

1. AI 모델 구축/운영 및 테스트를 위한 환경 구축 내역

<표 1> 환경 상세

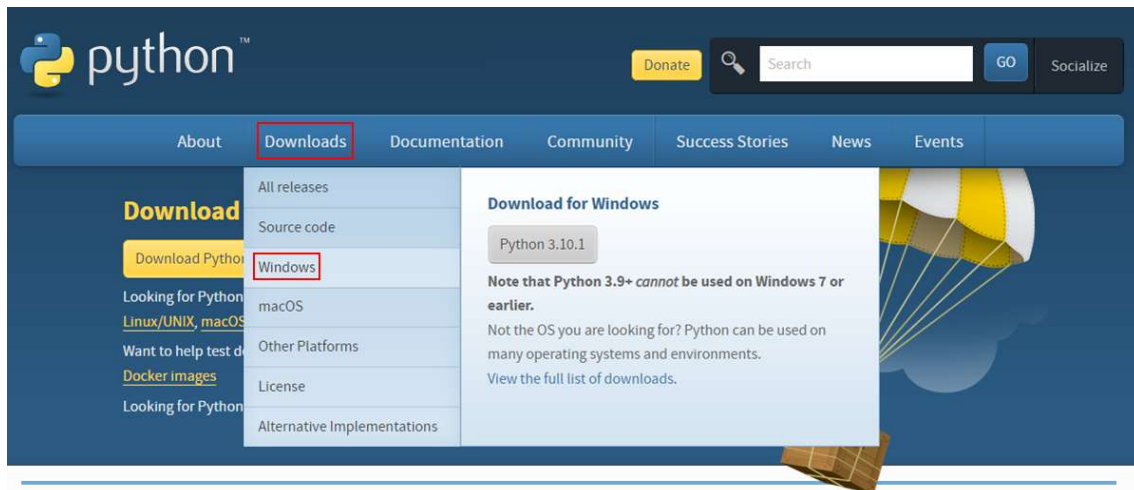
OS	Windows 10 64bit ※Pro, Home, Education, Enterprise 버전에서 정상 작동함.
프로그램 언어	Python 3.8
AI 프로젝트 실행을 위해 필요한 패키지 목록	Keras-Preprocessing==1.1.2
	Markdown==3.3.6
	Pillow==8.4.0
	Pygments==2.2.0
	Werkzeug==2.0.2
	absl-py==1.0.0
	astunparse==1.6.3
	blessings==1.7
	cachetools==4.2.4
	certifi==2021.10.8
	charset-normalizer==2.0.8
	cycler==0.11.0
	flatbuffers==2.0
	fonttools==4.28.2
	gast==0.3.3
	google-auth==2.3.3
	google-auth-oauthlib==0.4.6
	google-pasta==0.2.0
	gpustat==0.6.0
	grpcio==1.42.0
	h5py==2.10.0
	idna==3.3
	importlib-metadata==4.8.2
	kiwisolver==1.3.2
	libclang==12.0.0
	matplotlib==3.5.0
	metrics==0.3.3
	numpy==1.18.5
	oauthlib==3.1.1
	opt-einsum==3.3.0
	packaging==21.3
	pandas==1.3.4
	pathlib2==2.3.6
	pathspec==0.5.5

	pip==20.0.2 pkg-resources==0.0.0 protobuf==3.19.1 psutil==5.8.0 pyasn1==0.4.8 pyasn1-modules==0.2.8 pyparsing==3.0.6 python-dateutil==2.8.2 pytz==2021.3 requests==2.26.0 requests-oauthlib==1.3.0 rsa==4.8 scikit-learn==1.0.1 scipy==1.4.1 setuptools==44.0.0 setuptools-scm==6.3.2 six==1.16.0 sklearn==0.0 tensorboard==2.7.0 tensorboard-data-server==0.6.1 tensorboard-plugin-wit==1.8.0 tensorflow==2.7.0 tensorflow-estimator==2.7.0 tensorflow-gpu==2.7.0 tensorflow-io-gcs-filesystem==0.22.0 termcolor==1.1.0 threadpoolctl==3.0.0 tomli==1.2.2 typing-extensions==4.0.1 urllib3==1.26.7 wheel==0.37.0 wrapt==1.13.3 zipp==3.6.0
--	---

2. AI 모델 소스코드 활용을 위한 환경 구축 메뉴얼

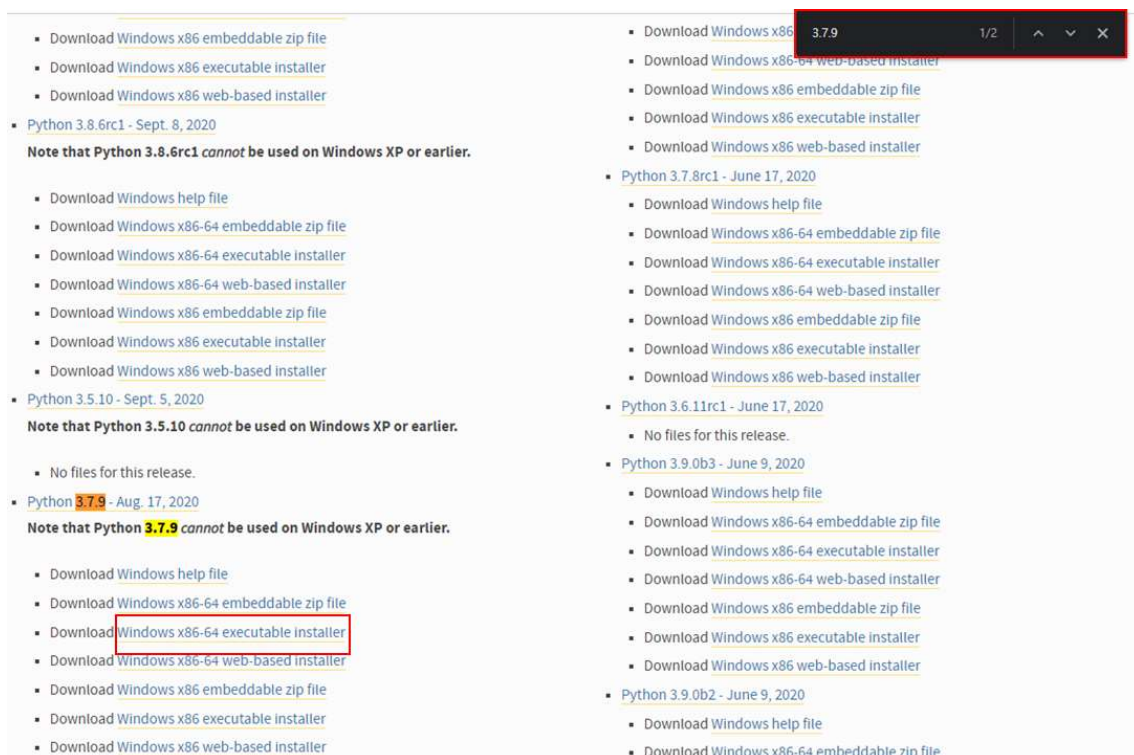
1. Python 설치 메뉴얼

- 1) <https://www.python.org/> 접속
- 2) 아래의 그림과 같이 상단의 Downloads → Windows 클릭



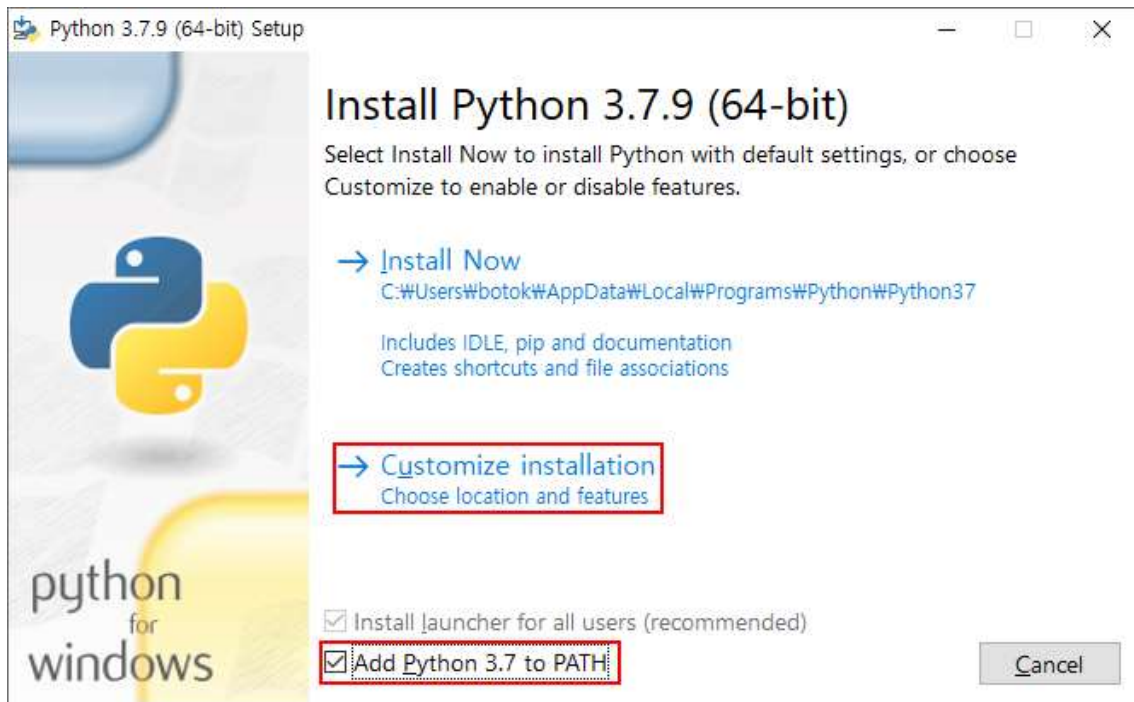
<그림 1> Python 홈페이지

3) 브라우저의 검색기능(ctrl + f)을 활용하여 3.7.9를 검색한 후 “Windows x86-64 executable installer”를 클릭하여 python 3.7.9 64bit 버전 설치 파일을 다운로드



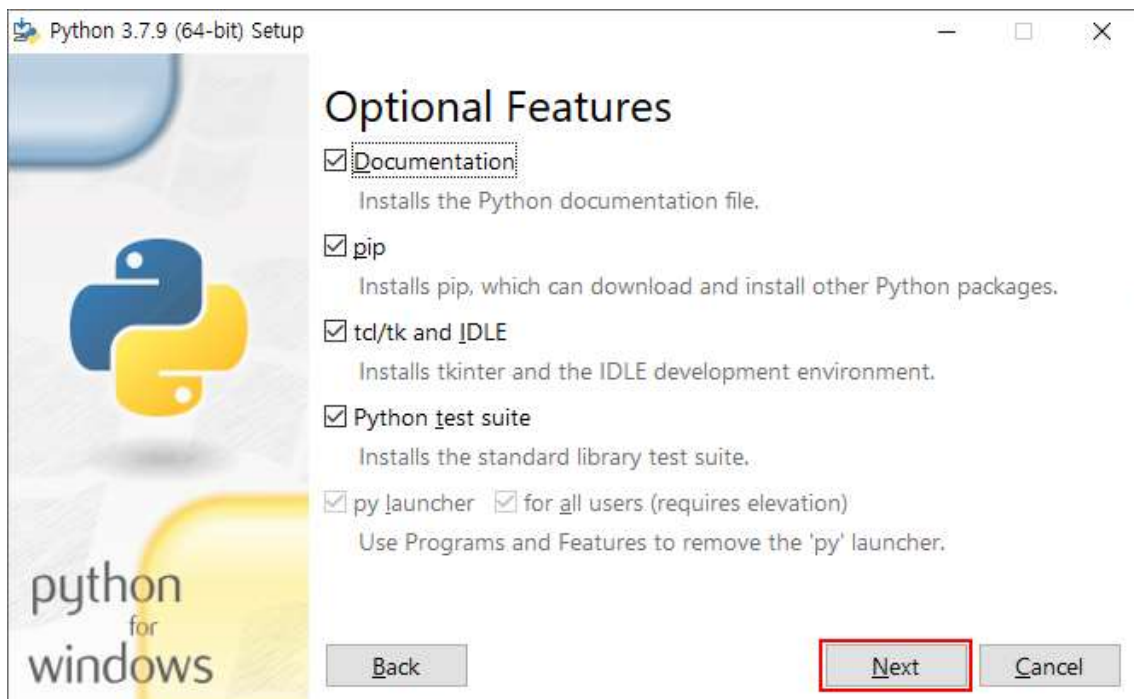
<그림 2> Python 3.7.9 download

4) 다운로드된 파일을 실행하고, “Add Python 3.7.9 to PATH” 박스를 클릭하여 체크한 후 “Customize installation” 클릭



<그림 3> Python 설치 화면 1

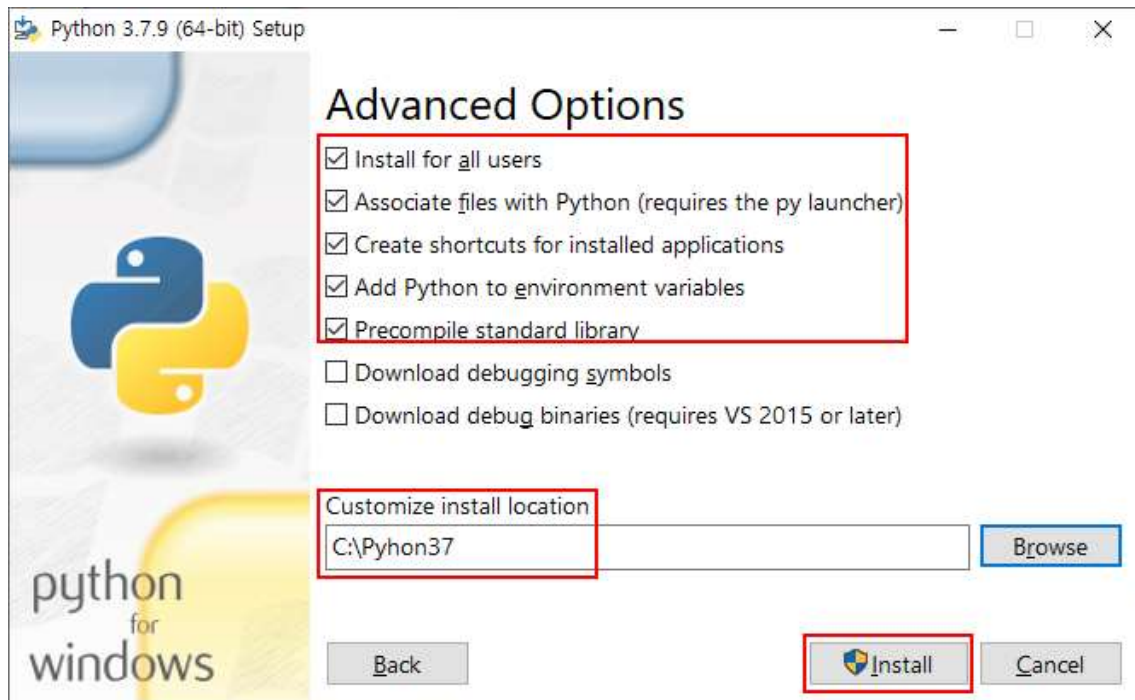
5) 모든 항목을 체크한 후 "Next" 버튼 클릭



<그림 4> Python 설치 화면 2

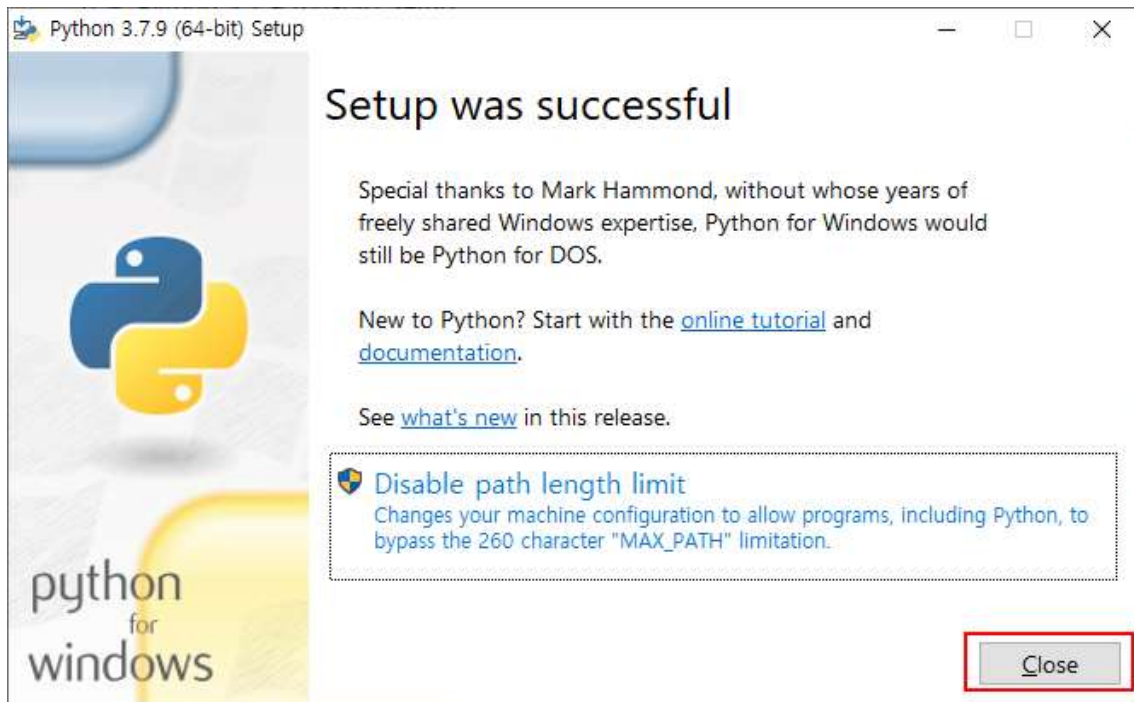
6) "Install for all users", "Associate files with Python (requires the py launcher)",

"Create shortcuts for installed applications", "Add Python to environment variables", "Precompile standard library" 선택 후 "Customize install location"에 "C:/Python37"을 기입한 후 Install 버튼을 클릭하여 설치 진행



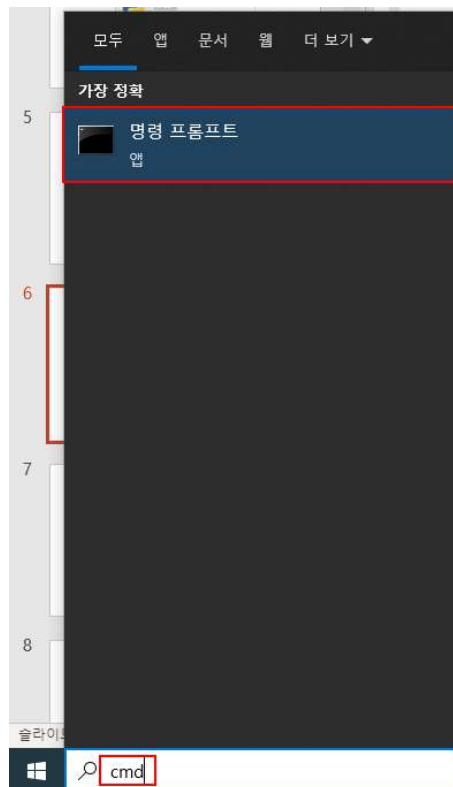
<그림 5> Python 설치 화면 3

7) Close 버튼을 클릭하여 설치 종료



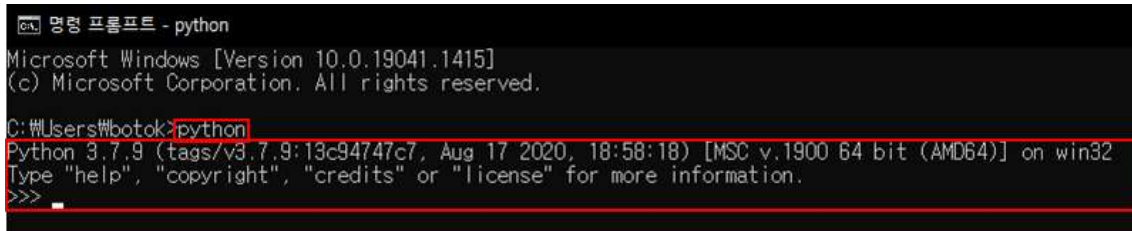
<그림 6> Python 설치 화면 4

8) Python 설치 확인을 위해 윈도우 검색창에서 cmd를 검색한 뒤 실행



<그림 7> 명령 프롬프트 실행

- 9) 명령 프롬프트 창에서 'python' 입력 후 엔터. 아래 그림과 같은 메시지가 출력되면 설치 완료



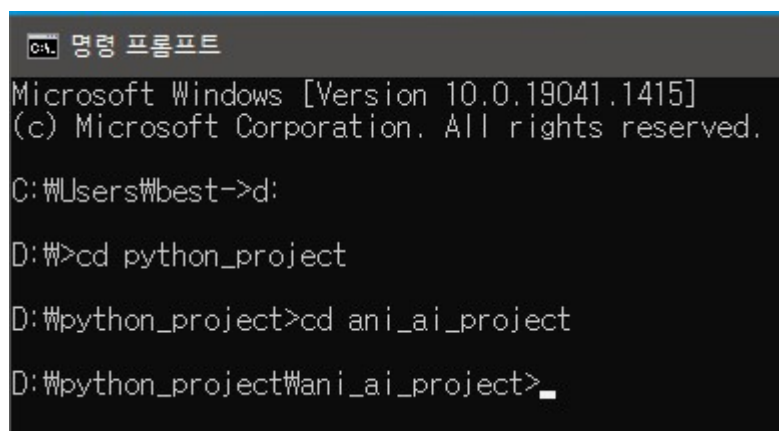
```
C:\> 명령 프롬프트 - python
Microsoft Windows [Version 10.0.19041.1415]
(c) Microsoft Corporation. All rights reserved.

C:\> python
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

<그림 8> Python 설치 확인

2. 프로젝트 환경 구축 매뉴얼

- 1) AI HUB에서 제공하는 반려동물 안구 데이터를 다운로드 하여 프로젝트 폴더에 저장
- 2) 명령 프롬프트 창에서 아래와 같이 압축 해제된 경로로 이동

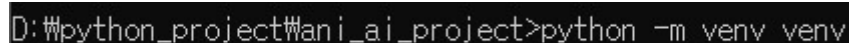


```
C:\> 명령 프롬프트
Microsoft Windows [Version 10.0.19041.1415]
(c) Microsoft Corporation. All rights reserved.

C:\> d:
D:\> cd python_project
D:\python_project> cd ani_ai_project
D:\python_project\ani_ai_project>
```

<그림 9> 프로젝트 경로로 이동

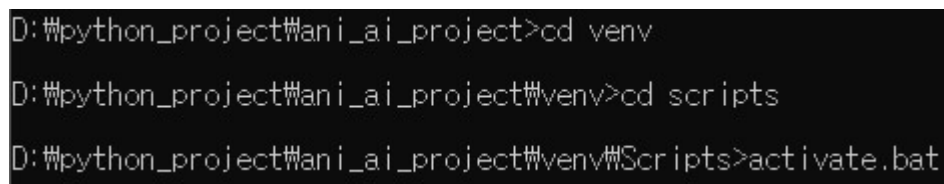
- 3) 프로젝트 경로에서 'python -m venv venv' 입력 후 엔터



```
D:\python_project\ani_ai_project> python -m venv venv
```

<그림 10> Python 가상환경 생성


- 4) "{프로젝트 경로}/venv/Scripts"로 이동 → 'activate.bat' 입력 후 엔터



```
D:\python_project\ani_ai_project> cd venv
D:\python_project\ani_ai_project\venv> cd scripts
D:\python_project\ani_ai_project\venv\Scripts> activate.bat
```

<그림 11> Python 가상환경 실행

5) 아래 그림과 같이 프로젝트 경로로 다시 이동



```
(venv) D:\python_project\wani_ai_project\venv\Scripts>cd..  
(venv) D:\python_project\wani_ai_project\venv>cd..  
(venv) D:\python_project\wani_ai_project>
```

<그림 12> 프로젝트 경로로 이동

6) 아래 그림과 같이 'pip install -r requirements.txt'를 입력하여 프로젝트에 필요한 필수 패키지 설치



```
(venv) D:\python_project\wani_ai_project>pip install -r requirements.txt_
```

<그림 13> 프로젝트에 필요한 환경 설치

3. AI 모델 소스코드 실행 매뉴얼

1. 모델 학습

- 1) 명령 프롬프트 실행
- 2) 프로젝트 폴더 경로로 이동 / (사용자가 지정한 프로젝트 폴더로 이동)
- 3) "{프로젝트 경로}/venv/Scripts"로 이동 → 'activate.bat' 입력 후 엔터

```
D:\python_project\Wani_ai_project>cd venv
D:\python_project\Wani_ai_project\venv>cd scripts
D:\python_project\Wani_ai_project\venv\Scripts>activate.bat
```

<그림 14> Python 가상환경 실행

- 4) 아래 그림과 같이 프로젝트 경로로 다시 이동

```
C:\> 명령 프롬프트
(venv) D:\python_project\Wani_ai_project\venv\Scripts>cd ..
(venv) D:\python_project\Wani_ai_project\venv>cd ..
(venv) D:\python_project\Wani_ai_project>
```

<그림 15> 프로젝트 경로로 이동

- 5) 아래와 같이 'python efficientNet_ai_train.py' 과 'python ResNet_v1_50_ai_train.py' 입력 후 엔터

```
(venv) D:\python_project\Wani_ai_project>python efficientNet_ai_train.py
```

<그림 16> 프로젝트 학습

```
(venv) D:\python_project\Wani_ai_project>python ResNet_v1_50_ai_train.py
```

- 6) 참고사항

- 1) batch 사이즈 지정: --batch_size {정수}
 - (1) 예시: python train.py --batch_size 8
 - (2) 기본 값은 32로 지정됨
- 2) epoch 지정: --epochs {정수}
 - (1) 예시: python train.py -- epochs 100

(2) 기본 값은 10으로 지정됨

2. 사용자가 학습한 모델 테스트

- 1) 명령 프롬프트 실행
- 2) 프로젝트 폴더 경로로 이동
- 3) "{프로젝트 경로}/venv/Scripts"로 이동 → 'activate.bat' 입력 후 엔터

```
D:\python_project\Wani_ai_project>cd venv
D:\python_project\Wani_ai_project\venv>cd scripts
D:\python_project\Wani_ai_project\venv\Scripts>activate.bat
```

<그림 18> Python 가상환경 실행

- 4) 아래 그림과 같이 프로젝트 경로로 다시 이동

```
명령 프롬프트
(venv) D:\python_project\Wani_ai_project\venv\Scripts>cd ..
(venv) D:\python_project\Wani_ai_project\venv>cd ..
(venv) D:\python_project\Wani_ai_project>
```

<그림 19> 프로젝트 경로로 이동

- 5) 아래 그림과 같이 'python efficientNet_ai_test.py'와 'python ResNet_v1_50_ai_test' 입력 후 엔터

```
(venv) D:\python_project\Wani_ai_project>python efficientNet_ai_test.py
```

<그림 20> efficientNet 학습 테스트 경로

```
(venv) D:\python_project\Wani_ai_project>python ResNet_v1_50_ai_test.py
```

<그림 21> ResNet 학습 테스트 경로

4. AI 모델 구축/운영 및 테스트를 위한 환경 구축 내역

<표 2> Ubuntu 환경 상세

OS	Ubuntu 20.04.3 LTS
프로그램 언어	Python 3.8
AI 프로젝트 실행을 위해 필요한 패키지 목록	Keras-Preprocessing==1.1.2
	Markdown==3.3.6
	Pillow==8.4.0
	Pygments==2.2.0
	Werkzeug==2.0.2
	absl-py==1.0.0
	astunparse==1.6.3
	blessings==1.7
	cachetools==4.2.4
	certifi==2021.10.8
	charset-normalizer==2.0.8
	cycler==0.11.0
	flatbuffers==2.0
	fonttools==4.28.2
	gast==0.3.3
	google-auth==2.3.3
	google-auth-oauthlib==0.4.6
	google-pasta==0.2.0
	gpustat==0.6.0
	grpcio==1.42.0
	h5py==2.10.0
	idna==3.3
	importlib-metadata==4.8.2
	kiwisolver==1.3.2
	libclang==12.0.0
	matplotlib==3.5.0
	metrics==0.3.3
	numpy==1.18.5
	nvidia-ml-py3==7.352.0
	oauthlib==3.1.1
	opt-einsum==3.3.0
	packaging==21.3
	pandas==1.3.4
	pathlib2==2.3.6
	pathspec==0.5.5
	pip==20.0.2

	pkg-resources==0.0.0
	protobuf==3.19.1
	psutil==5.8.0
	pyasn1==0.4.8
	pyasn1-modules==0.2.8
	pyparsing==3.0.6
	python-dateutil==2.8.2
	pytz==2021.3
	requests==2.26.0
	requests-oauthlib==1.3.0
	rsa==4.8
	scikit-learn==1.0.1
	scipy==1.4.1
	setuptools==44.0.0
	setuptools-scm==6.3.2
	six==1.16.0
	sklearn==0.0
	tensorboard==2.7.0
	tensorboard-data-server==0.6.1
	tensorboard-plugin-wit==1.8.0
	tensorflow==2.7.0
	tensorflow-estimator==2.7.0
	tensorflow-gpu==2.7.0
	tensorflow-io-gcs-filesystem==0.22.0
	termcolor==1.1.0
	threadpoolctl==3.0.0
	tomli==1.2.2
	typing-extensions==4.0.1
	urllib3==1.26.7
	wheel==0.37.0
	wrapt==1.13.3
	zipp==3.6.0

Ubuntu 20.04 LTS 기준으로 python이 기본적으로 설치되어 나옵니다.

terminal창에 #python3 --version 이라고 명령어를 입력하면 python의 버전이 출력됩니다.

python이 기본적으로 설치가 되어 있지 않거나 다른 버전의 python이 설치되어 있다면 python 3.8 버전으로 설치를 진행합니다.

apt-get을 update 해주고 python을 설치합니다.

```
$ sudo apt update  
$ sudo apt -y install software-properties-common
```

<그림 22> 터미널에서 패키지 목록을 업데이트

```
$ sudo add-apt-repository -y ppa:deadsnakes/ppa
```

<그림 23> deadsnakes PPA추가

```
$ sudo apt -y install python3.8
```

<그림 24> 명령어를 입력하여 python 3.8 버전 설치

```
$ python3.8 --version
```

<그림 25> Python 3.8 버전 설치 완료