

인공지능 프로젝트 최종 보고서

Super resolution with sub-pixel layer

B611095 서정욱/ B711050 김혁진/ B711121 윤석찬/ B811016 김도연

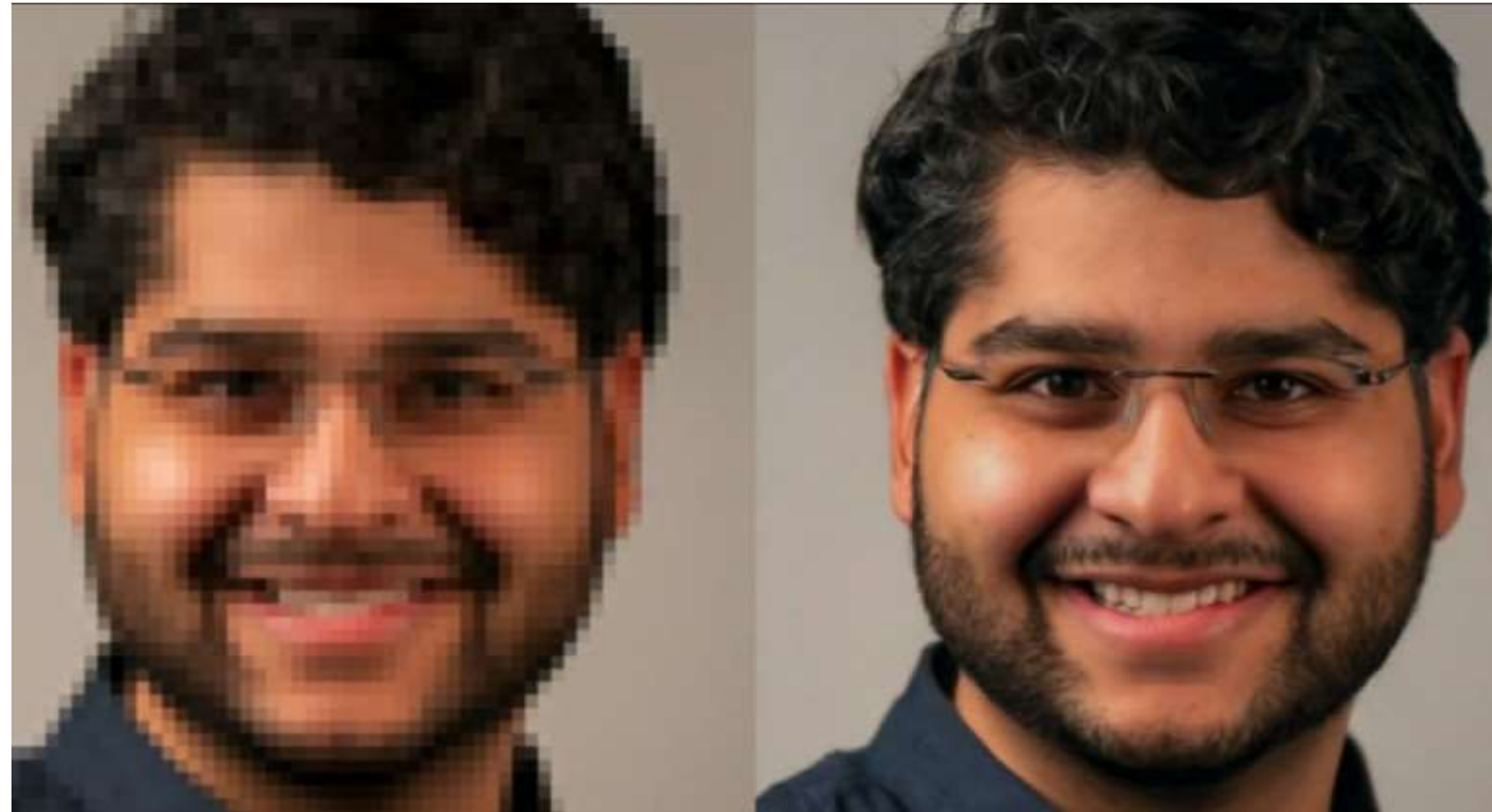
목차

- 프로젝트 개요
- 데이터 설명
- 사용한 머신러닝 모델
- 개발에 사용한 툴
- 학습 과정
- 결과 및 결과분석
- 차용한 부분
- 직접 구현한 부분
- 소스코드 및 데이터url

프로젝트 개요

Super resolution with sub-pixel layer

- 저화질의 이미지를 고화질로 바꿔주는 cnn 모델 구현



데이터 설명

데이터 구성

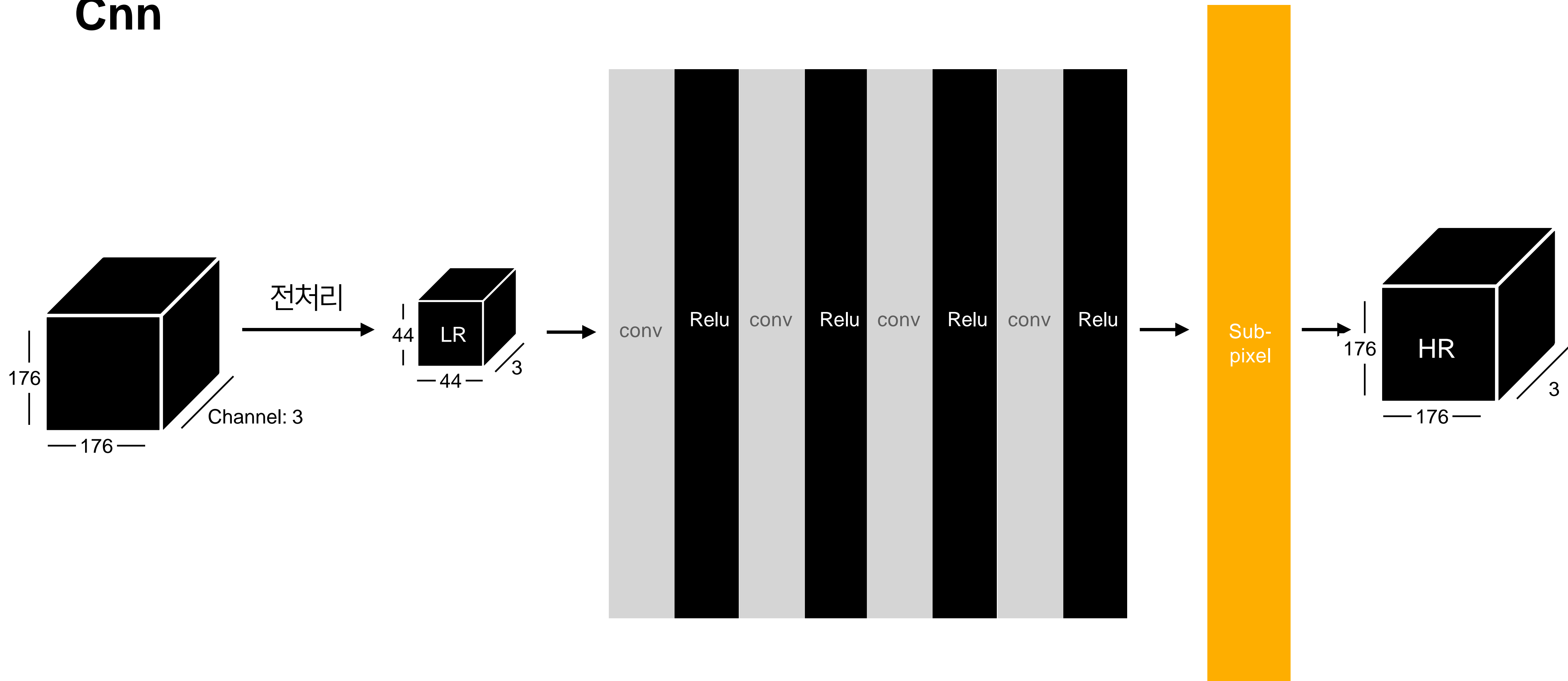
- train_size = 162770
- validation_size = 19867
- test_size = 19962

- Input feature = (44,44,3)
- Target output = (176,176,3)

* 데이터 특이사항
cv2로 이미지를 읽기 때문에 B,G,R 의 채널 순서

사용한 머신러닝 모델

Cnn



개발에 사용한 툴



Preprocess

Numpy Pickle
Scipy PyTorch
openCV matplotlib

Convolution Layer

Numpy

Sub-pixel Layer

Numpy
Pytorch

학습 과정

* 학습하기 전에 가장 효과적인 optimizer를 찾기 위해 분석한 과정

train_size = 162770

batch_size = 10

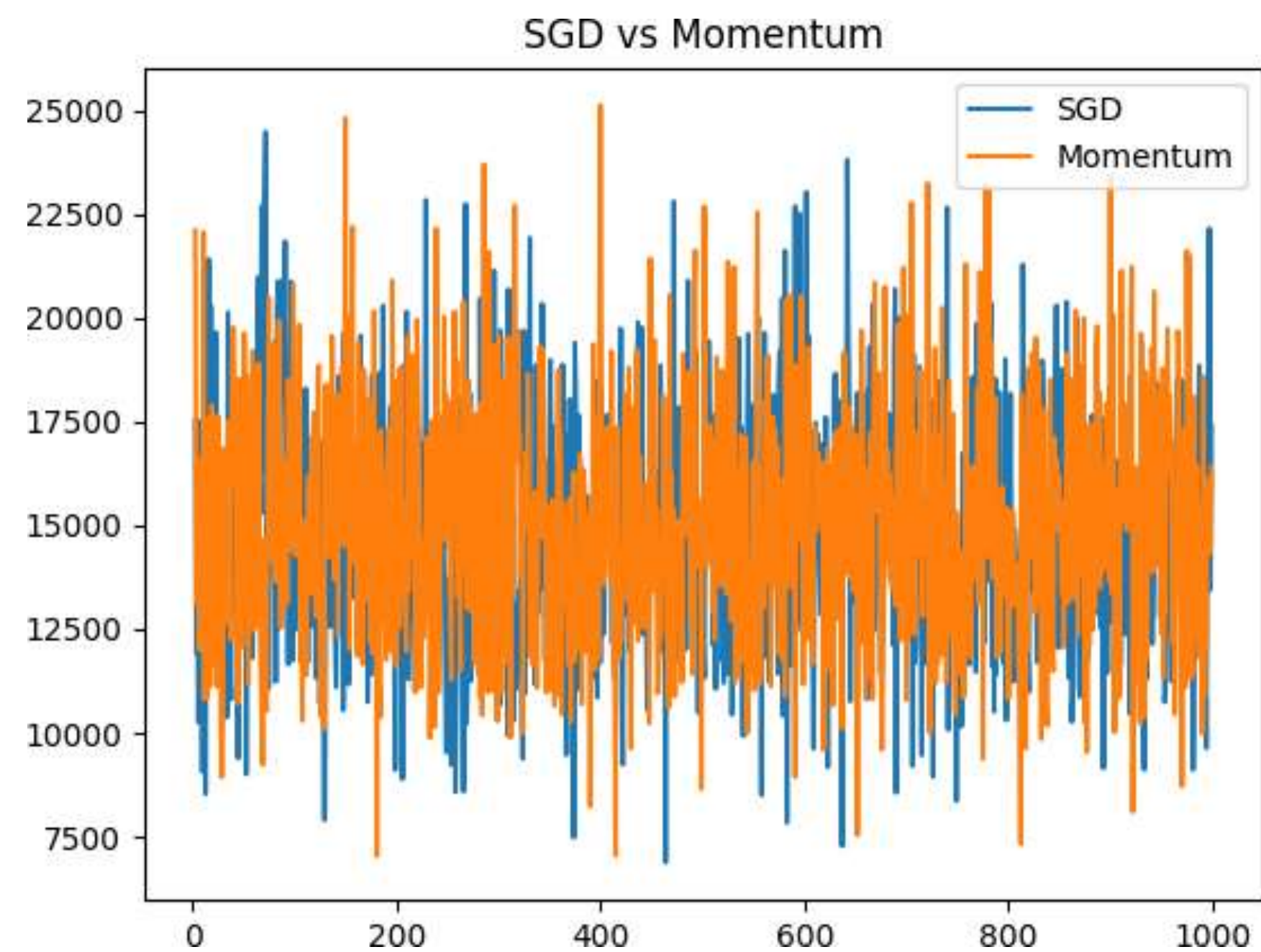
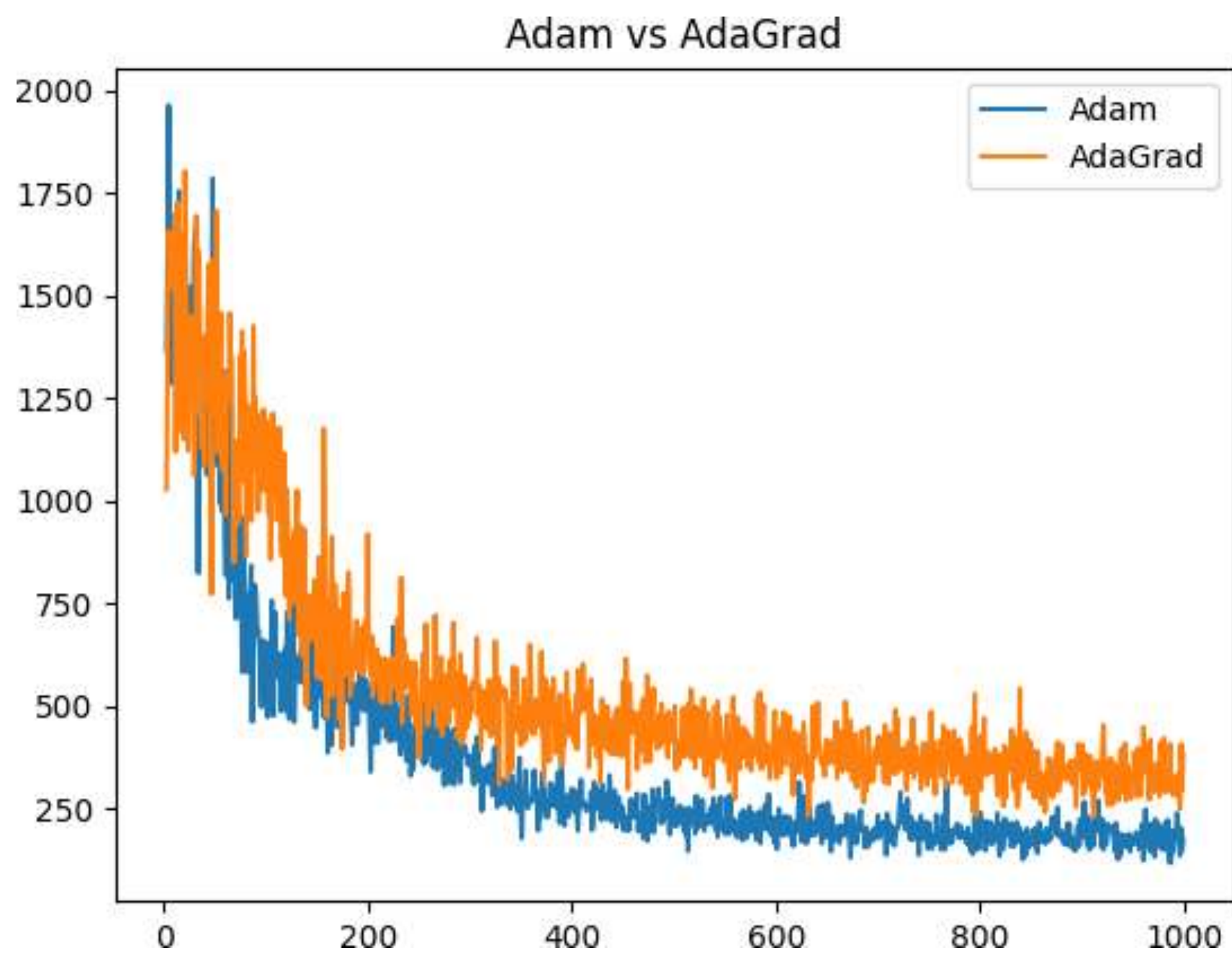
iters_num = 1000

크기 10의 batch를 1000회 학습

Optimizer 종류

1. SGD
2. Adam
3. AdaGrad
4. Momentum

각Optimizer 사용시mse값 비교



Adam

LR



HR



ANSWER

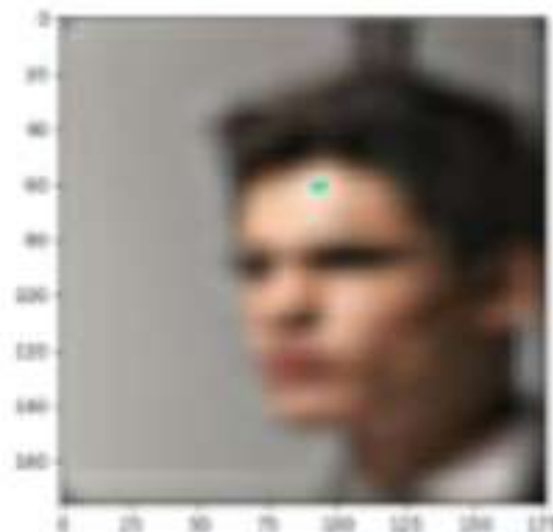


AdaGrad

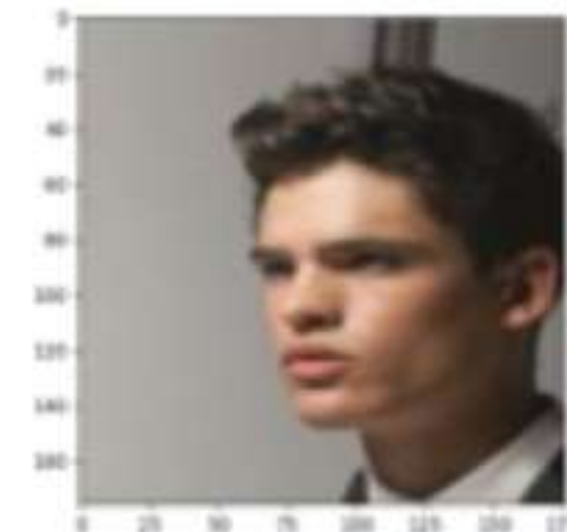
LR



HR



ANSWER



Adam을 사용해서 돌린 결과

train_size = 162770

validation_size=182637

test_size = 202599

batch_size= 10

iters_num = 43114

크기 10의 batch를 43114회 학습

PSNR
28.6

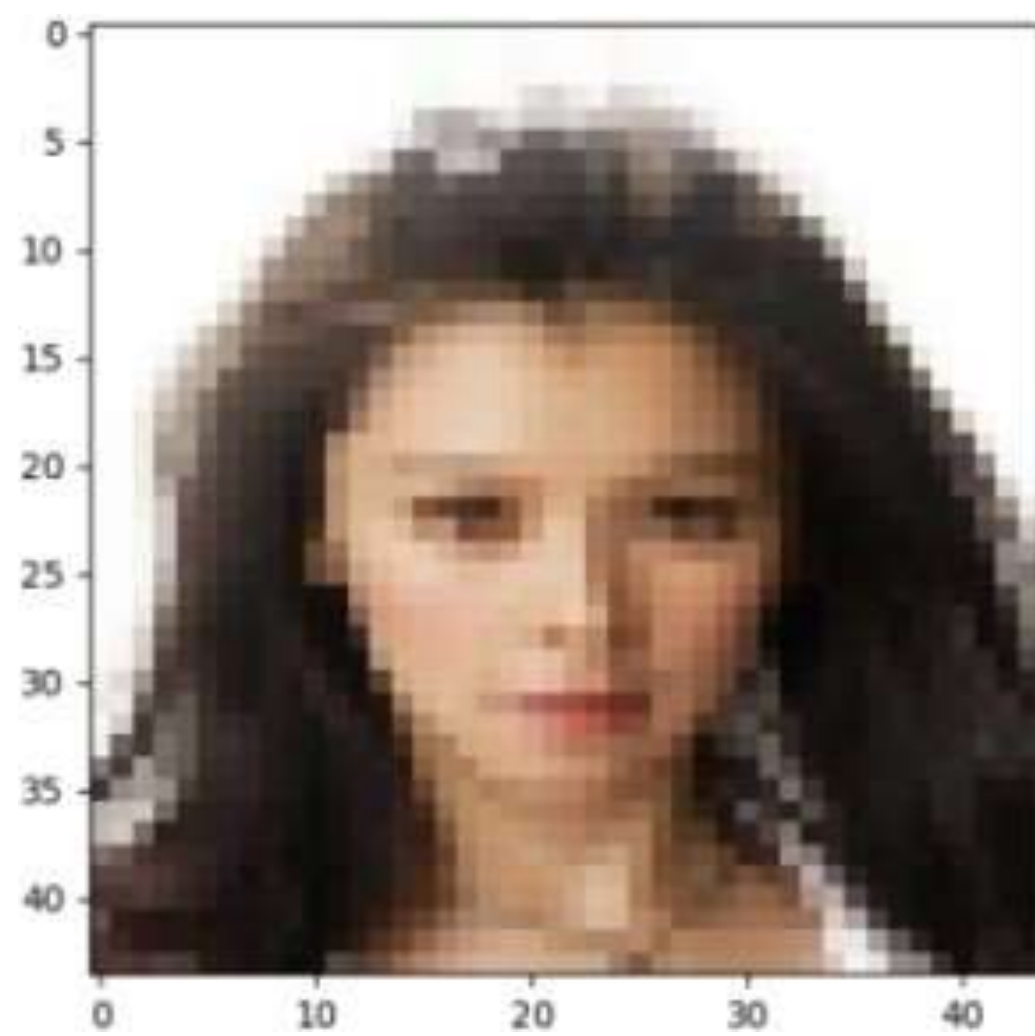
PSNR

평가지표

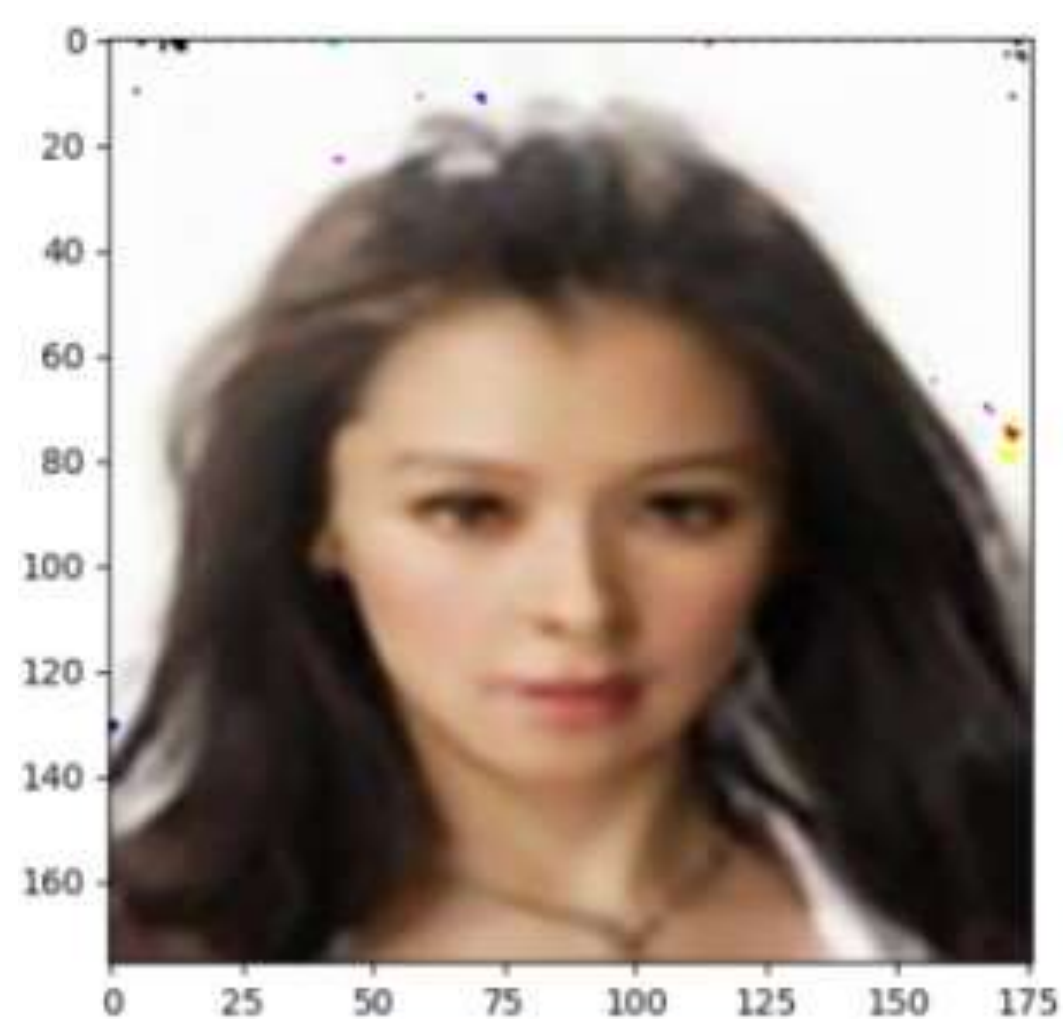
$$PSNR = 10 \times \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

```
# 입력 x 는 (B, 1, 44, 44) 저화질 이미지  
# 입력 t 는 answer images (B, 1, 176, 176)  
def accuracy(self, x, t):  
    mse = self.loss(x, t)  
    return 10*np.log10(255*255/mse)
```

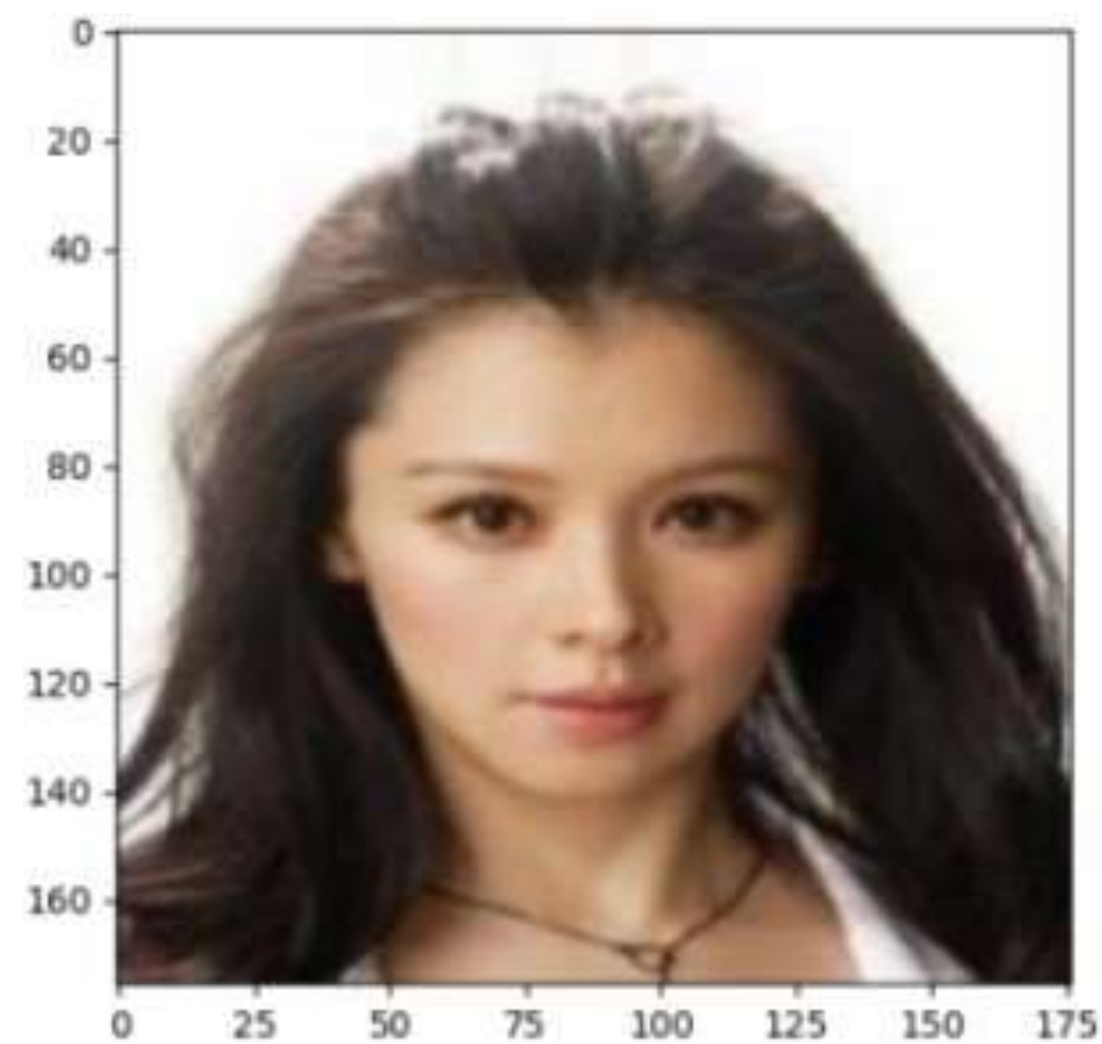
결과 이미지



LR



HR



ANSWER

독립적 네트워크

RGB 값을 동시에 학습하지 않고, 각 채널 값을 독립적으로 학습하는 3가지 네트워크 구현

```
train_size = 162770
```

```
validation_size=182637
```

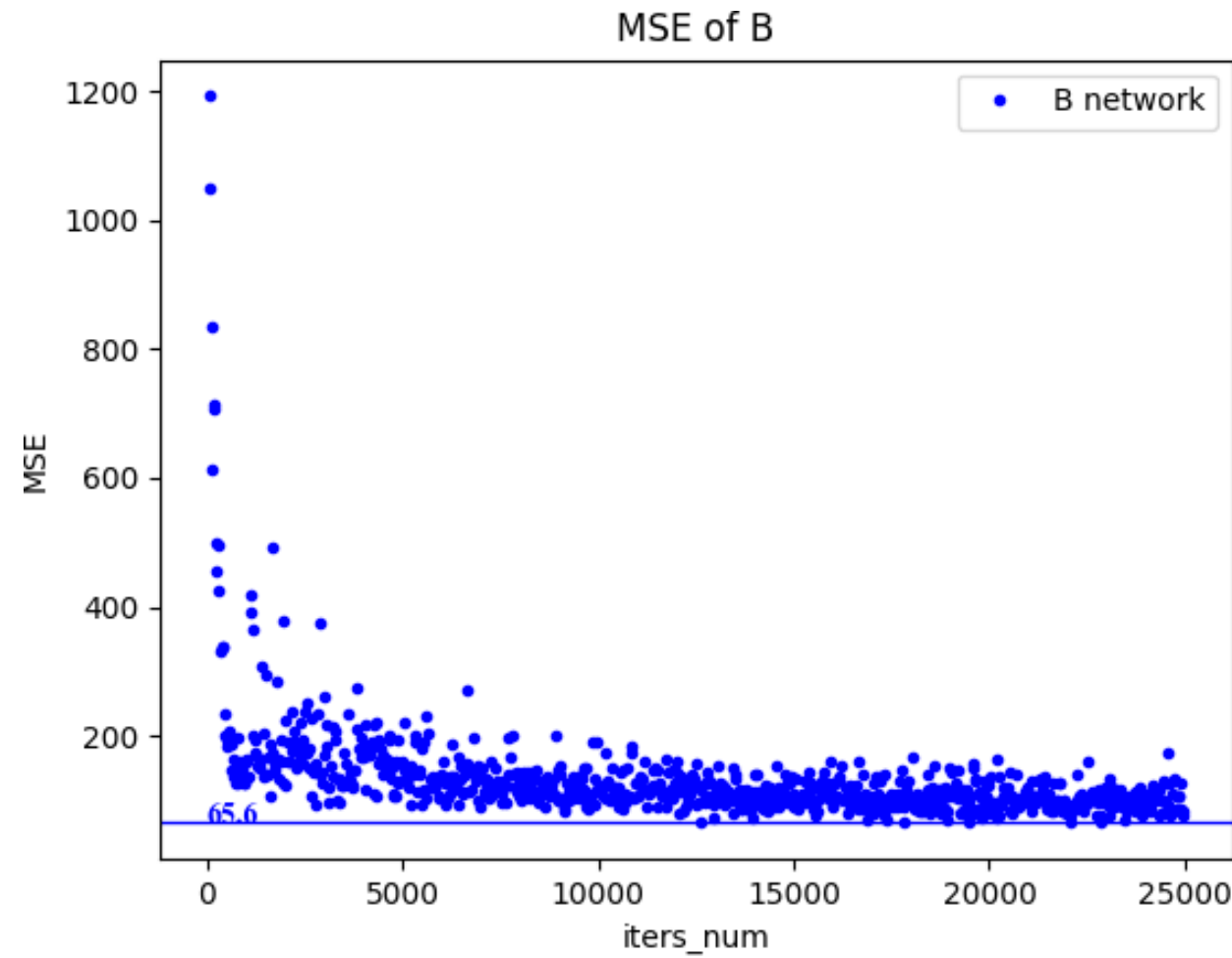
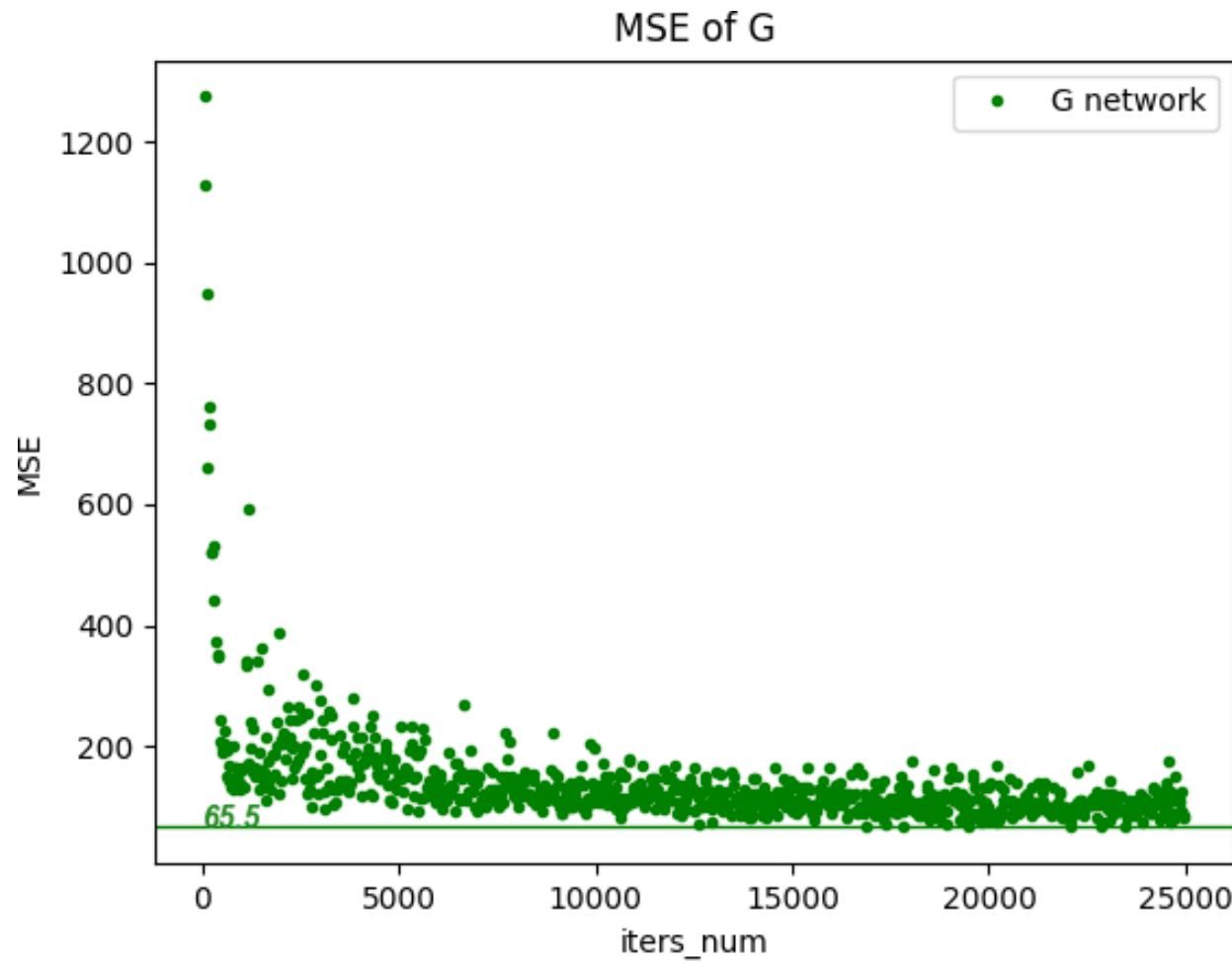
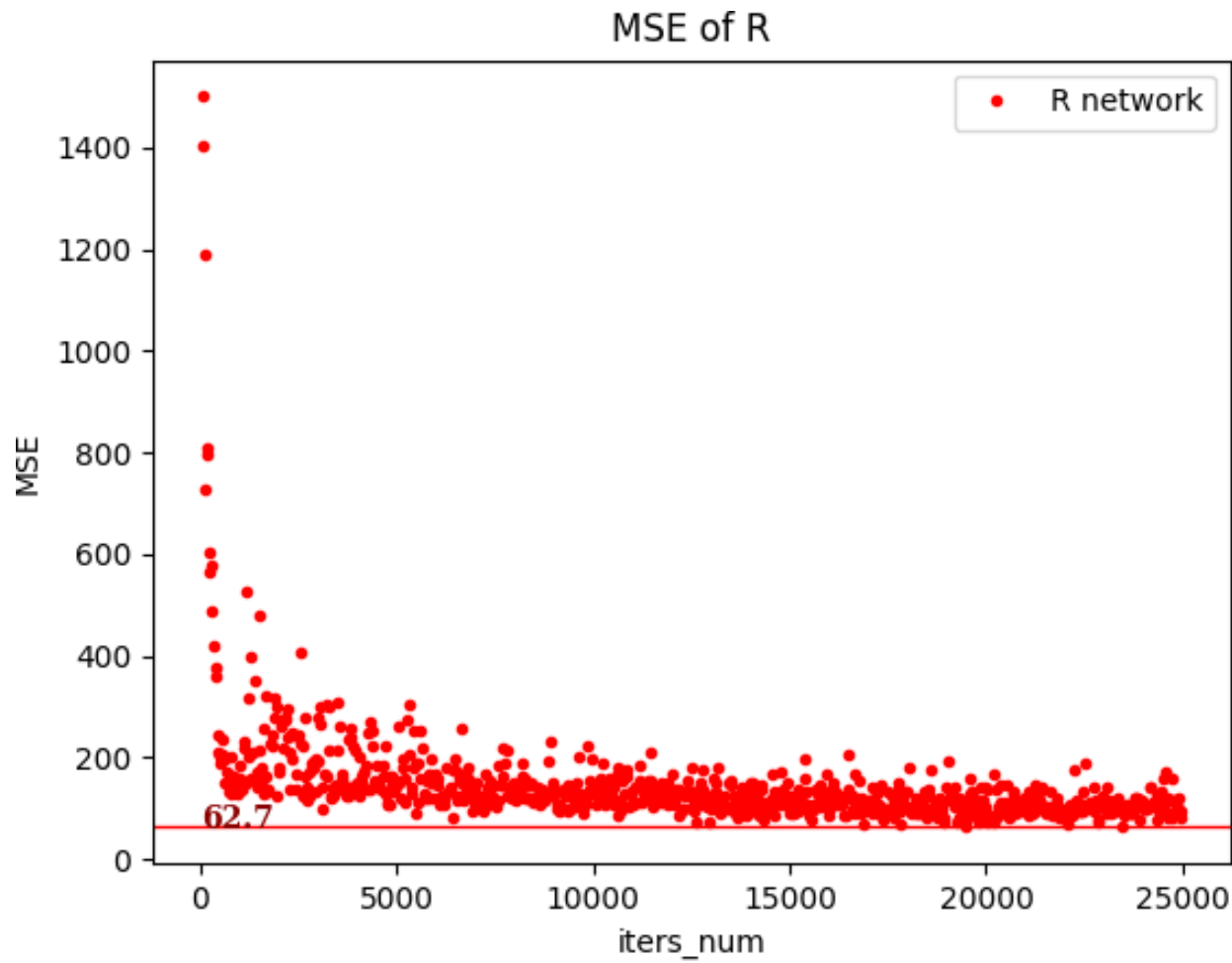
```
test_size = 202599
```

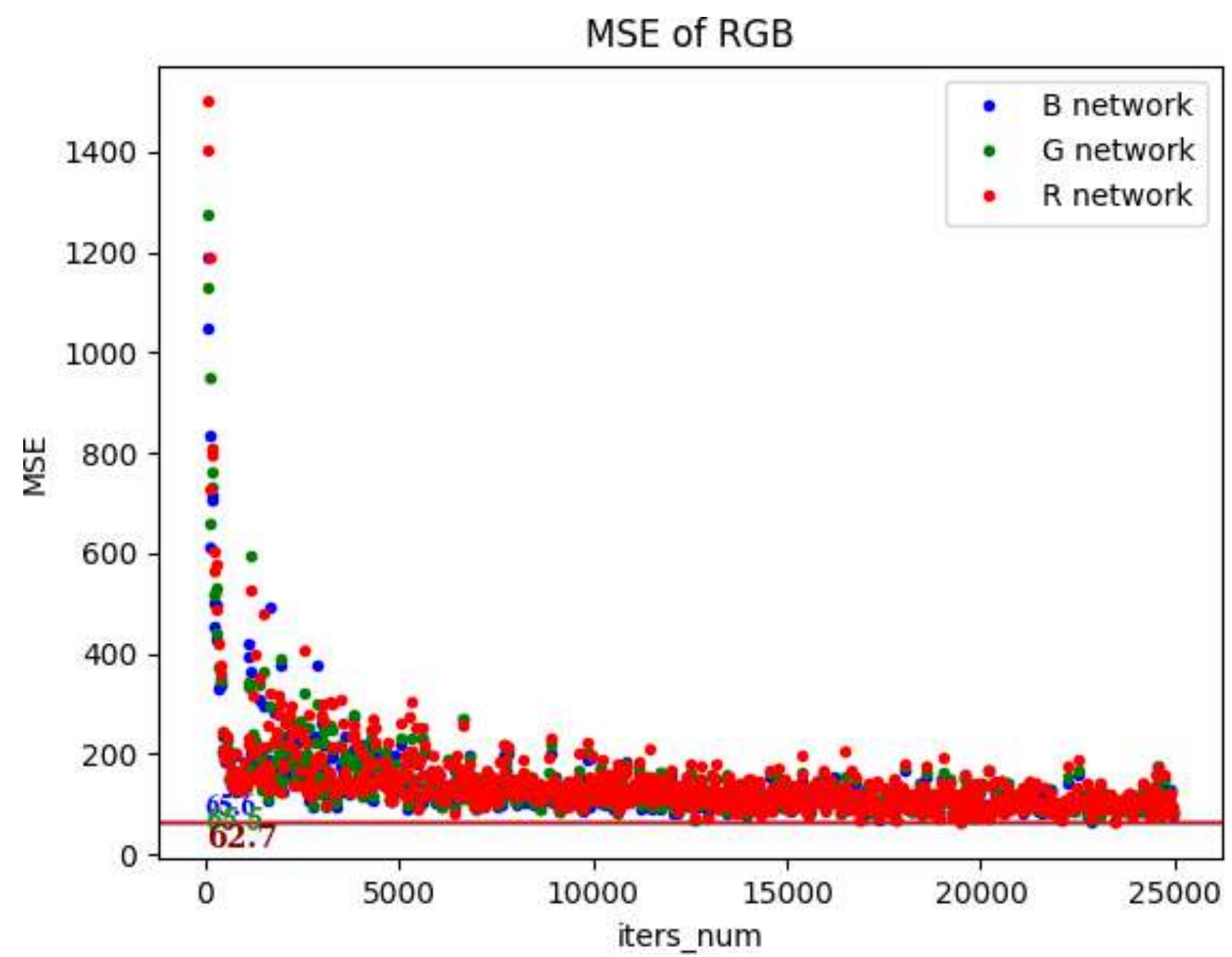
```
batch_size= 10
```

```
iters_num = 25000
```

크기 10의 batch를 25000회 학습

각 네트워크를 따로 학습시켰을 때의 mse변화

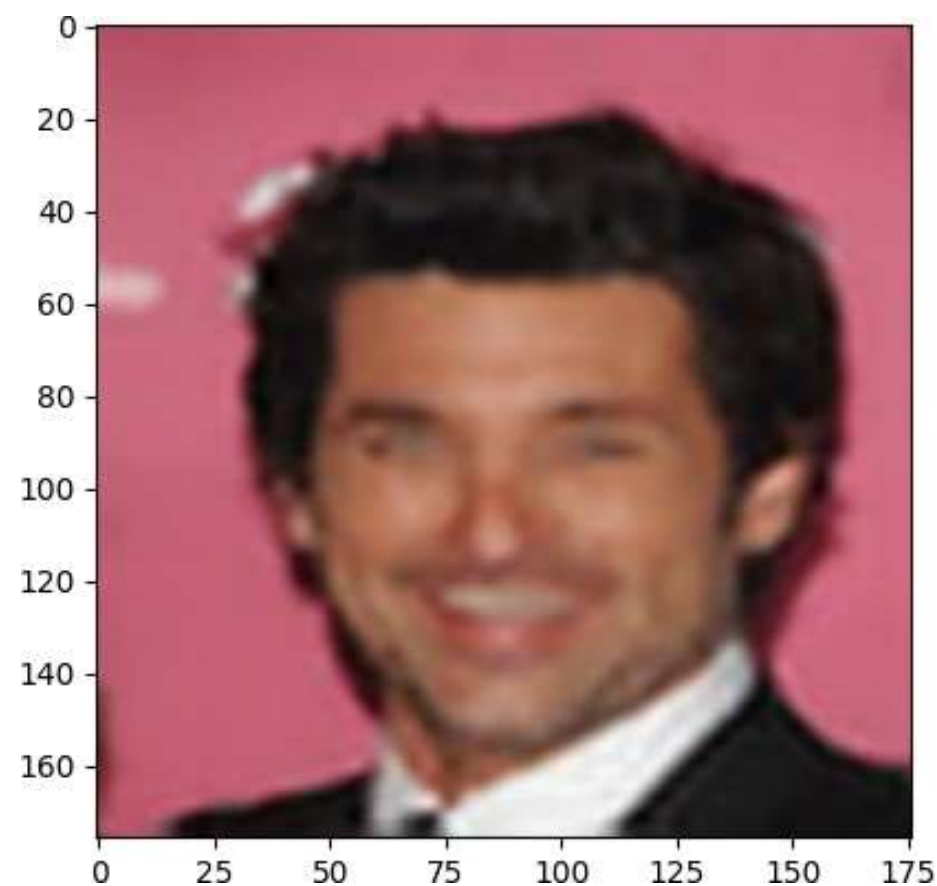




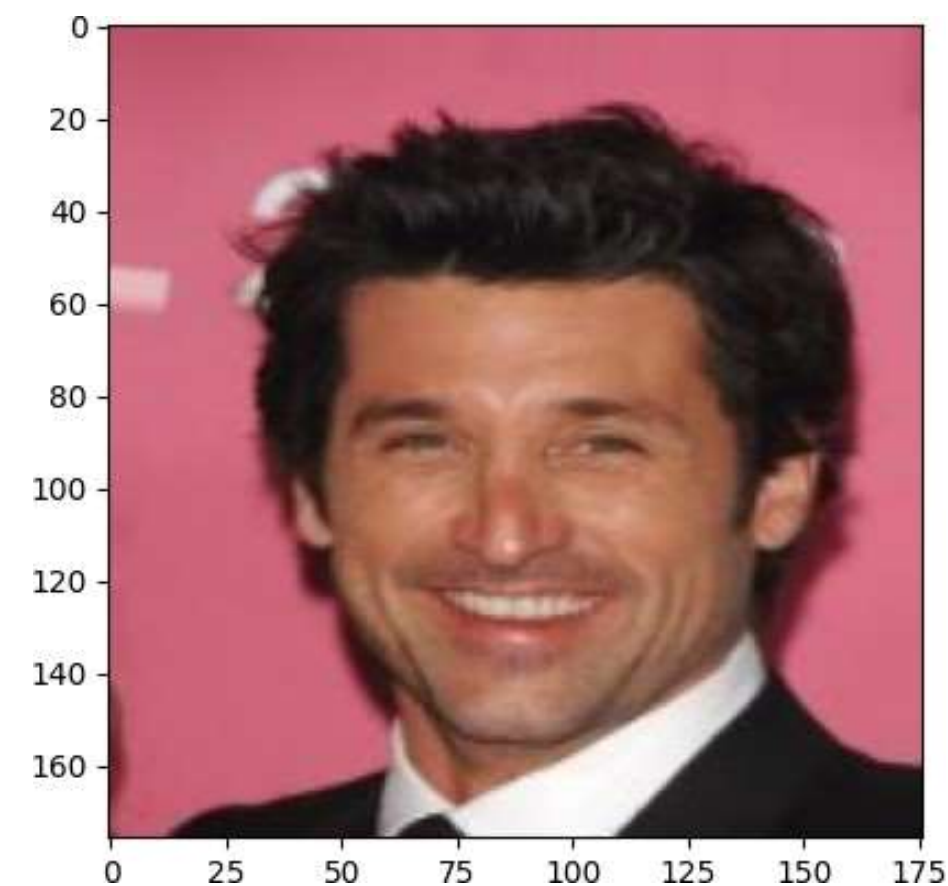
독립적 네트워크 결과 이미지



LR



HR



ANSWER

PSNR = 26.8

구현한 부분

- 이미지 전처리 preprocess.py
- 3_colors_network_train.py
- 3_colors_network_test.py
- 3_colors_network_print.py
- optimizer_compare_train.py
- optimizer_compare_test.py
- optimizer_compare_print.py
- sr_train.py
- sr_test.py
- sr_print.py
- sr_net.py test 와 test2 메소드
- common.function.MSE
- common.function.PSNR
- common.layer.SubPixel 구현 (pytorch의 PixelShuffle 과 PixelUnshuffle 이용)

차용한 부분

- Preprocess 중scikit-image의 pyramid-reduce 함수
- convolution layer
 - im2col 함수
 - col2im 함수
- relu activation 함수
- optimizer 코드
 - Adam
 - SGD
 - AdaGrad
 - Momentum
- sr_net 의 전체적인 학습flow
 - https://github.com/youbeebee/deeplearning_from_scratch
- sub-pixel layer
 - pytorch의 PixelShuffle 과PixelUnshuffle 함수

소스코드 및 데이터url

- https://github.com/zungwooker/2022_ai_project_super_resolution.git
- <https://www.kaggle.com/datasets/jessicali9530/celeba-dataset?resource=download>

감사합니다.