

Deep Learning - Lab4

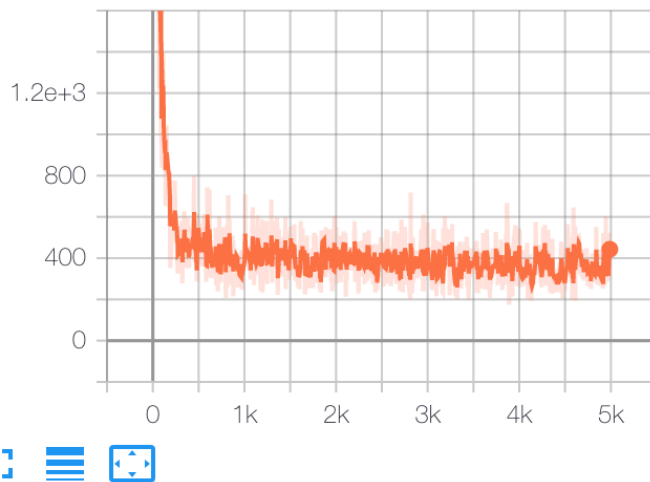
컴퓨터전공

2013011695 정태화

1. Tensorboard result

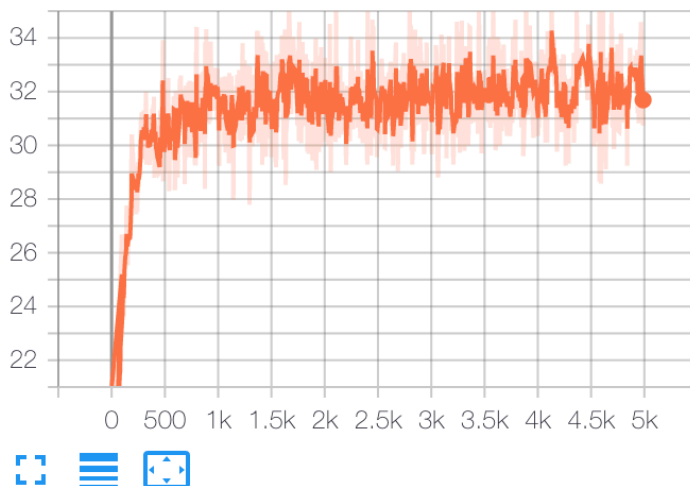
<Loss(cost)>

Cost



<PSNR>

PSNR_1

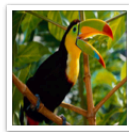


2. PSNR for Set5 validation set

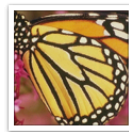
(512, 512, 1)
PSNR: 35.413906
(256, 256, 1)
PSNR: 28.364067
(344, 228, 1)
PSNR: 32.085464
(288, 288, 1)
PSNR: 36.70462
(280, 280, 1)
PSNR: 32.93249



001_HR.png



002_HR.png



003_HR.png



004_HR.png



005_HR.png



0.jpg



1.jpg



2.jpg



3.jpg



4.jpg

3. Report

전부 291장의 사진 중에, training set은 256장 그리고 test set을 35장으로 나누어 진행하였습니다.

각 사진을 전부 벡터화 시켜 grayscale로 부른 후, resize를 이용하여 사진의 해상도를 떨어트린 후에 32*32 size로 crop하여 training set을 구성하였습니다.

또한 매 epoch마다 training set으로 쓰일 사진을 셔플하여 다른 사진들도 training 될 수 있게 하였고, 또한 crop도 랜덤하게 진행하여 batch_size를 128로 잡고 학습시켰습니다.

처음에는 모든 사진을 다 32*32로 전체 분할시켜 트레이닝을 진행하였으나 3만장 이상의 사진이 생겨났고 그것들을 학습시키려니 300시간의 예상시간이 잡혀 이렇게 수정하여 진행하였습니다. 또한 매 epoch마다 사진을 선택한 후 크롭 진행 후, 저화질로 변경시키고 흑백으로 만드는 것의 처리가 너무나 시간을 많이 잡아먹어 이미지 전처리를 하는 방식에 조금 더 주력하였습니다. 현재는 epoch 마다 사진 크롭만 하면 되는 방식으로 모델을 구현하였습니다

이번에는 rnn의 형태로 recurrent하게 모델을 돌려야 했고, 그래서 중간 h값의 placeholder가 위에서 내려오는 인자, h인자, 그리고 아래 필터로 내릴 인자까지 3개의 filter를 구성하여 모델을 만들었습니다. 저번의 1만번이 너무나 오래 걸려서 이번에는 5000번의 epoch로 과제를 돌렸고, 기타 사진의 인풋과 아웃풋, cost, accuracy는 전부 비슷하게 출력하였습니다.

```
epoch: 606    cost: 399.3147    PSNR: 31.423576
epoch: 607    cost: 340.5083    PSNR: 29.975105
epoch: 608    cost: 415.1473    PSNR: 30.661171
epoch: 609    cost: 380.06168   PSNR: 32.147346
epoch: 610    cost: 477.41156    PSNR: 31.838568
epoch: 611    cost: 418.3415    PSNR: 30.670902
epoch: 612    cost: 543.1686    PSNR: 30.854126
epoch: 613    cost: 440.07437   PSNR: 28.918179
epoch: 614    cost: 425.70322   PSNR: 33.291534
epoch: 615    cost: 400.45398   PSNR: 34.19809
epoch: 616    cost: 423.452     PSNR: 32.58716
```

Cnn보다는 좀더 멀게, 600번 정도의 시도에서 코스트가 안정화되었고 psnr의 값도 논문의 값과 비슷하게 나와 안심했고, 시간도 얼추 괜찮게 나와서 그 후에 5000번의 시도를 하였습니다.

```
epoch: 4991    cost: 358.85403    PSNR: 31.390873
epoch: 4992    cost: 303.592     PSNR: 31.478506
epoch: 4993    cost: 313.69623    PSNR: 30.823944
epoch: 4994    cost: 328.5702    PSNR: 30.776653
epoch: 4995    cost: 471.94556    PSNR: 35.323006
epoch: 4996    cost: 330.08734    PSNR: 33.431988
epoch: 4997    cost: 487.69626    PSNR: 32.453438
epoch: 4998    cost: 376.0408    PSNR: 31.358036
epoch: 4999    cost: 307.56107    PSNR: 31.336245
```

그 결과 400분을 cpu로만 돌린 끝에 결과를 얻었고 tensorboard에도 만족스러운 그래프의 형태가 찍혔습니다. 또한 tf.saver() 를 이용한 parameter와 딥러닝 모델 저장도 500 epoch마다 하였습니다.

마지막으로 Set5의 5장의 사진에 대한 validation이 필요했기 때문에 이 5장을 불러와 psnr을 계산한 결과 아래와 같은 결과가 나와 만족스러운 값을 받았습니다.

```
(512, 512, 1)
PSNR: 35.413906
(256, 256, 1)
PSNR: 28.364067
(344, 228, 1)
PSNR: 32.085464
(288, 288, 1)
PSNR: 36.70462
(280, 280, 1)
PSNR: 32.93249
```