

CycleGAN을 활용한 Cezanne의 화풍 입히기

<https://github.com/doobee98>

1. 요약

본 과제는 인터넷 상에 공개되어 있는 CycleGAN Source를 참고하여 직접 딥러닝 학습을 해 보고 Test 결과물이 의미 있게 변환되었는지 확인하는 것을 목적으로 한다. Cezanne의 그림들과 일반 사진들을 입력으로 받아, Cezanne의 그림을 사진으로 바꾸고 사진을 Cezanne의 그림으로 바꾸는 GAN을 학습한다. 그 후 일반 사진들을 Cezanne의 화풍으로 바꾸는 것을 테스트한다. 학습 과정에서는 몇몇 HyperParameter들을 조정해본다.

2. 데이터 집합 설명

```
!bash ./datasets/download_cyclegan_dataset.sh cezanne2photo
```

참조한 CycleGAN GitHub에서 제공하는 공개 데이터 Training Set과 Test Set을 사용하며, Test Set에는 작성자의 사진 몇 가지를 추가한다. A는 Cezanne의 그림들이고, B는 실제 사진들이다. CycleGAN은 GAN_A->B와 GAN_B->A를 모두 학습하며, 공개 데이터에 없는 Cezanne의 그림을 추가로 얻는 것은 다소 어렵기 때문에 Test Set은 GAN_B->A를 사용하여 실제 사진을 Cezanne의 그림으로 바꾸는 것을 확인하였다. HyperParameter 조정 시에는 Validation Set이 따로 필요하므로 Training Set을 적당히 쪼개서 사용하였다.

- Training Set (./datasets/cezanne2photo_train)
 - Source (trainA): Cezanne의 그림들 300장.
 - Target (trainB): 일반 사진들 4620장. 1 Epoch당 500/1000장 subset 학습.
- Validation Set (./datasets/cezanne2photo_valid)
 - Source (trainA): Cezanne의 그림들 225장

- Target (trainB): 일반 사진들 1667장. 1 Epoch당 500/1000장 subset 학습.
- Test Set (./datasets/cezanne2photo)
 - Source (testB): 일반 사진들 751장.
 - Source (testMy): 작성자가 추가한 일반 사진 23장.
 - Target (./results): 학습된 GAN에 의해 Cezanne의 화풍으로 변환된 사진들.

3. 실험 목적

일반 사진들(B)에 Cezanne의 화풍(A)을 입히는 것을 목적으로 한다. 전체적인 사물, 배경 등의 형태는 최대한 그대로 유지하고, 스케치 기법이나 표현, 색감 등은 사진이 아닌 그림처럼 보이도록 변환한다. A와 B를 거꾸로 매치할 수도 있지만(일반사진-A, Cezanne-B), CycleGAN은 학습 과정에서 두 가지 방향의 GAN을 모두 학습하며, 일반 사진의 개수가 Cezanne의 그림 개수보다 훨씬 많았기에 개수가 더 적은 Cezanne의 그림들을 A로 설정하였다. 우선 HyperParameter를 조정하면서 적당한 Model을 찾고, 그 Model로 학습 후 Test Set을 넣어 최종 결과를 확인한다. 조정할 HyperParameter는 lr(learning rate), beta1(momentum), batch_size 세 가지이다.

4. 프로그램 간단 설명

[출처]

CycleGAN Github: <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>
 프로그램 명령어: <https://colab.research.google.com/github/junyanz/pytorch-CycleGAN-and-pix2pix/blob/master/CycleGAN.ipynb>

수업 시간에 사용되었던 DLL 예시들 중, 화풍을 입히는 것이 매력적으로 느껴져서 GitHub에서 해당 CycleGAN 프로그램을 찾아 추가적인 코드 수정 없이 사용하였다.

data	improve _scale_width function
datasets	added documentation about the new Cityscapes dataset download policy
docs	update qa
imgs	update README
models	fix typo
options	change niter to n_epochs
scripts	Update batch_hed.py
util	replace scipy.misc.imresize (suggested by @grochfort)
.gitignore	Fix multiprocessing for Windows by using the __name__ == '__main__' i...
CycleGAN.ipynb	Add CycleGAN colab notebook
LICENSE	first commit
README.md	fix a typo in README about the PyTorch url
environment.yml	Update environment.yml
pix2pix.ipynb	add colab notebook
requirements.txt	Update requirements.txt
test.py	add opt.load_iter to the test code
train.py	change niter to n_epochs

options 폴더에 있는 base, train, test_options.py에서 옵션을 참조할 수 있으며, 실제 학습과 테스트는 각각 train.py, test.py를 사용하여 진행한다. 학습의 결과는 checkpoints라는 새로운 폴더에 담기고, 테스트의 결과는 results라는 새로운 폴더에 담긴다.

5. 실험 과정

- HyperParameter 조정

```
!python3 train.py --dataroot ./datasets/cezanne2photo_train --name cezanne2photo_<test_name> --
model cycle_gan --max_dataset_size 500 --save_latest_freq 500 --n_epochs 9 --n_epochs_decay 1 --lr
<lr_value> --beta1 <beta1_value> --batch_size <batch_size_value>
```

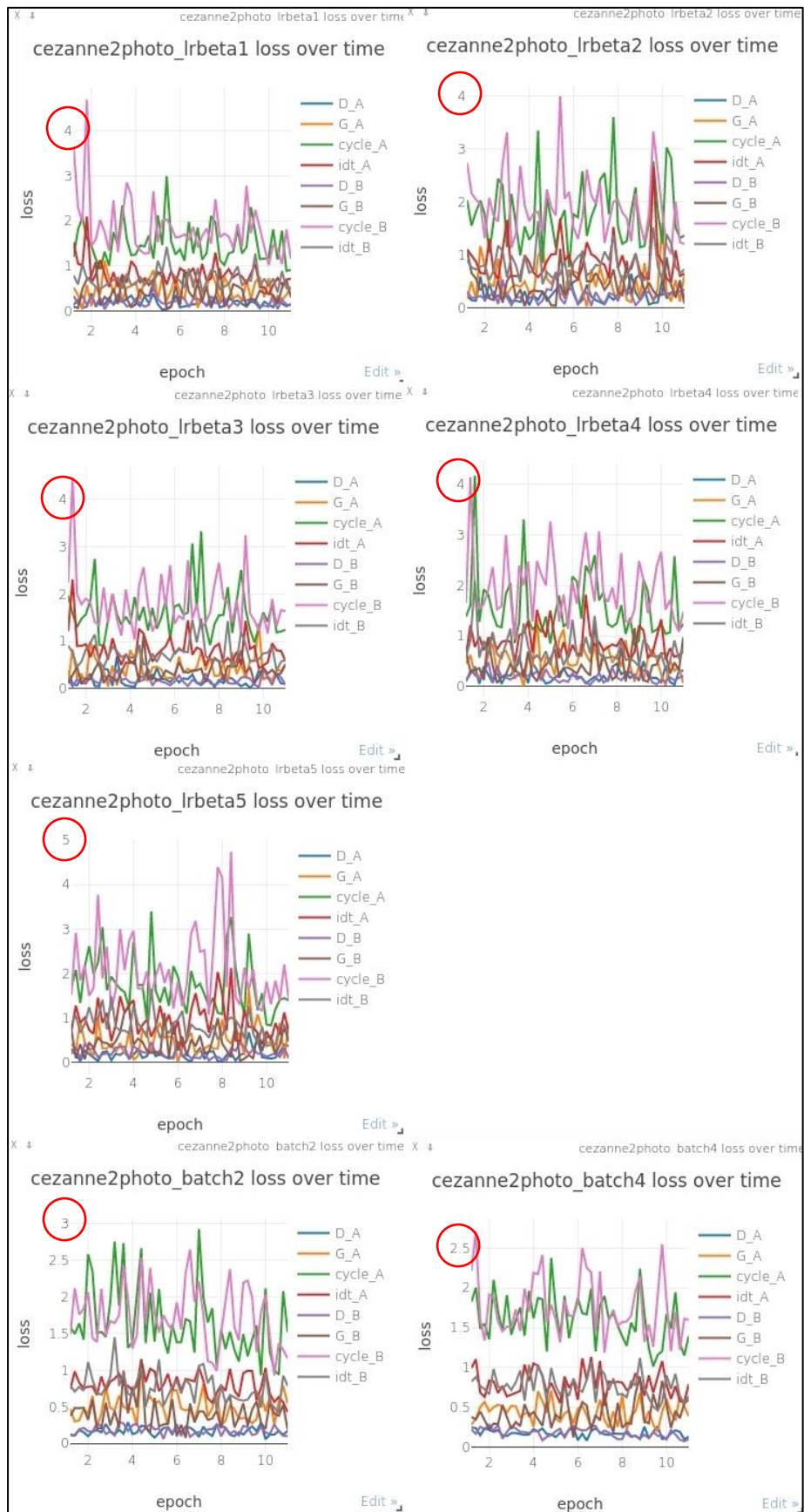
```
!python3 train.py --dataroot ./datasets/cezanne2photo_valid --name cezanne2photo_<test_name> --
model cycle_gan --max_dataset_size 500 --save_latest_freq 500 --lr <lr_value>--beta1 <beta1_value>
--batch_size <batch_size_value> --n_epochs 9 --n_epochs_decay 1 --phase 'valid' --continue_train
```

이번 과제에서는 Learning Rate(lr), Momentum(beta1), Batch Size(batch_size) 총 세 가지의 HyperParameter를 Manually Optimization 하였다. 학습을 하는데 드는 시간이 다소 오래 걸렸기 때문에 500개의 data를 가진 10개의 Epoch만 가지고 실험했다.

	lr	beta1	batch_size	비고
<u>lrbeta1</u>	<u>0.0002</u>	<u>0.8</u>	<u>1</u>	<u>선택</u>
lrbeta2	0.0004	0.5	1	
lrbeta3	0.0001	0.9	1	
lrbeta4	0.0002	0.5	1	
lrbeta5	0.0004	0.8	1	
batch2	0.0002	0.8	2	
<u>batch4</u>	<u>0.0002</u>	<u>0.8</u>	<u>4</u>	<u>선택</u>

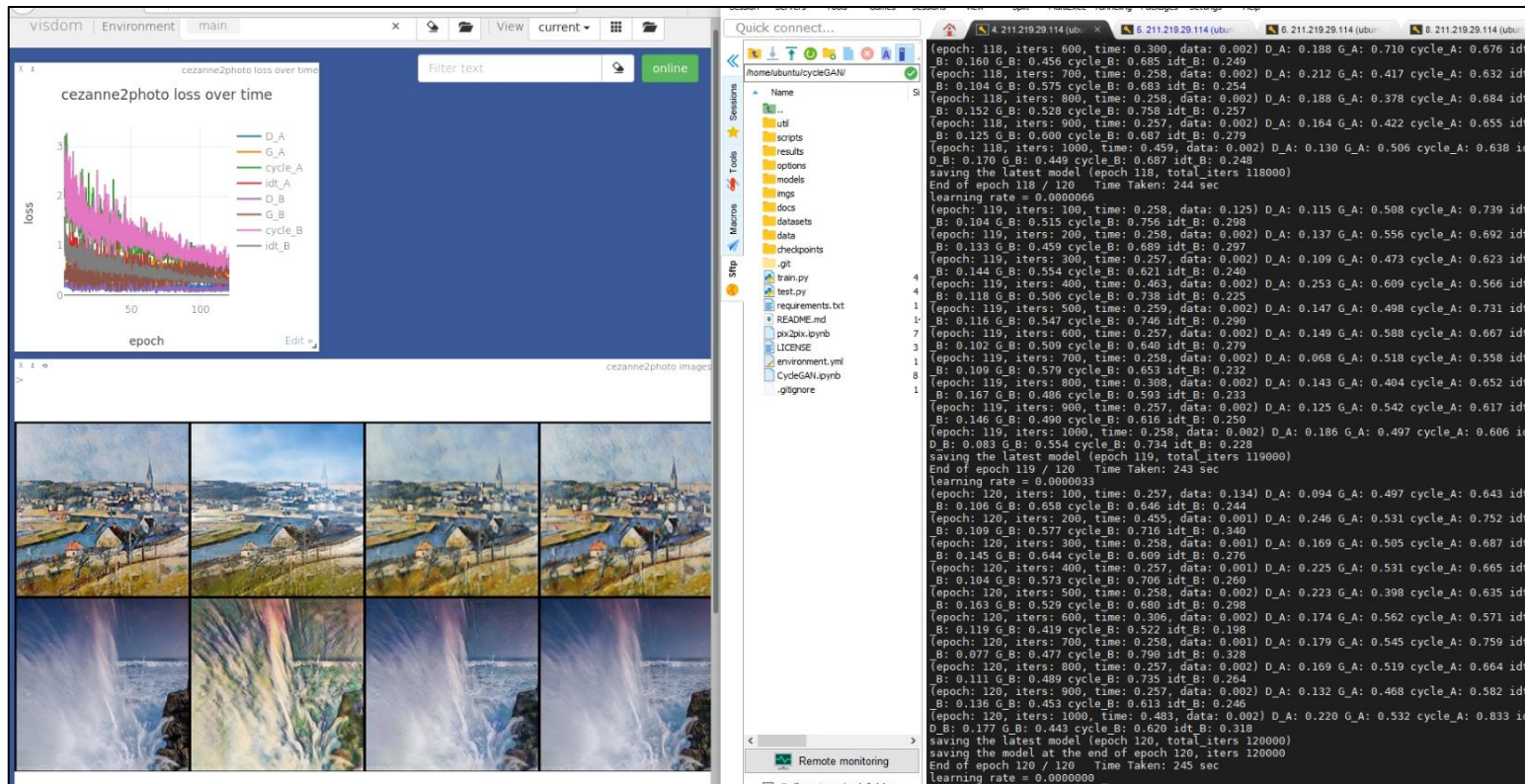
[표 1: 테스트한 모델 상세 내역]

먼저 Learning Rate와 Momentum을 동시에 조정하는 실험을 다섯 번 진행하여 두 파라미터를 확정 후, Batch Size를 최종적으로 조정하였다. 상세한 Validation Loss Graph는 다음 쪽의 [그림 1]에서 확인할 수 있다. 평균적인 Loss가 가장 적었던 모델을 선택하였다. Learning Rate와 Momentum을 조정한 첫 다섯 번의 실험에서는 각각 0.0002, 0.8 값을 선택하였고, 해당 값을 가지고 Batch Size를 조정하였다. Batch Size를 8 이상 줄 경우 Memory Out of Bound 에러가 발생하여 더 이상 진행할 수 없었기에 4까지만 실험하였으며, 그 중 4 값을 선택하였다.



[그림 1: Validation loss graph. 좌측의 x축 선이 제각각 임에 유의한다.]

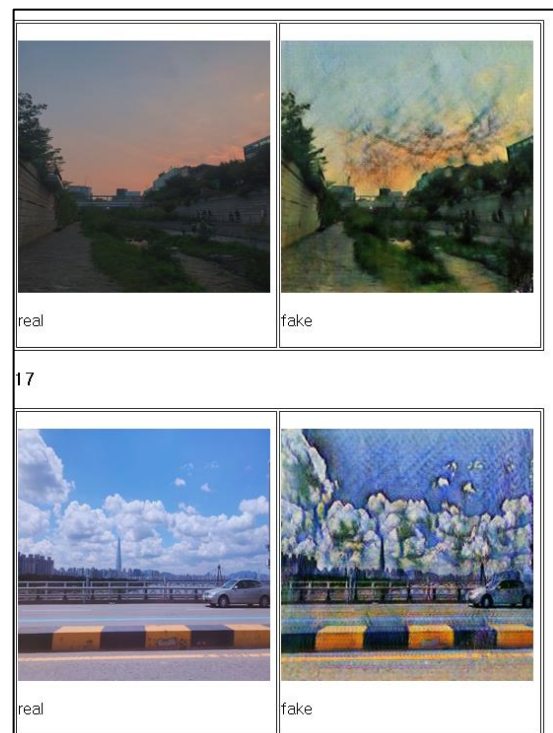
- 학습



앞서 정한 <Learning Rate = 0.0002, Momentum = 0.8, Batch Size = 4> 모델로 학습을 진행한다. 충분한 양의 학습을 진행하기 위해 1000개의 데이터를 가진 120개의 Epoch(60개: initial learning rate, 60개: 시간에 따른 감소된 learning rate)로 8시간동안 진행했으며, 최종 학습은 이전의 Training Set과 Valid Set을 합친 기존의 공개 Training Set을 사용하여 진행하였다.

- 테스트

공개 데이터에 있던 testB와 작성자가 추가한 testMy까지 총 774개의 사진을 변환하는 테스트를 진행하였다. 예시 결과는 오른쪽 그림과 같다.



17