-cycleGAN Public

MIT License

117 stars 42 forks

Star

Watch

Code

Issues 3

Pull requests

Actions

Projects

Wiki

Security

Insights

master

...



jiechen2358 ...

on Sep 18, 2019

[View code](#)

FaceAging-by-cycleGAN

This project was graded 101/100 by cs230(fall semester 2018) of Stanford University.

Contribution

This repo is heavily based on [Original CycleGAN implementation](#). Most of our work involves adding code to better handle the dataset we are working with, and adding a couple of small features that enables transfer learning.

Data Process Related

The code related to data processing is located in folder [data_processing](#), the folder contains utilities perform following tasks

- `get_test_images.py`: Small utility to gather specific test cases images from result folders of all models.
- `imdb_mat_to_pickle.py`: Read metadata of IMDB data of IMDB-WIKI dataset (.mat format) and convert it to .pkl format.
- `wiki_mat_to_pickle.py`: Read metadata of WIKI data of IMDB-WIKI dataset (.mat format) and convert it to .pkl format.

- process_cacd.py: Read CACD dataset and retrieve two groups of images, one group is old people and another group is young people.
- process_wiki.py: Read IMDB-WIKI dataset and retrieve two groups of images, one group is old people and another group is young people.
- log_dataset.py: Create a log directory with 4 log files (trainA.log, trainB.log, testA.log and testB.log) that captures the filename of all the files within trainA, trainB, testA, and testB directories. This is especially useful when we are doing data processing and want to keep track of what files got moved at what time.
- remove_grayscale_and_small_images.py: process a directory of images and remove all grayscale images as well as small images. These images are not suitable for training.
- validate_image.py: Validate all images in a directory are of .jpg format instead of other format. This is done through reading the metadata of the file, instead of relying on filename extensions.
- process_loss_log.py: Reads a loss_log.txt file generated by CycleGAN training process and plot the loss curve.

Beyond the data_processing directory, we also added a print_structure.py script under project directory to print the network architecture based on the user selections.

Also, we added scripts/unzip_dataset_all.sh, scripts/unzip_dataset_male.sh, scripts/unzip_dataset_female.sh scripts to help processing the unpacking of our dataset.

Support Transfer Learning

Modified original model implementation to added Transfer Learning functionality (**models**). 4 files(models/base_model.py, models/networks.py, models/cycle_gan_model.py and models/test_model.py) were modified in order to support this.

We also modified options/base_option.py and options/train_options.py. These files are located in folder: **options**. We added following flags for transfer learning:

- --use_pretrained_model: whether we should use pretrained model
- --pretrained_model_name: name of the pretrained model we want to use for transfer learning, e.g. orange2apple
- --pretrained_model_subname: name of the sub models (comma delimited) we want to use for transfer learning, e.g. GA,GB
- --pretrained_model_epoch: epoch of the pretrained model we want to use for transfer learning
- --G_A_freeze_layer: freeze these many initial layers (inclusive) of G_A network during training

- --G_B_freeze_layer: freeze these many initial layers (inclusive) of G_B network during training
- --D_A_freeze_layer: freeze these many initial layers (inclusive) of D_A network during training
- --D_B_freeze_layer: freeze these many initial layers (inclusive) of D_B network during training

Pre-trained Model

The trained models are located in folder **trained_model**, including:

- 0_cacd_regular - checkpoints trained on CACD all gender datasets
- 1_wiki_base_mix - checkpoints trained on IMDB-WIKI all gender datasets
- 2_wiki_base_female - checkpoints trained on IMDB-WIKI female-only datasets
- 3_wiki_base_male - checkpoints trained on IMDB-WIKI male-only datasets
- 4_wiki_smaller_model - checkpoints trained on IMDB-WIKI datasets, with 6 ResNet blocks for generator network
- 5_wiki_transfer_horse2zebra - checkpoints trained on IMDB-WIKI datasets, transfer learning with horse2zebra and locking * ResNet blocks #1 to #7
- 6_wiki_transfer_summer2winter - checkpoints trained on IMDB-WIKI datasets, transfer learning with summer2winter and locking ResNet blocks #1 to #7
- 7_wiki_transfer_monet2photo - checkpoints trained on IMDB-WIKI datasets, transfer learning with monet2photo and locking ResNet blocks #1 to #7
- 8_wiki_fine_tune_horse2zebra - checkpoints trained on IMDB-WIKI datasets, fine tuning with horse2zebra further trained 100 epochs
- 9_wiki_fine_tune_male - checkpoints trained on IMDB-WIKI datasets, fine tuning with 3_wiki_base_male further trained 100 epochs

Supplementary Classifier

Implemented a **Gender & Age Classifier** intent to perform multiple tasks.

- From application perspective, we plan to develope a model selector in the future since our experiments show that the results of our model are influenced by the gender of the input images. For example, model trained on female has better performance on female image inputs.

Quantitative Assessment

Age Estimation Tool Created some small utility code that can quantitatively estimate CycleGAN's performance by estimating the age of all the people in the test set, before and after the age progression. This allow us to quantitatively see how much our model has made people become.

Introduction

In this project, we proposed a simple, yet intuitive deep learning model based on **CycleGAN** that can generate predictive images of people's look after certain years based on their current images, without the need of paired dataset.

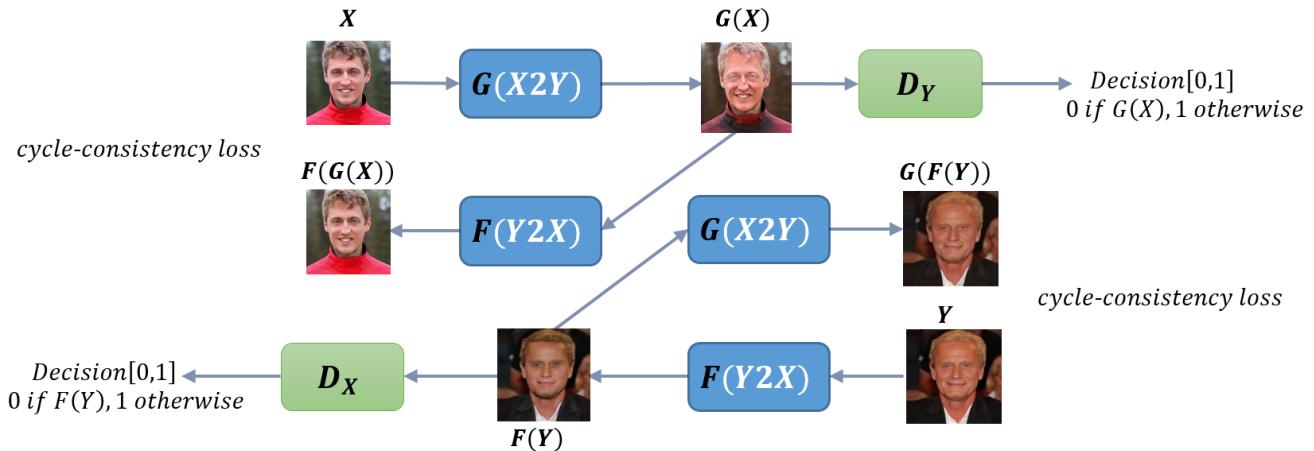
Dataset

1. **IMDB-WIKI-500k+ face images with age and gender labels** - contains 500k+ images of celebrities from IMDb and Wikipedia. The metadata of this dataset contains the date of birth of the person portrayed in the image and the date of which the image was taken.
2. **Cross-Age Celebrity Dataset (CACD)** - contains 163k+ images of 2,000 celebrities. "The images are collected from search engines using celebrity name and year (2004-2013) as keywords. We can therefore estimate the ages of the celebrities on the images by simply subtract the birth year from the year of which the photo was taken."

Modeling and Implementation

We picked the **CycleGAN** model because we think it can solve two major problems we face:

1. Making sure the original image and the translated image represent the same person.
Adversarial losses alone cannot guarantee that the learned function can map an individual input to a desired output. CycleGan regularizes the space of possible mapping by adding the cycle-consistency losses. Combining this loss with the adversarial losses on the domains "young" and "old", it made sure that the transformed images still represent the same person in the original images.
2. Obtain photos of one person that were taken at different ages is challenging and expensive. Therefore, we need an algorithm that can pass around domain knowledges between groups without paired input-output examples. CycleGAN can capture characteristics of faces of young people portraited in one image collection and figure out how these characteristics can be applied to those facial images in a collection of old people.



Getting Started

Installation

1. Clone this repo

```
git clone https://github.com/jiechen2358/FaceAging-by-cycleGAN.git
cd FaceAging-by-cycleGAN
```

2. Install PyTorch 0.4+ and torchvision from <http://pytorch.org> and other dependencies (e.g., visdom and dominate). You can install all the dependencies by

```
pip install -r requirements.txt
```

3. For Conda users, we include a script ./scripts/conda_deps.sh to install PyTorch and other libraries.

Train/Test

1. Unpack datasets:

```
./scripts/unzip_datasets.sh
```

2. Train a model:

```
python train.py --dataroot ./datasets/young2old --name aging_cyclegan --model
cycle_gan
```

To view training results and loss plots, run `python -m visdom.server` and click the URL <http://localhost:8097>. To see more intermediate results, check out [./checkpoints/aging_cyclegan/web/index.html](#).

3. Test the model:

```
python test.py --dataroot ./datasets/young2old --name aging_cyclegan --model
cycle_gan
```

The test results will be saved to a html file here:
[./results/aging_cyclegan/latest_test/index.html](#).

Results

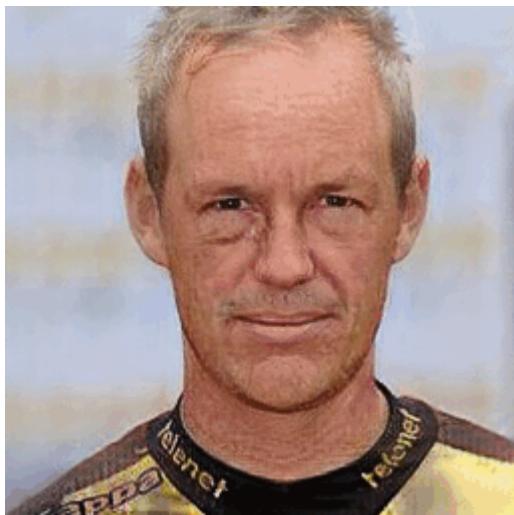
The following table lists all models we have explored in our study and their quantitative result. The columns are (from left to right): model number, data source, data composition, number of epochs trained, pre-trained network to initialize with, freeze until what layer of generator net, the size of the generator net, maximum age progression (years), average age progression, last 3 columns are the % of test cases where age progression is over 10 years, 15 years and 20 years. Results in model #4 is N/A due to model collapsing.

#	Source	Mix	Epochs	Preloaded?	Freeze until	G Size	Max	Avg	10+	15+	20+
0	CACD	All	200	N/A	N/A	9 blocks	25.8	6.7	22%	5.5%	1.7%
1	WIKI	All	200	N/A	N/A	9 blocks	31.2	8.8	37%	14%	5.8%
2	WIKI	Female	200	N/A	N/A	9 blocks	19.5	4.6	7.1%	2.5%	0.0%
3	WIKI	Male	200	N/A	N/A	9 blocks	27.3	10.3	50%	19%	5.1%
4	WIKI	Male	200	N/A	N/A	6 blocks	N/A	N/A	N/A	N/A	N/A
5	WIKI	All	200	horse2zebra	8th block	9 blocks	27.4	11.0	55%	20%	6.3%
6	WIKI	All	200	summer2winter	8th block	9 blocks	25.0	8.9	36%	10%	1.7%
7	WIKI	All	200	monet2photo	8th block	9 blocks	20.1	6.6	15%	2.5%	0.4%
8	WIKI	Male	100	horse2zebra	N/A	9 blocks	25.8	9.9	46%	12%	1.3%
9	WIKI	Male	100	Model #2	N/A	9 blocks	32.8	10.3	51%	18%	6.0%

Young to Old

Old to Young

Old to Young



☰ README.md



Input

Output

reconstruction





Report

[AgingGAN: Age Progression with CycleGAN](#)

References

- CycleGAN paper [link](#)
 - Official source code pyTorch implementation [link](#)
-

Releases

No releases published

Packages

No packages published

Contributors 3

-  JunwenBu [junwenbu](#)
 -  zhaoyu92
 -  jiechen2358 Jie
-

Languages

