CV 2팀 : 맑은 눈의 광인 👀

이미지 초해상도 프로젝트

15기 염윤석 장수혁 최민경
16기 박민규

Contents

Project Introduction

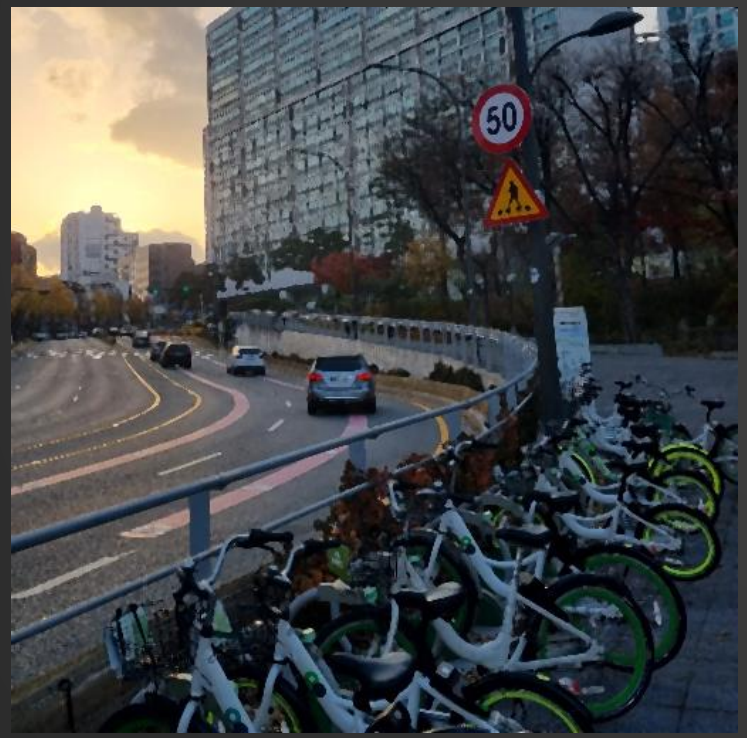
1
2
3
4
5

1 목표

저해상도(512X512) → 고해상도(2048X2048) 생성을 위한 AI 알고리즘 개발

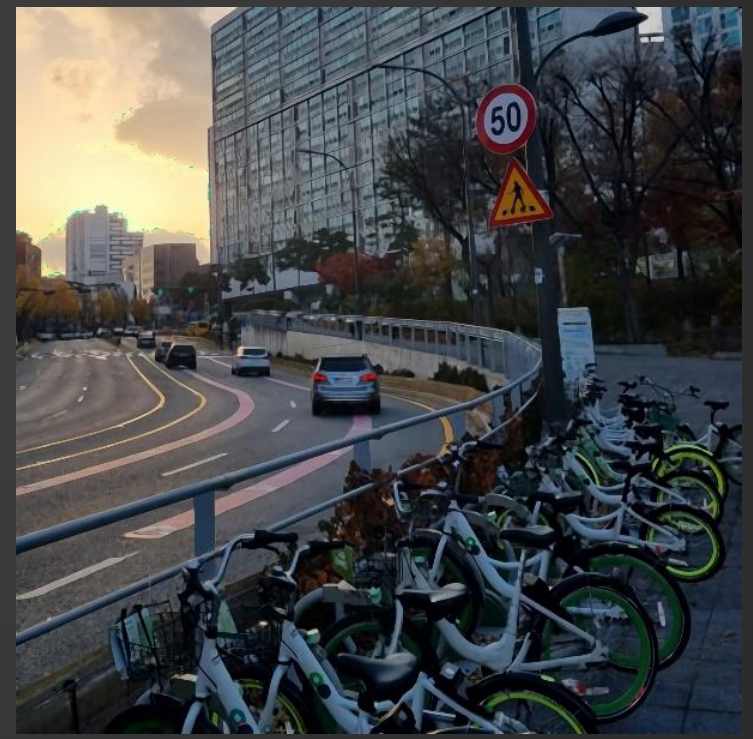
2 평가산식

PSNR (Peak Signal-to-Noise Ratio)



저해상도 이미지 (512X512)

초해상화 알고리즘



고해상도 이미지 (2048X2048)

Data Explanation

1
2
3
4
5

1 Training Data

Low-Resolution image(512 x 512) = 1640장
High-Resolution image(2048 x 2048 = 1640장

** 저해상도와 고해상도 이미지는 쌍(pair)으로 존재

LR	HR
./train/lr/0000.png	./train/hr/0000.png
./train/lr/0001.png	./train/hr/0001.png
./train/lr/0002.png	./train/hr/0002.png
./train/lr/0003.png	./train/hr/0003.png
./train/lr/0004.png	./train/hr/0004.png
./train/lr/0005.png	./train/hr/0005.png
./train/lr/0006.png	./train/hr/0006.png
./train/lr/0007.png	./train/hr/0007.png
./train/lr/0008.png	./train/hr/0008.png
./train/lr/0009.png	./train/hr/0009.png
./train/lr/0010.png	./train/hr/0010.png
./train/lr/0011.png	./train/hr/0011.png

2 Test Data

Low-Resolution image(512 x 512) = 18장

LR
./test/lr/20000.png
./test/lr/20001.png
./test/lr/20002.png
./test/lr/20003.png
./test/lr/20004.png

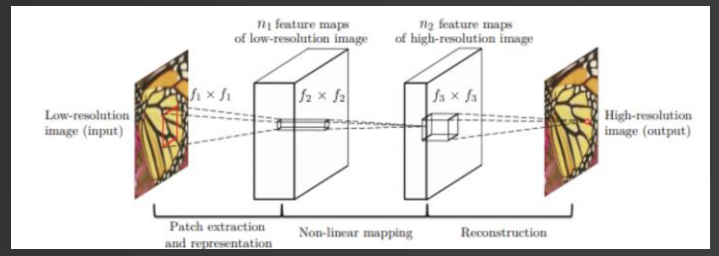
Model Reference Review

1
2
3
4
5

*초해상도 분야 최초의 딥러닝 모델

*전처리 부터 후처리까지 모델 학습에 포함 → 속도 향상 및 성능 향상

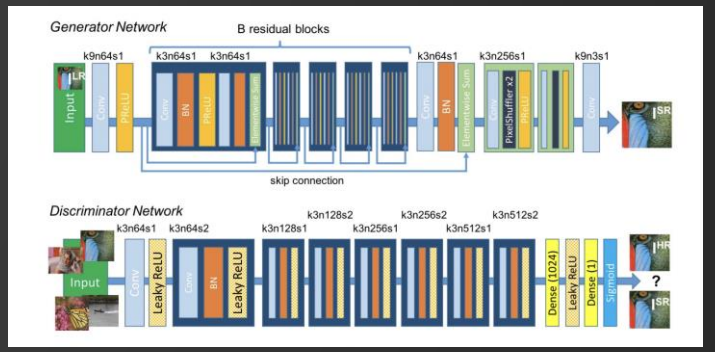
SRCNN



SRGAN

*Generator vs Discriminator

*기존 MSE loss 기반 모델보다 더 실제적인 시각효과 질감 형성

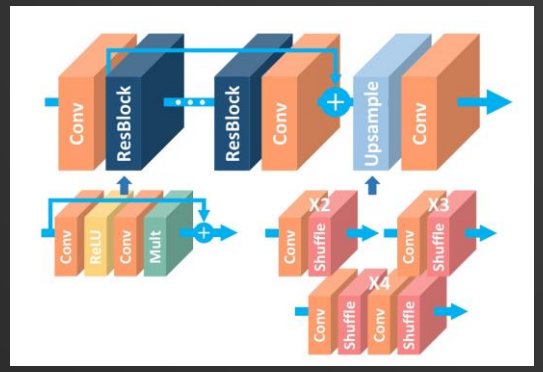


*Residual block을 활용한 RESNET 활용

* 배치 정규화 제거하여 성능향상 및 메모리 사용 감소

*Residual block 증가 → feature map의 증가 → 모델의 capacity 증가

EDSR



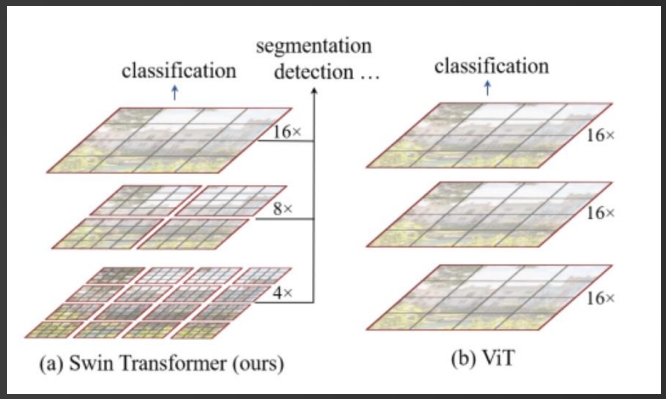
Model Reference Review

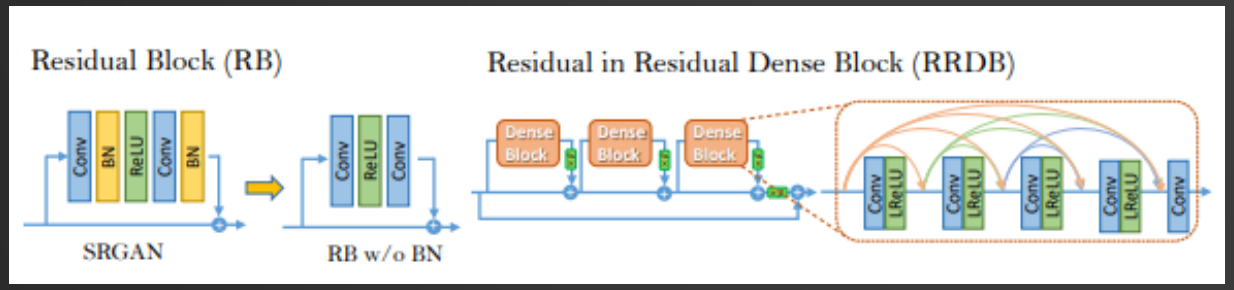
1
2
3
4
5

SwinIR

*Tranformer를 활용한 ViT 모델에서 향상된 Swin Transformer 활용

*Swin Transformer = CNN + ViT





RRDB

* SRGAN 모델을 향상시킨 ESRGAN 모델의 일부분

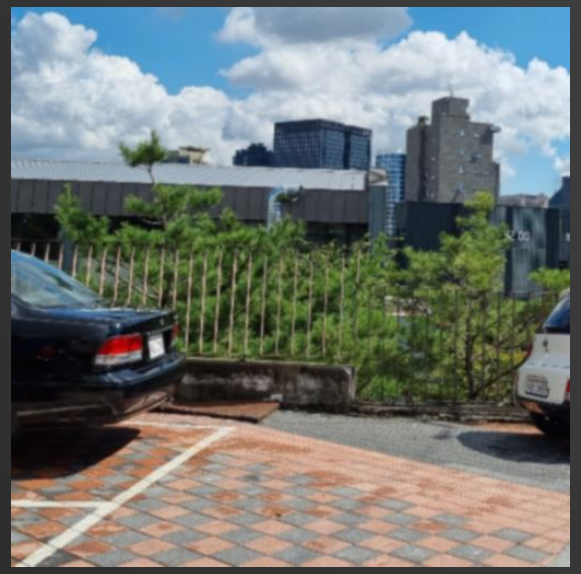
*Residual Block을 이중으로 중첩

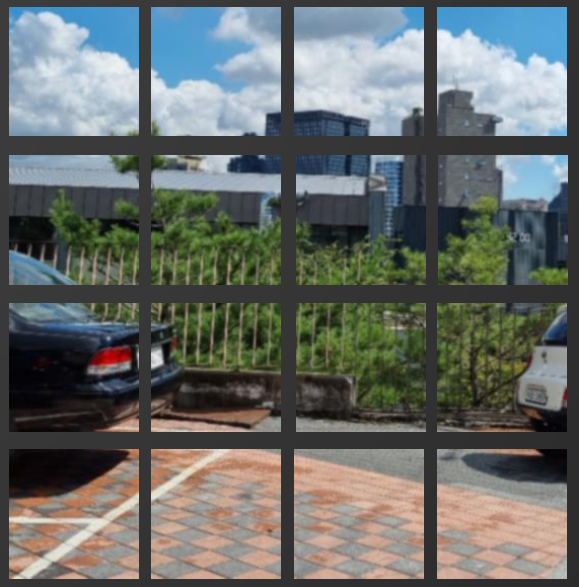
*RB 자체를 여러 층으로 쌓아서 Dense block 형태로 형성

Model learning - Patch

```python
def making_patches(img_path,patch_size, train = True, is_lr = True):
  img_name = img_path.split("/")[-1][:-4]
  sample_large_img = cv2.imread(img_path) # 여기서 BGR로 읽고 잘라서 그럼

  patches_img = patchify(sample_large_img, (patch_size,patch_size,3), step  = patch_size) ##patch 형성


  for i in range(patches_img.shape[0]):
    for j in range(patches_img.shape[1]):

      #one each patch
      single_patch_img = patches_img[i,j, :,:, :]#patch 한장
      single_patch_img = np.squeeze(single_patch_img, axis = 0)

      #saving on patch
      if train :
        if is_lr:
          lr_file = "/content/drive/MyDrive/patches_new/lr/{}_{}_{}.png".format(img_name, i,j)
          if not os.path.isfile(lr_file):
            cv2.imwrite(lr_file,single_patch_img)
        else:
          hr_file = "/content/drive/MyDrive/patches_new/hr/{}_{}_{}.png".format(img_name, i,j)
          if not os.path.isfile(hr_file):
            cv2.imwrite(hr_file,single_patch_img)

      else:
        test_file ="./patches_new/test_new/{}_{}_{}.png".format(img_name, i,j)
        # test_file ="/content/drive/MyDrive/test_new/{}_{}_{}.png".format(img_name, i,j)
        if not os.path.isfile(test_file):
          cv2.imwrite(test_file,single_patch_img)

  print("processing :", img_name)
```

* "patchify" 모듈을 활용하여 patch를 만들 수 있는 함수 형성
* 원본 이미지 → 16장의 patch 형성
* 1 patch size
    * 저해상도 이미지 = 128 x 128
    * 고해상도 이미지 = 512 x 512
* 총 train image 1640장 —> 1640 X 16 = 26240 장

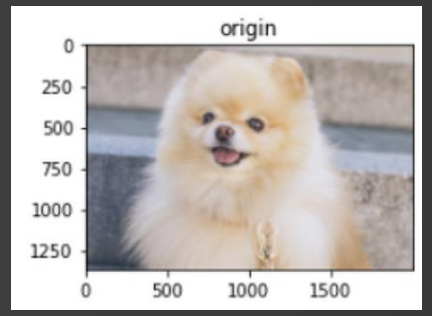


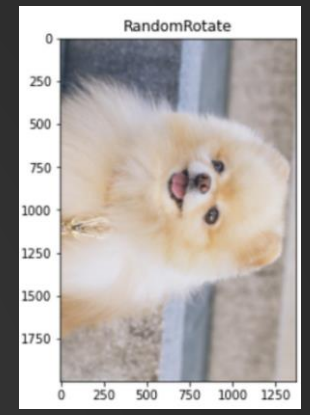


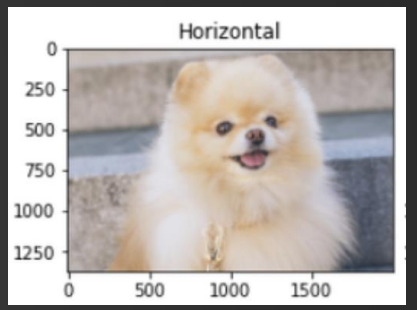
Model Learning - Data Augmentation

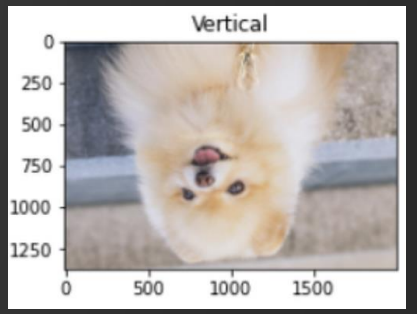
1
2
3
4
5

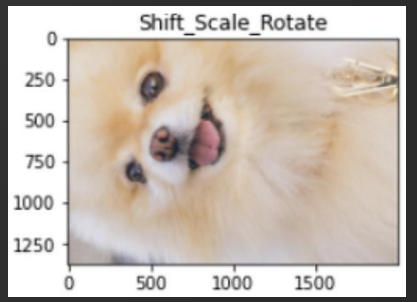
```
def get_train_transform():

  #모두 Non-destructive transformations <- 정보를 추가하거나 손실하지 않게
  return A.Compose([
      A.RandomRotate90(p=0.7),
      A.OneOf([
          A.HorizontalFlip(p=1), #수평회전
          A.VerticalFlip(p=1),
          A.ShiftScaleRotate(p=1)],p=0.7), #수직회전
      A.Transpose(p=0.7), #X,Y축 반전
      ToTensorV2(p = 1.0)],
      additional_targets = {"image" : "image", "label":"image"}
  )
```











- Transpose : x, y축 반전
- Flip : 좌우 반전
- RandomRotate : 랜덤으로 회전
- ShiftScaleRotate : 회전하면서 줌인

Model learning - Loss

- PSNR (Peak Signal-to-Noise Ratio): 최대 신호에서 잡음 비율

사용 목적 : 생성 혹은 압축된 영상의 화질에 대한 "손실 정보"를 평가

손실이 적을수록, 즉 화질이 좋을수록 높은 값

무손실일 경우, MSE= 0 일 때는 정의가 불가

→ L2 loss

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

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N}$$

Model learning - Loss

- PSNR (Peak Signal-to-Noise Ratio): 최대 신호에서 잡음 비율

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

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N}$$

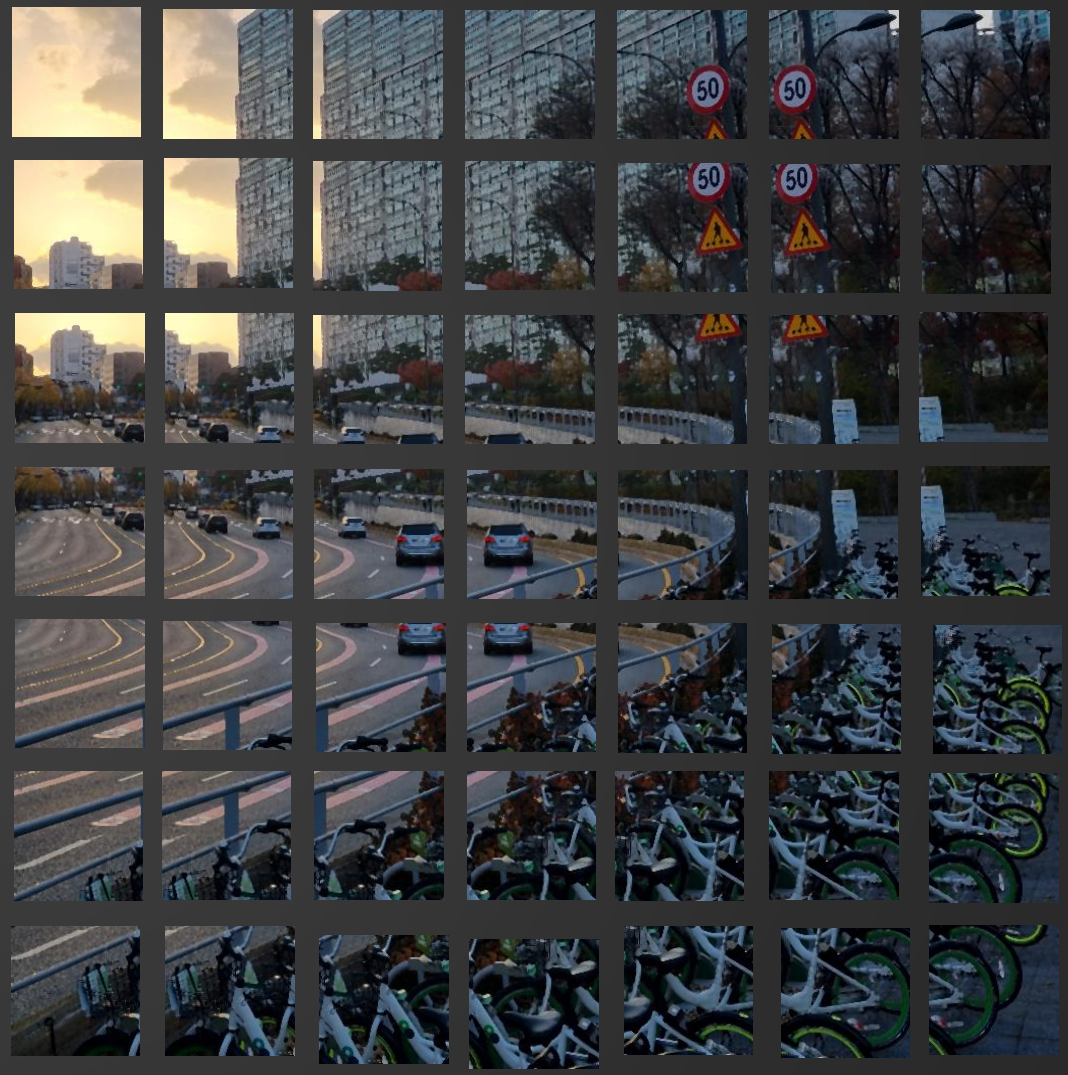
학습에서는 수렴성이 좋지 않다 → L1 Loss를 혼용해서 사용

L1 : L2 = 2 : 1

Test Performance - Patch Overlap

512 X 512 → 128 X 128의 49장의 patch → test input

* Train시 128x128의 이미지들로 학습하였기에…

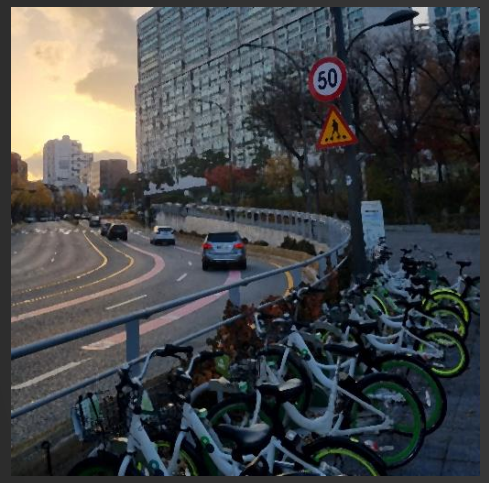


Test Performance - Patch Overlap

1
2
3
4
5

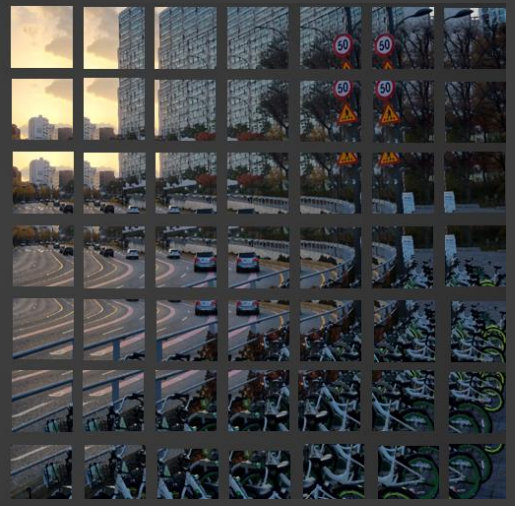
모델 output

512x512 고해상도의 49장의 patch들 → 중첩되는 부분은 평균화 → 2048x2048 고해상도 이미지



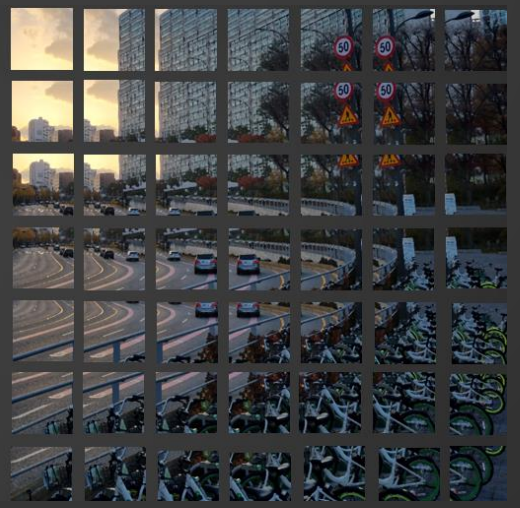
<test>
512x512

<test patches>
128x128 – 49장



Network

<test patches results>
512x512 – 49장

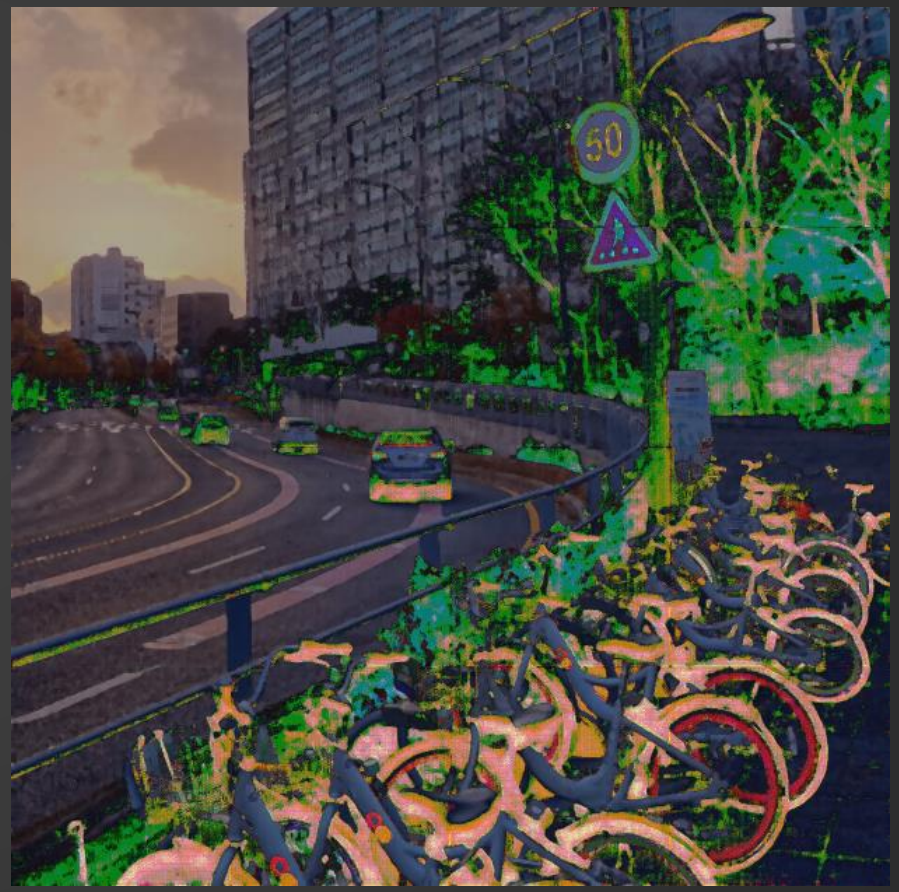


<test result>
2048x2048

Test Performance

SRGAN : 12.1875

<원본>

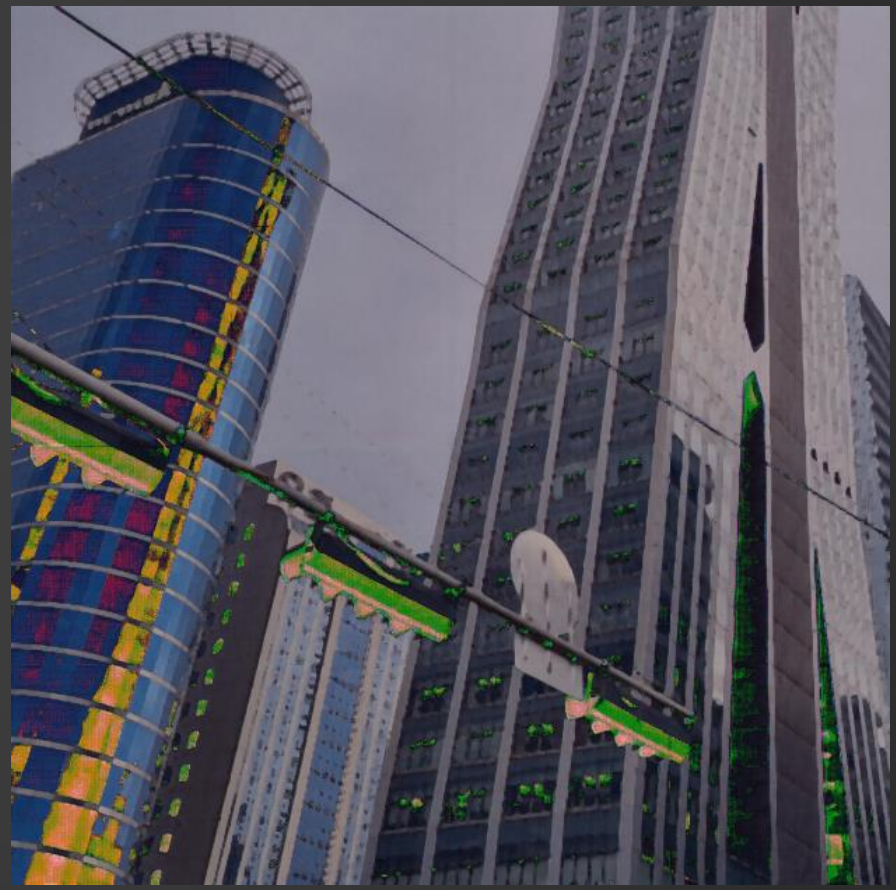


<SRGAN 결과>

Test Performance

SRGAN : 12.1875

<원본>

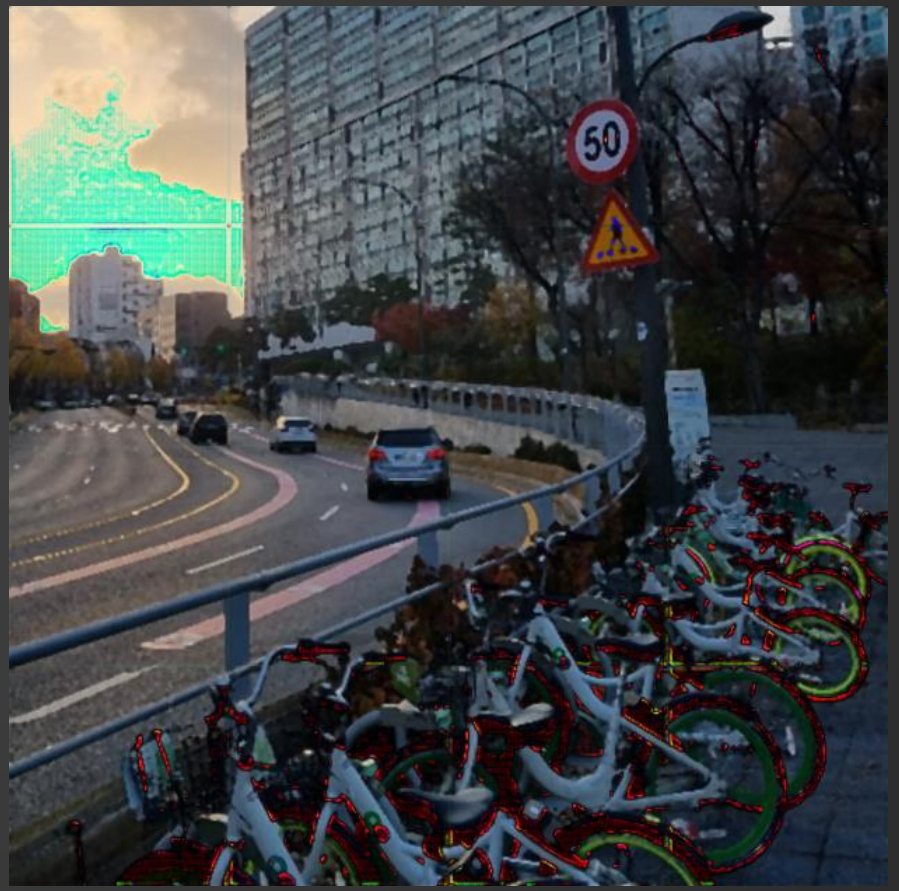


<SRGAN 결과>

Test Performance

EDSR : 20.0568

<원본>

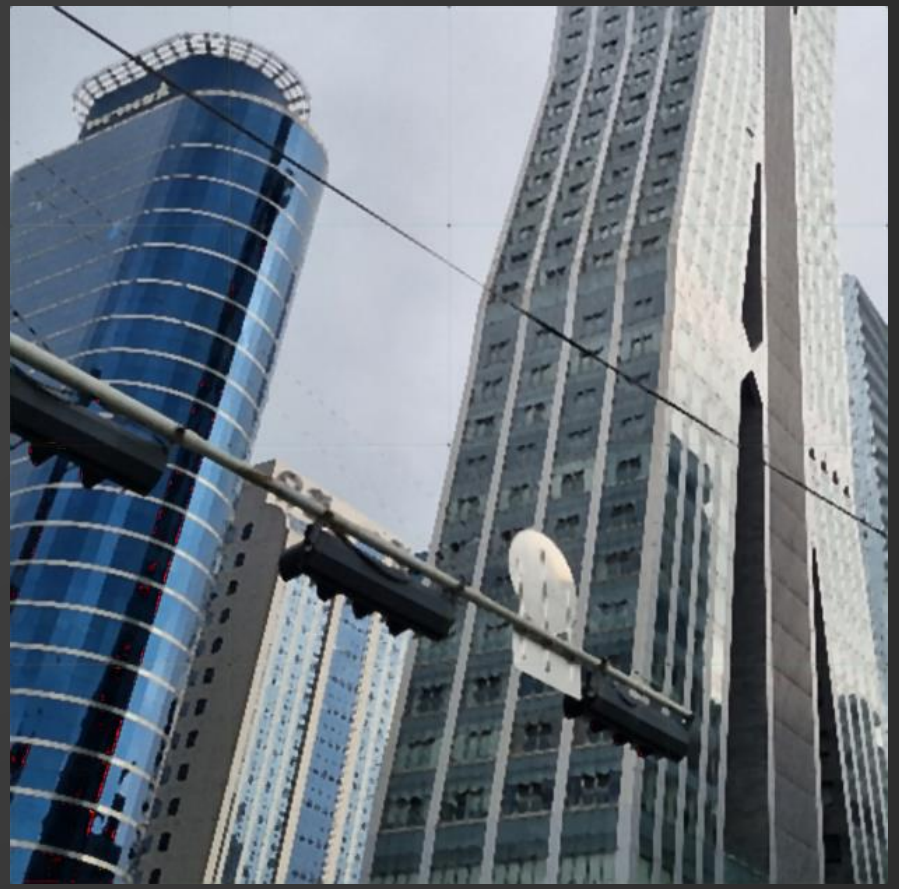


<EDSR 결과>

Test Performance

EDSR : 20.0568

<원본>



<EDSR 결과>

Test Performance

SWinIR : 22.1805

<원본>

<SwinIR 결과>

Test Performance

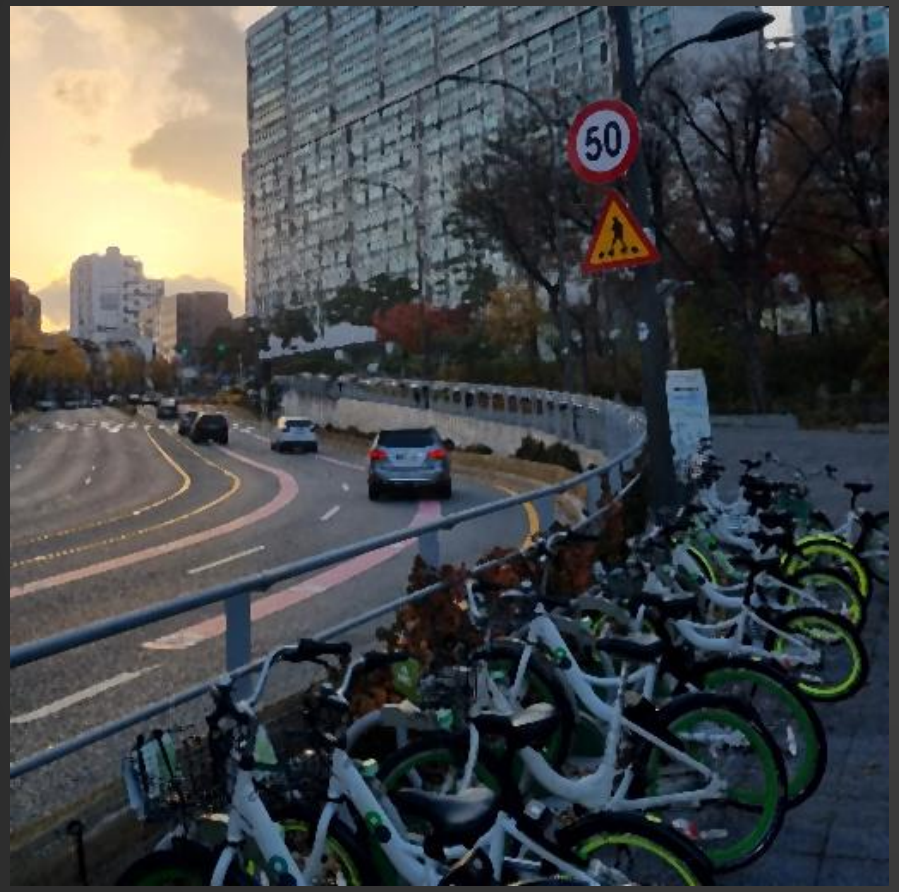
SWinIR : 22.1805

<원본>

<SwinIR 결과>

Test Performance

RRDB : 23.2260



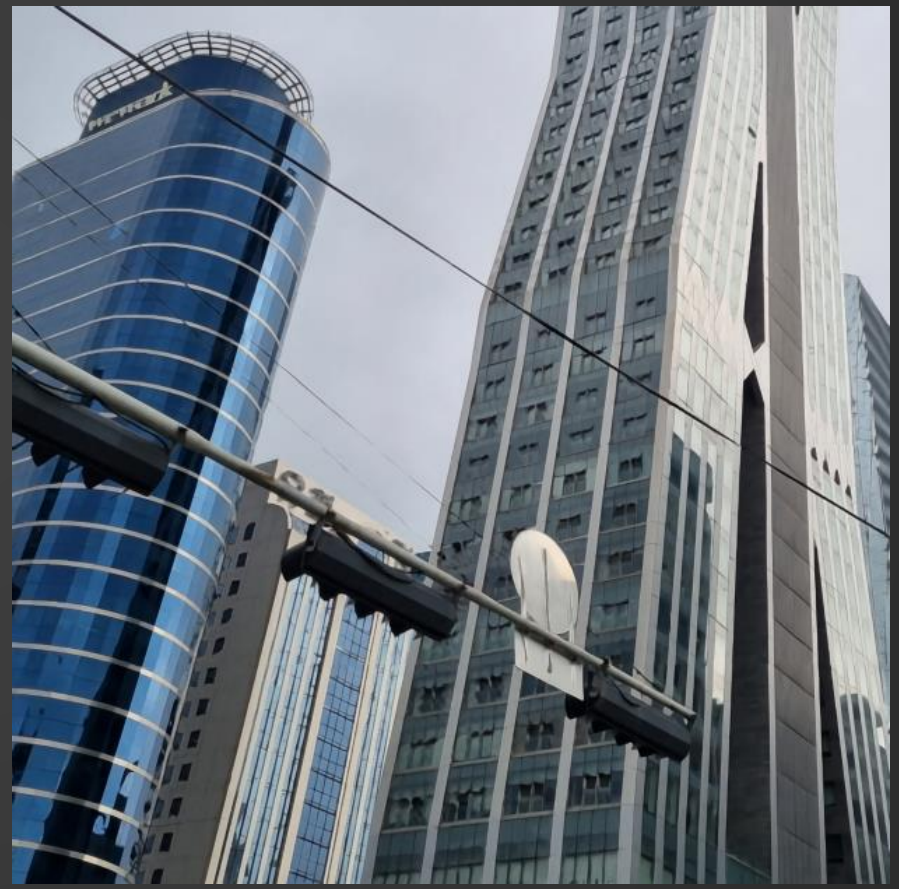
<원본>

<RRDB 결과>

Test Performance

RRDB : 23.2260

<원본>



<RRDB 결과>

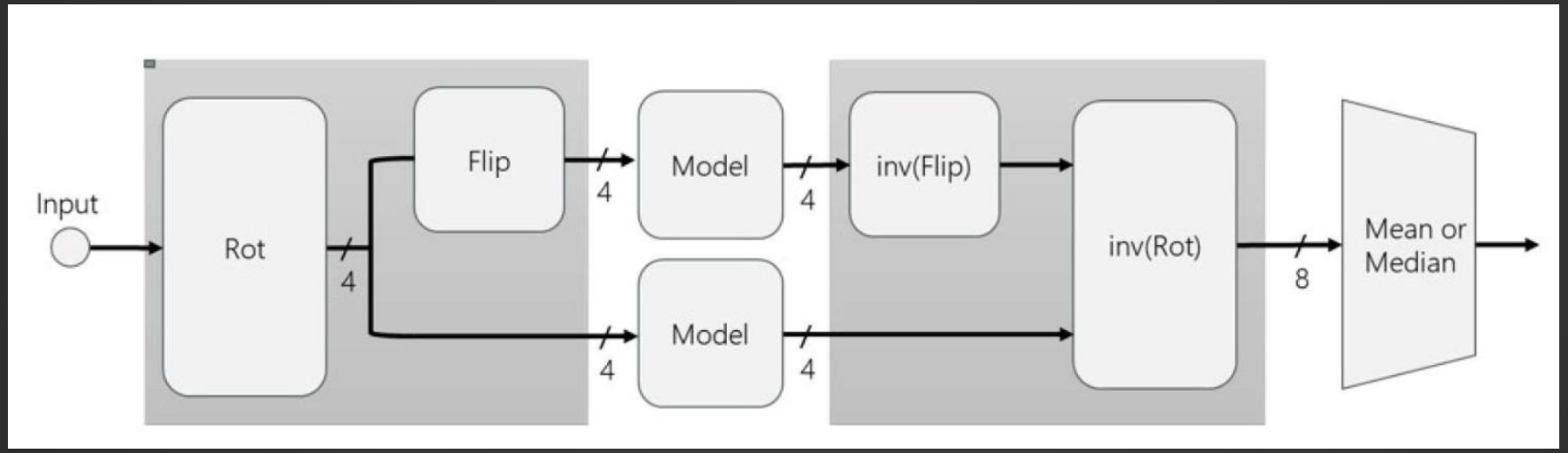
Test Performance

1
2
3
4
5

RRDB+ : RRDB에 Geometric Self Ensemble을 더한 모델

geometric self ensemble :

이미지에 flip & rotate 적용한 이미지를 네트워크에 통과시킨 후, 모두 합쳐서 평균을 내는 과정



In <Enhanced Deep Residual Networks for Single Image Super-Resolution 4.3>

Test Performance

RRDB+ : 23.4081

<원본>

<RRDB+ 결과>

Test Performance

RRDB+ : 23.4081

<원본>

<RRDB+ 결과>

Discussion

1
2
3
4
5

SwinIR

항상 상위권의 좋은 모델이지만 Transformer 특성상 한정적인 데이터에 취약

RRDB

Batch Normalization을 제거하고 Dense network를 형성하였기에

안정적인 정보전달과 학습이 가능

→ 가장 좋은 성능

RRDB+

Geometric Self Ensemble → 추가적인 학습 필요 X
이미지 초해상도라는 긴 시간이 필요한 Task에서 효율적으로 성능을 높일 수 있었다.

Discussion - Limitation

- GPU자원 부족

  충분한 epoch 수로 학습을 진행 불가 (learning epoch = 7)

- 어려운 초행길

  여러 model 논문 리뷰 후, 모델 사용 자체에 어려움

  해당 팀만의 경쟁적인 차별점 부족

- Pretrained Model 소극적 활용

  다른 수상작들처럼 큰 데이터로 학습된 pretrained model을 충분히 사용하지 못함

Conclusion

1
2
3
4
5

38	민규라면	23.40812	1	6일 전

최종 38등　PSNR : 23.40812

Conclusion

1
2
3
4
5

윤석 " CV 분야 첫걸음이었지만, 공부와 결과 그리고 팀워크 모두를 잡은 기분 좋은 프로젝트였다."

수혁 " 프로젝트지만 논문스터디를 병행하여 기초부터 다질 수 있어 좋았다.
작업물의 결과가 선명하게 보였기에, 동기부여와 성취감이 더 컸던 프로젝트였다."

민경 " 다양한 모델의 구현원리를 알아보며 초해상도의 기초부터 다질 수 있었던 프로젝트였다.
능력 출중한 팀원들 덕분에 재밌게 프로젝트를 진행할 수 있었다."

민규 " 많이 부족했지만, 팀원들이 이끌어줘서 성공적으로 프로젝트를 완수한 뜻깊은 경험이었다."

Thank You