

강원대학교
AI 소프트웨어학과

데이터 전처리

- 이미지 데이터 수집하기 -

import cv2 → cv2: OpenCV 라이브러리로 이미지 처리 및 비디오 처리에 사용됨

import os → os: 운영 체제와 상호 작용하기 위한 라이브러리로 파일과 디렉터리를 관리하는 데 사용됨

os.makedirs('폴더명', exist_ok=True)

해당 경로에 새로운 이름의 폴더 생성

file_list = os.listdir()
print(file_list)

현재 경로에 존재하는 파일 리스트

os.rmdir('폴더명')

현재 경로에 존재하는 해당 파일 삭제

이미지 데이터를 수집하는 방법

```
cwd = os.getcwd()  
print(cwd)
```

현재 내가 작업하는 환경의 경로

```
os.chdir("원하는 경로")  
cwd = os.getcwd()  
print(cwd)
```

바꾸고 싶은 작업 환경의 경로

```
os.path.exists("파일명 또는 경로 명")
```

파일 또는 경로가 존재하는지 확인하는 방법

이미지 데이터를 수집하는 방법

`import cv2` → `cv2`: OpenCV 라이브러리로 이미지 처리 및 비디오 처리에 사용됨

`import os` → `os`: 운영 체제와 상호 작용하기 위한 라이브러리로 파일과 디렉터리를 관리하는 데 사용됨

```
if not os.path.exists('captures'):  
    os.makedirs('captures')
```

'captures'라는 이름의 폴더가 존재하지 않으면, 이를 생성함 → 이미지 파일들이 이 폴더에 저장됨

```
cap = cv2.VideoCapture("경로")
```

동영상을 사용하기 위해 사용하는 동영상의 경로과 파일명을 지정해줌

```
while True:  
    ret, frame = cap.read()
```

무한 루프를 통해 카메라로부터 이미지(프레임)를 지속적으로 읽어옴

```
    if not ret:  
        print("동영상이 끝났습니다.")  
        break
```

동영상이 끝났을 때를 알기 위해

프레임의 처리 내용

```
cv2.imshow('Video', display_image)
```

처리하고 싶은 프레임의 내용과 이를 보여주기

```
if cv2.waitKey(1) & 0xFF == ord('1'):  
    실행하고 싶은 작업내용
```

1을 누리면 아래의 작업을 실행함 → 실행하고 싶은 작업내용
Q를 누리면 아래의 작업을 실행함(break) → 종료

```
if cv2.waitKey(1) & 0xFF == ord('q'):  
    break
```

```
cap.release()  
cv2.destroyAllWindows()
```

카메라 장치를 해제함
생성된 모든 윈도우를 닫음

01 이미지 데이터

이미지 데이터 수집 방법 - 웹캠

```
cap = cv2.VideoCapture(0)
```

웹캠을 사용하기 위한 객체를 생성함 0은 시스템의 첫 번째 카메라를 의미함

```
while True:
```

```
    ret, frame = cap.read()
```

무한 루프를 통해 카메라로부터 이미지(프레임)를 지속적으로 읽어옴

```
    if not ret:
```

```
        print("웹캠이 끝났습니다.")
```

```
        break
```

웹캠이 끝났을 때를 알기 위해

프레임의 처리 내용

```
cv2.imshow('Webcam', display_image)
```

처리하고 싶은 프레임의 내용과 이를 보여주기

```
if cv2.waitKey(1) & 0xFF == ord('1'):
```

실행하고 싶은 작업내용

1을 누리면 아래의 작업을 실행함 → 실행하고 싶은 작업내용

Q를 누르면 아래의 작업을 실행함(break) → 종료

```
if cv2.waitKey(1) & 0xFF == ord('q'):
```

```
    break
```

```
cap.release()
```

카메라 및 비디오 입력을 해제함

```
cv2.destroyAllWindows()
```

생성된 모든 윈도우를 닫음

01 이미지 데이터

이미지 데이터 수집 방법 - 동영상/웹캠 인코딩 및 저장

```
cap = cv2.VideoCapture("동영상 파일 명")
```

동영상 파일을 읽어 옴

```
if not cap.isOpened():  
    print("Error: Could not open video.")  
    exit()
```

비디오가 정상적으로 열렸는지 확인

```
is_video_capturing = False
```

```
fourcc = cv2.VideoWriter_fourcc(*'mp4v') → 인코딩
```

```
output_file_name = 'output.avi' → 저장 파일명
```

```
fps = cap.get(cv2.CAP_PROP_FPS) → 동영상의 프레임 속도
```

```
frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)) →프레임의 너비
```

```
frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)) →프레임의 높이
```

```
cap = cv2.VideoWriter(output_file_name, fourcc, fps, (frame_width, frame_height))
```

```
is_video_capturing = True
```

```
cap.release()
```

```
out.release()      비디오 저장을 하기 위해
```

사용 가상환경 저장

사용 가상환경 저장

```
pip freeze > requirements.txt
```

사용했던 패키지 저장

```
pip install -r requirements.txt
```

사용했던 패키지 다른 가상환경에서 실행