코딩으로 생활 속 문제 해결:
파이썬을 통한 인공지능 맛보기

초해상화(Super Resolution) 해보기
이미지 인식 (Detectron2) 해보기

2023년 01월 12일

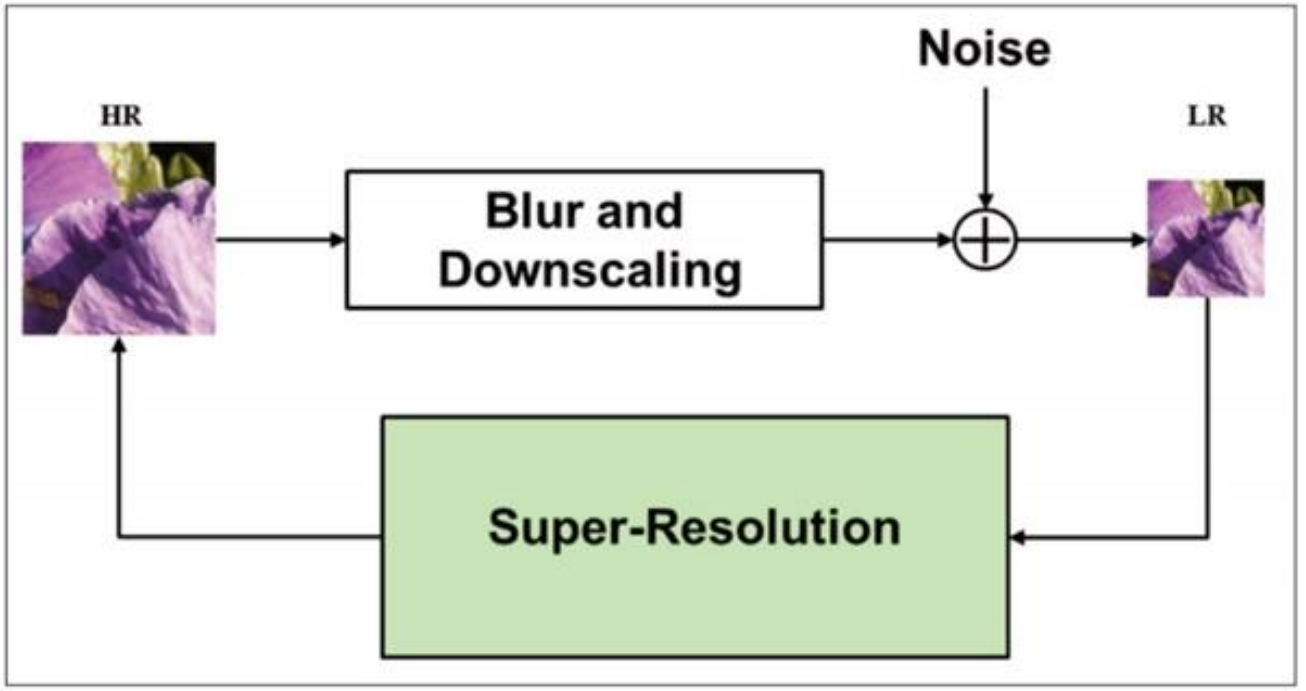
안재관

금일 목표

1. Git clone 및 가상환경 세팅
2. Pretrained model 받기
3. Inference 무작정 따라해보기

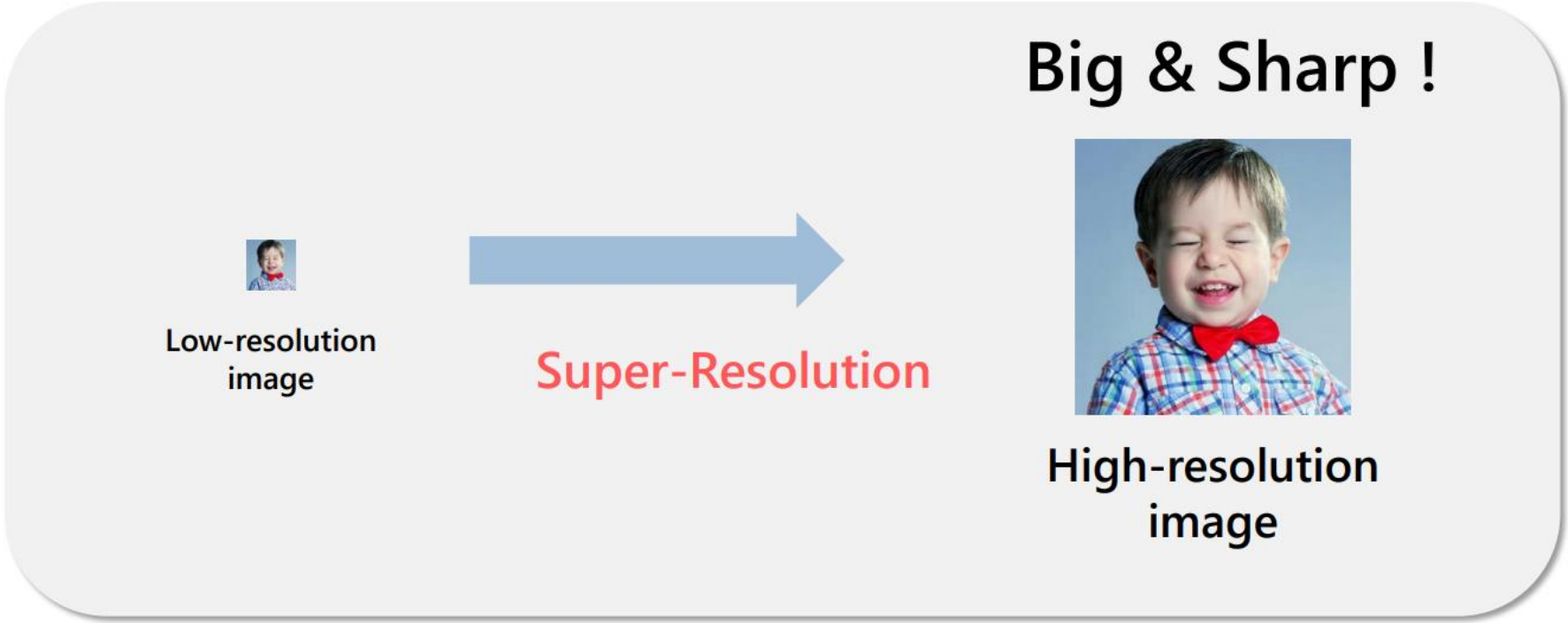
Super Resolution

저해상도 영상을 고해상도 영상으로 변환하는 작업을 의미



Super Resolution

저해상도 영상을 고해상도 영상으로 변환하는 작업을 의미



Super Resolution:BSRGAN

Designing a Practical Degradation Model for Deep Blind Image Super Resolution

Designing a Practical Degradation Model for Deep Blind Image Super-Resolution

Kai Zhang[1] Jingyun Liang[1] Luc Van Gool[1,2] Radu Timofte[1]
[1]Computer Vision Lab, ETH Zurich, Switzerland [2]KU Leuven, Belgium
{kai.zhang, jinliang, vangool, timofter}@vision.ee.ethz.ch
https://github.com/cszn/BSRGAN

Abstract

It is widely acknowledged that single image super-resolution (SISR) methods would not perform well if the assumed degradation model deviates from those in real images. Although several degradation models take additional factors into consideration, such as blur, they are still not effective enough to cover the diverse degradations of real images. To address this issue, this paper proposes to design a more complex but practical degradation model that consists of randomly shuffled blur, downsampling and noise degradations. Specifically, the blur is approximated by two convolutions with isotropic and anisotropic Gaussian kernels; the downsampling is randomly chosen from nearest, bilinear and bicubic interpolations; the noise is synthesized by adding Gaussian noise with different noise levels, adopting JPEG compression with different quality factors, and generating processed camera sensor noise via reverse-forward camera image signal processing (ISP) pipeline model and RAW image noise model. To verify the effectiveness of the new degradation model, we have trained a deep blind ESRGAN super-resolver and then applied it to super-resolve both synthetic and real images with diverse degradations. The experimental results demonstrate that the new degradation model can help to significantly improve the practicability of deep super-resolvers, thus providing a powerful alternative solution for real SISR applications.

1. Introduction

Single image super-resolution (SISR), which aims to reconstruct the natural and sharp detailed high-resolution (HR) counterpart $\mathbf{x}$ from a low-resolution (LR) image $\mathbf{y}$ [10, 47], has recently drawn significant attention due to its high practical value. With the advance of deep neural networks (DNNs), there is a dramatic upsurge of using feed-forward DNNs for fast and effective SISR [17, 23, 25, 27, 49, 61]. This paper contributes to this strand.

Whereas SISR methods map an LR image onto an HR counterpart, degradation models define how to map an HR image to an LR one. Two representative degradation models are bicubic degradation [46] and traditional degradation [28, 45]. The former generates an LR image via bicubic interpolation. The latter can be mathematically modeled by

$$\mathbf{y} = (\mathbf{x} \otimes \mathbf{k})\downarrow_s + \mathbf{n}. \quad (1)$$

It assumes the LR image is obtained by first convolving the HR image with a Gaussian kernel (or point spread function) $\mathbf{k}$ [12] to get a blurry image $\mathbf{x} \otimes \mathbf{k}$, followed by a downsampling operation $\downarrow_s$ with scale factor s and an addition of white Gaussian noise $\mathbf{n}$ with standard deviation σ. Specifically, the bicubic degradation can be viewed as a special case of traditional degradation as it can be approximated by setting a proper kernel with zero noise [3, 52]. The degradation model is generally characterized by several factors such as blur kernel and noise level. Depending on whether these factors are known beforehand or not, DNNs-based SISR methods can be broadly divided into non-blind methods and blind ones.

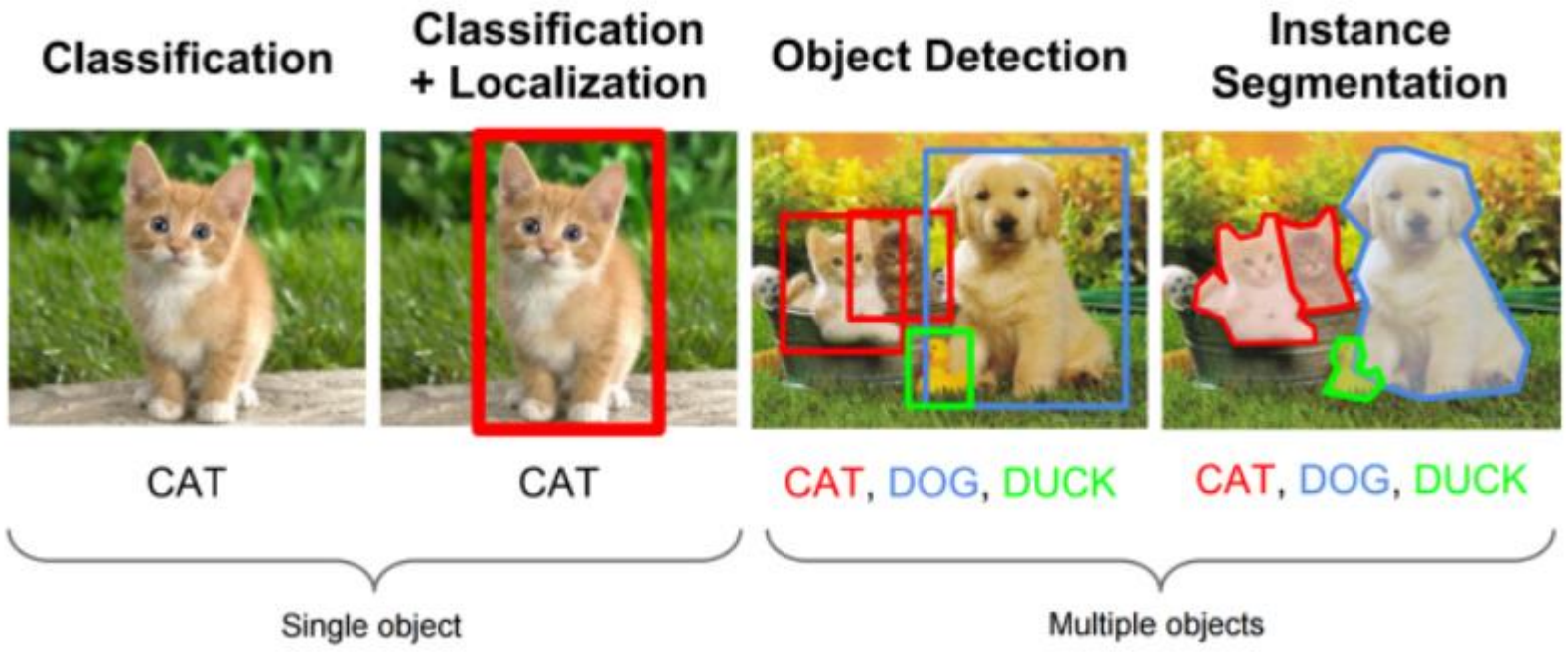
Early non-blind SISR methods were mainly designed for bicubic degradations [10]. Although significant improvements on the PSNR [27, 61] and perceptual quality [24, 49] have been achieved, such methods usually do not perform well on real images. It is worth noting that this also holds for deep models trained with a generative adversarial loss. The reason is that blur kernels play a vital role for the success of SISR methods [12] and a bicubic kernel is too simple. To remedy this, some works use a more complex degradation model which involves a blur kernel and additive white Gaussian noise (AWGN) and a non-blind network that takes the blur kernel and noise level as conditional inputs [3, 58]. Compared to methods based on bicubic degradation, these tend to be more applicable. Yet, they need an accurate estimation of the kernel and the noise level. Otherwise the performance deteriorates seriously [12]. Meanwhile, only a few methods are specially designed for the kernel estimation of SISR [3]. As a further step, some

Some visual examples: oldphoto2; butterfly; comic; oldphoto3; oldphoto6; comic_01; comic_03; comic_04

- Testing code

이미지 처리(Image Processing)이란?

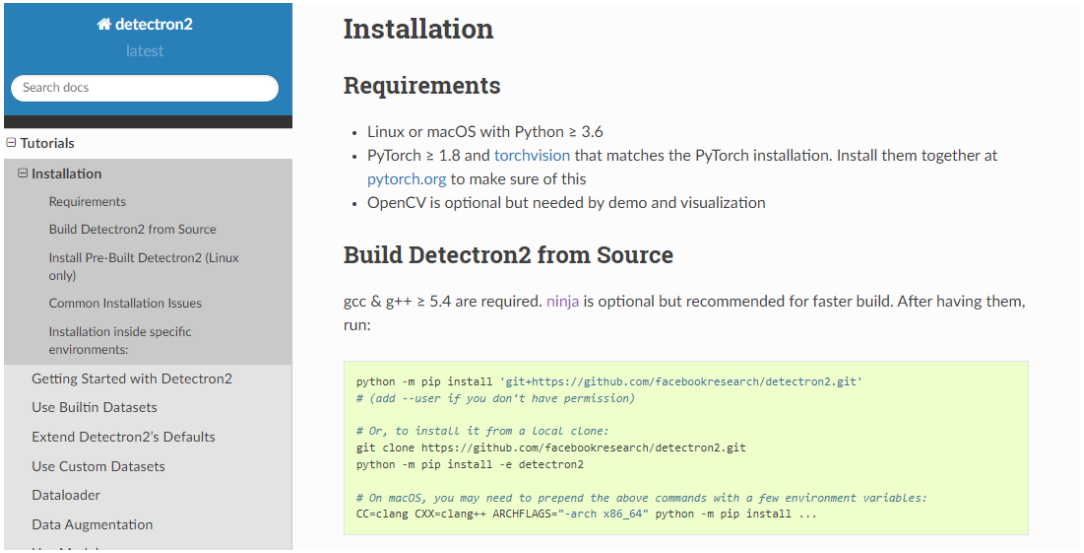
Neural Network를 이용한 이미지 처리에는 다음과 같이 분류

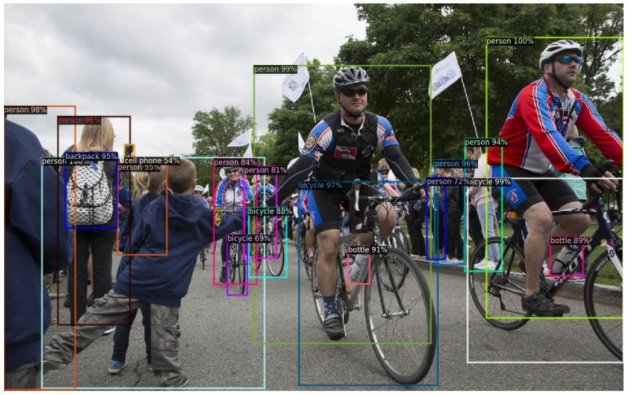


Processing:Detectron2

[facebookresearch/detectron2: Detectron2 is a platform for object detection, segmentation and other visual recognition tasks. (github.com)]

- FAIR(Facebook Artificial Intelligence Research)에서 만든 Pytorch기반 object detection과 semantic segmentation training/inference 플랫폼





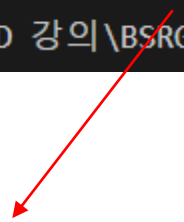
VS Code에서 아래 링크를 git clone “github link”.git

[kairess/BSRGAN: 4배 고해상도 복원 BSRGAN (옛날 사진, 인물 사진, 옛날 만화책) (github.com)]

```
(base) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의>git clone https://github.com/kairess/BSRGAN.git
```

```
(base) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의>cd BSRGAN

(base) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의\BSRGAN>
```



“리눅스 코드 cd는 폴도 이동하기 기능"

가상환경 및 Dependency 문제 해결을 위해 아래와 같이 설치한다

Python 설치

파이썬 가상환경(virtualenv) 구성

- **가상환경(virtualenv)은 여러 개의 파이썬 프로젝트가 하나의 컴퓨터에서 충돌을 일으키지 않고 존재할 수 있도록 하는 과정.**
- **각 프로그램별로 완전히 독립적인 가상의 환경을 만들어서 각 프로그램별로 패키지/라이브러리/모듈 등의 버전을 별도로 지정**
- **Windows나 Ubuntu 환경 모두 사용 가능**
  - 여기서는 windows 10 기준으로 설명
- **Python IDE 설치 후 anaconda prompt 혹은 terminal 검색 및 실행**
  - 명령어(https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html 참조)를 이용하여 가상환경 구성

```
conda update conda 입력하여 conda 업데이트 우선 진행
conda create –n [환경이름] python=3.7 실행하여 가상환경 만들기
conda activate [환경이름] 입력하여 가상환경 실행
```

* **Jupyter notebook에서는 추가 작업 필요:**

```
conda install jupyter notebook
python -m ipykernel install --user --name [환경이름] --display-name "[디스플레이 이름]"
```

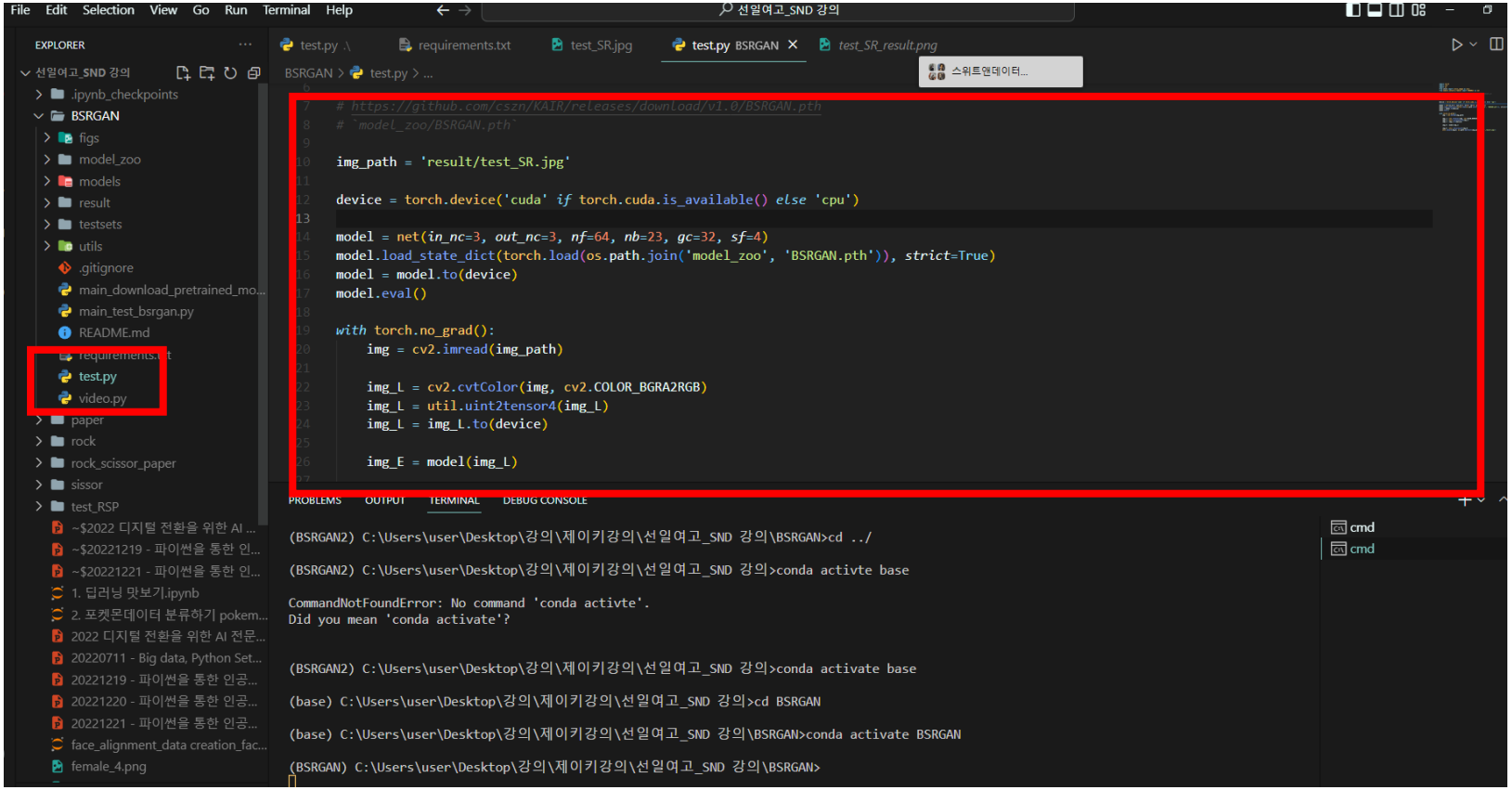
- Conda create -n BSRGAN python=3.8
- Conda activate BSRGAN
- Pip install -r requirements.txt

혹은

- Conda env create -file environment.yml

```
(BSRGAN) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의\BSRGAN>pip install -r requirements.txt
```

Test.py 코드 확인

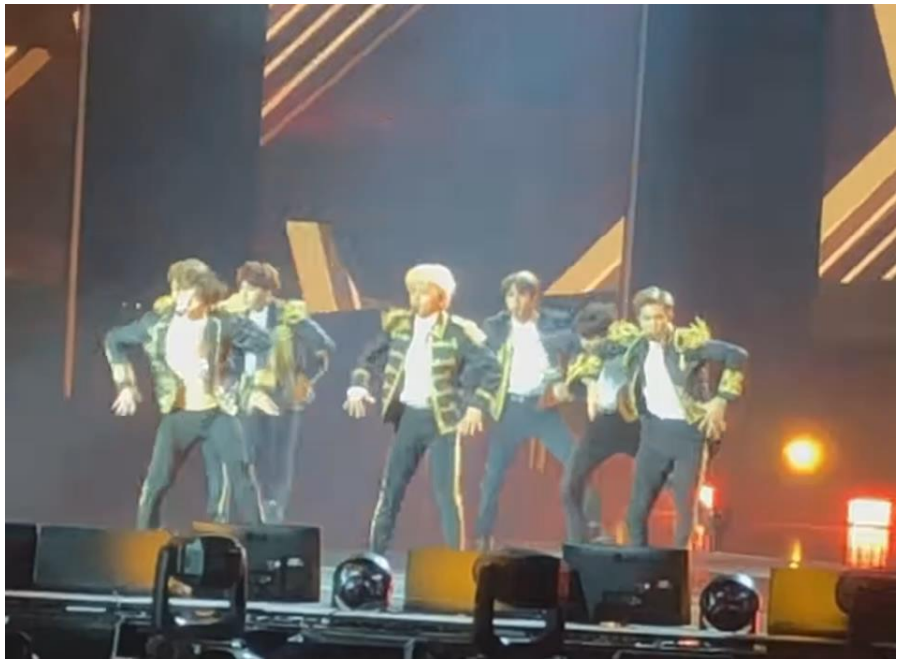

File Edit Selection View Go Run Terminal Help
선일여고_SND 강의
EXPLORER
test.py
requirements.txt
test_SR.jpg
test.py BSRGAN
test_SR_result.png
BSRGAN > test.py > ...
스위트앤데이터...
# https://github.com/cszn/KAIR/releases/download/v1.0/BSRGAN.pth
# `model_zoo/BSRGAN.pth`
img_path = 'result/test_SR.jpg'
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = net(in_nc=3, out_nc=3, nf=64, nb=23, gc=32, sf=4)
model.load_state_dict(torch.load(os.path.join('model_zoo', 'BSRGAN.pth')), strict=True)
model = model.to(device)
model.eval()
with torch.no_grad():
img = cv2.imread(img_path)
img_L = cv2.cvtColor(img, cv2.COLOR_BGRA2RGB)
img_L = util.uint2tensor4(img_L)
img_L = img_L.to(device)
img_E = model(img_L)
.ipynb_checkpoints
BSRGAN
figs
model_zoo
models
result
testsets
utils
.gitignore
main_download_pretrained_mo...
main_test_bsrgan.py
README.md
test.py
video.py
paper
rock
rock_scissor_paper
sissor
test_RSP
~$2022 디지털 전환을 위한 AI ...
~$20221219 - 파이썬을 통한 인...
~$20221221 - 파이썬을 통한 인...
1. 딥러닝 맛보기.ipynb
2. 포켓몬데이터 분류하기 pokem...
2022 디지털 전환을 위한 AI 전문...
20220711 - Big data, Python Set...
20221219 - 파이썬을 통한 인공...
20221220 - 파이썬을 통한 인공...
20221221 - 파이썬을 통한 인공...
face_alignment_data creation_fac...
female_4.png
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
cmd
cmd
(BSRGAN2) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의\BSRGAN>cd ../
(BSRGAN2) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의>conda activte base
CommandNotFoundError: No command 'conda activte'.
Did you mean 'conda activate'?
(BSRGAN2) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의>conda activate base
(base) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의>cd BSRGAN
(base) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의\BSRGAN>conda activate BSRGAN
(BSRGAN) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의\BSRGAN>

Python test.py 로 SR진행

```
(BSRGAN2) C:\Users\user\Desktop\강의\제이키강의\선일여고_SND 강의\BSRGAN>python test.py
[3, 3, 64, 23, 32, 4]
```

Python video.py 로 video SR진행

Python video.py 로 video SR진행



Python video.py 로 video SR진행

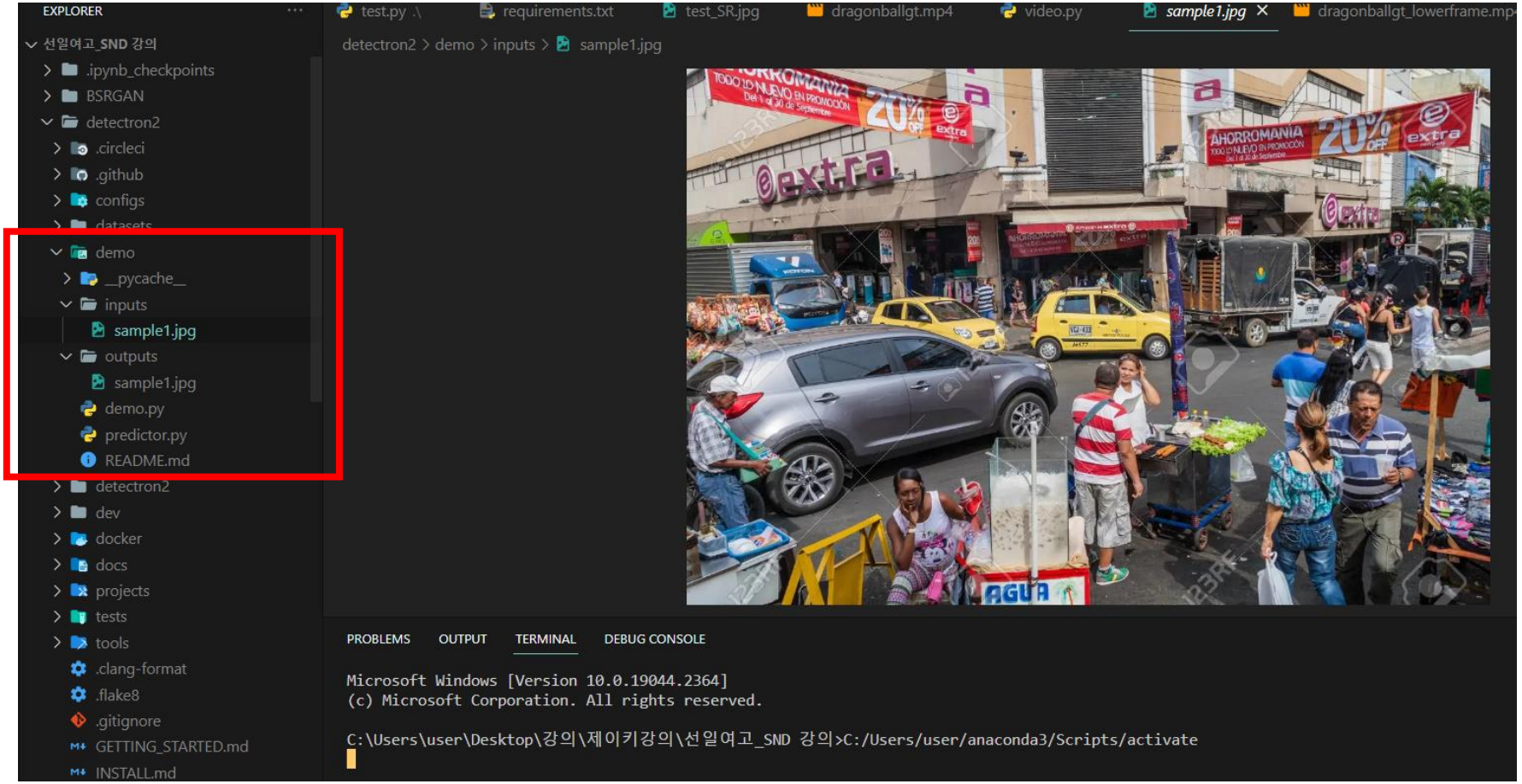
가상환경 및 Dependency 문제 해결을 위해 아래와 같이 설치한다

- conda create -n dtr python=3.8
- conda activate dtr
- Conda install git

- pip install torch==1.8.1 torchvision==0.9.1
- pip install opencv-python
- python -m pip install ninja
- pip install git+https://github.com/facebookresearch/detectron2

```
(pdf) jakey@ubuntu_88:~/detectron/detectron2/demo$ pip list
Package                 Version           Location
----------------------- ----------------- ------------------------------------------------------------
absl-py                 1.0.0
antlr4-python3-runtime  4.8
appdirs                 1.4.4
argon2-cffi             21.3.0
argon2-cffi-bindings    21.2.0
async-generator         1.10
attrs                   21.4.0
backcall                0.2.0
black                   21.4b2
bleach                  4.1.0
cachetools              4.2.4
certifi                 2021.5.30
cffi                    1.15.0
charset-normalizer      2.0.12
click                   8.0.4
cloudpickle             2.0.0
cycler                  0.11.0
Cython                  0.29.28
dataclasses             0.8
decorator               5.1.1
defusedxml              0.7.1
detectron2              0.6+cu101
easydict                1.9
entrypoints             0.4
faster-rcnn             0.0.0             /home/jakey/pdf_extraction/pdf_2/faster-rcnn.pytorch/lib
```

Detectron2/demo 폴더에 inputs 폴더 outputs 폴더를 생성한다.



가상환경 및 Dependency 문제 해결을 위해 아래와 같이 설치한다

Cd demo
python demo.py --config-file ../configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml --input inputs/sample1.jpg --outp
opts MODEL.DEVICE cpu MODEL.WEIGHTS detectron2://COCO-
InstanceSegmentation/mask_rcnn_R_50_FPN_3x/137849600/model_final_f10217.pkl

가상환경 및 Dependency 문제 해결을 위해 아래와 같이 설치한다

Cd demo
python demo.py --config-file ../configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml --video-input inputs/gangnam.m
outputs --opts MODEL.DEVICE cpu MODEL.WEIGHTS detectron2://COCO-
InstanceSegmentation/mask_rcnn_R_50_FPN_3x/137849600/model_final_f10217.pkl