

📄 junyanz / **pytorch-CycleGAN-and-pix2pix**

Image-to-image translation in PyTorch (e.g., horse2zebra, edges2cats, and more)

[#pytorch](#) [#gan](#) [#cyclegan](#) [#pix2pix](#) [#deep-learning](#) [#computer-vision](#) [#computer-graphics](#) [#image-manipulation](#) [#image-generation](#) [#generative-adversarial-network](#) [#gans](#)

📄 240 commits

🌿 3 branches

📦 0 releases

👤 26 contributors

Branch: master ▾


New pull request

Create new file

Upload files
















Find file

Clone or download ▾




SsnL Support visdom enviroment option

Latest commit bcfbb81 2 days ago

 data	fix the error doc in the data init helper function	a month ago
 datasets	Multiple changes regarding option management. See below.	2 months ago
 docs	Update qa.md	4 days ago
 imgs	update README	4 days ago
 models	Fixed a bug while loading old pix2pix models	6 days ago
 options	Support visdom enviroment option	2 days ago
 scripts	merge and reorganize the scripts	6 days ago
 util	Support visdom enviroment option	2 days ago
 .gitignore	Fix multiprocessing for Windows by using the __name__ == '__main__' i...	5 months ago
 LICENSE	first commit	a year ago
 README.md	Merge branch 'master' of https://github.com/junyanz/pytorch-CycleGAN-...	4 days ago
 environment.yml	Add environment.yml, for easier setup with Conda	5 months ago
 requirements.txt	update to 0.4	2 months ago
 test.py	move model.setup into train and test	2 months ago
 train.py	move model.setup into train and test	2 months ago

📄 README.md



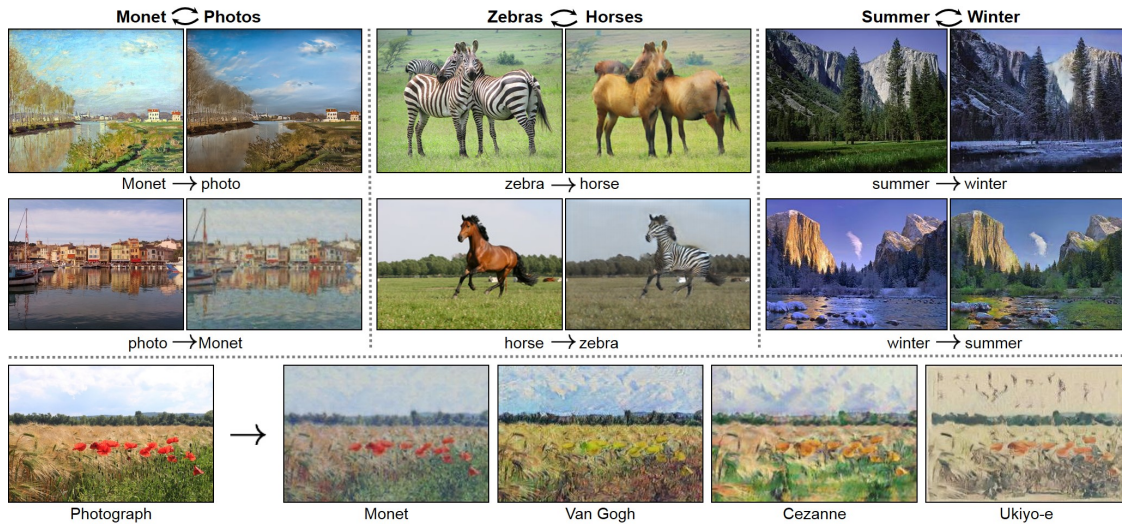
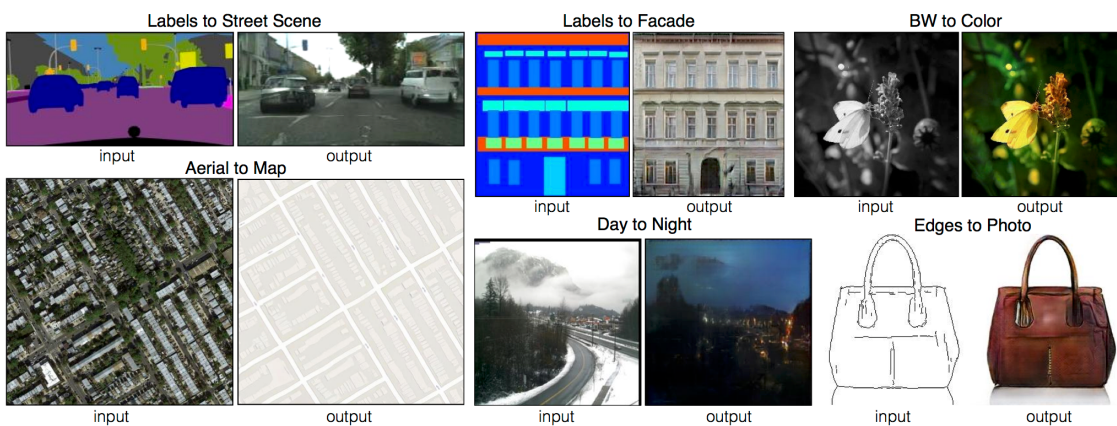
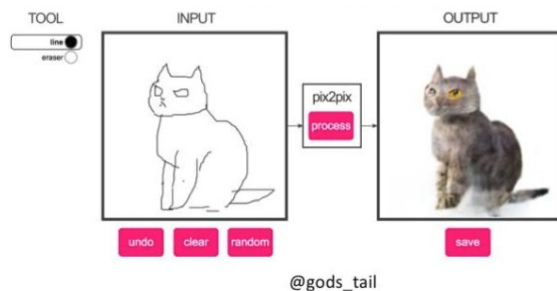
CycleGAN and pix2pix in PyTorch

This is our PyTorch implementation for both unpaired and paired image-to-image translation. It is still under active development.

The code was written by [Jun-Yan Zhu](#) and [Taesung Park](#), and supported by [Tongzhou Wang](#).

This PyTorch implementation produces results comparable to or better than our original Torch software. If you would like to reproduce the exact same results as in the papers, check out the original [CycleGAN Torch](#) and [pix2pix Torch](#) code

Note: The current software works well with PyTorch 0.4. Check out the older [branch](#) that supports PyTorch 0.1-0.3.

CycleGAN: [Project](#) | [Paper](#) | [Torch](#)Pix2pix: [Project](#) | [Paper](#) | [Torch](#)EdgesCats Demo | [pix2pix-tensorflow](#) | by Christopher Hesse

If you use this code for your research, please cite:

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Jun-Yan Zhu*](#), [Taesung Park*](#), [Phillip Isola](#), [Alexei A. Efros](#) In ICCV 2017. (* equal contributions) [[Bibtex](#)]

Image-to-Image Translation with Conditional Adversarial Networks [Phillip Isola](#), [Jun-Yan Zhu](#), [Tinghui Zhou](#), [Alexei A. Efros](#) In CVPR 2017. [[Bibtex](#)]

Course

CycleGAN course assignment [code](#) and [handout](#) designed by Prof. [Roger Grosse](#) for CSC321 "Intro to Neural Networks and Machine Learning" at University of Toronto. Please contact the instructor if you would like to adopt it in your course.

Other implementations

CycleGAN

[[Tensorflow](#)] (by Harry Yang), [[Tensorflow](#)] (by Archit Rathore), [[Tensorflow](#)] (by Van Huy), [[Tensorflow](#)] (by Xiaowei Hu), [[Tensorflow-simple](#)] (by Zhenliang He), [[TensorLayer](#)] (by luoxier), [[Chainer](#)] (by Yanghua Jin), [[Minimal PyTorch](#)] (by yunjey), [[Mxnet](#)] (by Ldpe2G), [[lasagne/keras](#)] (by tjwei)

pix2pix

[[Tensorflow](#)] (by Christopher Hesse), [[Tensorflow](#)] (by Eyyüb Sariu), [[Tensorflow \(face2face\)](#)] (by Dat Tran), [[Tensorflow \(film\)](#)] (by Arthur Juliani), [[Tensorflow \(zi2zi\)](#)] (by Yuchen Tian), [[Chainer](#)] (by mattya), [[tf/torch/keras/lasagne](#)] (by tjwei), [[Pytorch](#)] (by taey16)

Prerequisites

- Linux or macOS
- Python 2 or 3
- CPU or NVIDIA GPU + CUDA CuDNN

Getting Started

Installation

- Install PyTorch 0.4, torchvision, and other dependencies from <http://pytorch.org>
- Install python libraries [visdom](#) and [dominate](#).

```
pip install visdom dominate
```

- Alternatively, all dependencies can be installed by

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

- Clone this repo:

```
git clone https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix
cd pytorch-CycleGAN-and-pix2pix
```

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

CycleGAN train/test

- Download a CycleGAN dataset (e.g. maps):

```
bash ./datasets/download_cyclegan_dataset.sh maps
```

- Train a model:

```
#!/scripts/train_cyclegan.sh
python train.py --dataroot ./datasets/maps --name maps_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/maps_cyclegan/web/index.html`
- Test the model:

```
#!/scripts/test_cyclegan.sh
python test.py --dataroot ./datasets/maps --name maps_cyclegan --model cycle_gan
```

The test results will be saved to a html file here: `./results/maps_cyclegan/latest_test/index.html` .

pix2pix train/test

- Download a pix2pix dataset (e.g.facades):

```
bash ./datasets/download_pix2pix_dataset.sh facades
```

- Train a model:

```
#!/scripts/train_pix2pix.sh
python train.py --dataroot ./datasets/facades --name facades_pix2pix --model pix2pix --which_direction BtoA
```

- 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/facades_pix2pix/web/index.html`
- Test the model (`bash ./scripts/test_pix2pix.sh`):

```
#!/scripts/test_pix2pix.sh
python test.py --dataroot ./datasets/facades --name facades_pix2pix --model pix2pix --which_direction BtoA
```

The test results will be saved to a html file here: `./results/facades_pix2pix/test_latest/index.html` .

More example scripts can be found at `scripts` directory.

Apply a pre-trained model (CycleGAN)

- You can download a pretrained model (e.g. horse2zebra) with the following script:

```
bash ./scripts/download_cyclegan_model.sh horse2zebra
```

The pretrained model is saved at `./checkpoints/{name}_pretrained/latest_net_G.pth` . The available models are apple2orange, orange2apple, summer2winter_yosemite, winter2summer_yosemite, horse2zebra, zebra2horse, monet2photo, style_monet, style_cezanne, style_ukiyoe, style_vangogh, sat2map, map2sat, cityscapes_photo2label, cityscapes_label2photo, facades_photo2label, facades_label2photo, and iphone2dslr_flower.

- To test the model, you also need to download the horse2zebra dataset:

```
bash ./datasets/download_cyclegan_dataset.sh horse2zebra
```

- Then generate the results using

```
python test.py --dataroot datasets/horse2zebra/testA --name horse2zebra_pretrained --model test
```

The option `--model test` is used for generating results of CycleGAN only for one side. `python test.py --model cycle_gan` will require loading and generating results in both directions, which is sometimes unnecessary. The results will be saved at `./results/` . Use `--results_dir {directory_path_to_save_result}` to specify the results directory.

- If you would like to apply a pre-trained model to a collection of input images (rather than image pairs), please use `--dataset_mode single` and `--model test` options. Here is a script to apply a model to Facade label maps (stored in the directory `facades/testB`).

```
#!/scripts/test_single.sh
python test.py --dataroot ./datasets/facades/testB/ --name {your_trained_model_name} --model test
```

You might want to specify `--which_model_netG` to match the generator architecture of the trained model.

Apply a pre-trained model (pix2pix)

Download a pre-trained model with `./scripts/download_pix2pix_model.sh` .

- For example, if you would like to download label2photo model on the Facades dataset,

```
bash ./scripts/download_pix2pix_model.sh facades_label2photo
```

- Download the pix2pix facades datasets

```
bash ./datasets/download_pix2pix_dataset.sh facades
```

- Then generate the results using

```
python test.py --dataroot ./datasets/facades/ --which_direction BtoA --model pix2pix --name facades_label2ph
```

Note that we specified `--which_direction BtoA` as Facades dataset's A to B direction is photos to labels.

- See a list of currently available models at `./scripts/download_pix2pix_model.sh`

Datasets

Download pix2pix/CycleGAN datasets and create your own datasets.

Training/Test Tips

Best practice for training and testing your models.

Citation

If you use this code for your research, please cite our papers.

```
@inproceedings{CycleGAN2017,
  title={Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss},
  author={Zhu, Jun-Yan and Park, Taesung and Isola, Phillip and Efros, Alexei A},
  booktitle={Computer Vision (ICCV), 2017 IEEE International Conference on},
  year={2017}
}
```

```
@inproceedings{isola2017image,
  title={Image-to-Image Translation with Conditional Adversarial Networks},
  author={Isola, Phillip and Zhu, Jun-Yan and Zhou, Tinghui and Efros, Alexei A},
  booktitle={Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on},
  year={2017}
}
```

Related Projects

[CycleGAN-Torch](#) | [pix2pix-Torch](#) | [pix2pixHD](#) | [iGAN](#) | [BicycleGAN](#)

Cat Paper Collection

If you love cats, and love reading cool graphics, vision, and learning papers, please check out the Cat Paper Collection: [Github](#) | [Webpage](#)

Acknowledgments

Code is inspired by [pytorch-DCGAN](#).