

# AI Newsletter (No. 7)

연구관리처 AI연구센터 / (2024년 2월)

K-water연구원 AI연구센터에서 정기적으로 발간하는 뉴스레터입니다.  
AI 뉴스, Hands-on 프로젝트, 팁 등을 다양한 내용과 난이도로 담았습니다.

※ 코드 및 뉴스 등 외부링크가 다수 포함되어 있으므로 인터넷 환경 PC에서 사용 권장

## >> 7호 주요 콘텐츠

#AI연구센터

#Hello, K-water

#직접 돌려보기

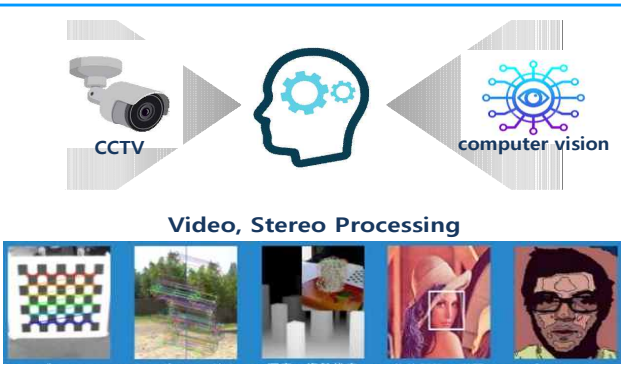
#AI 3학년 1반



[AI News] 구글의 생성형 AI 제미니 출시!



[AI News] 사람의 생각을 읽는 AI가 있다고?



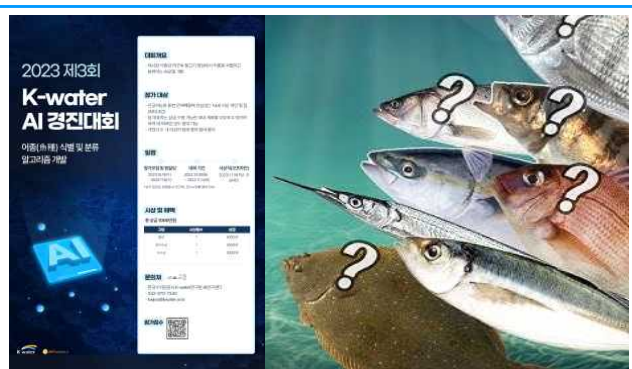
[Hands-on] 실시간 영상(CCTV) 활용



[Tips] 메모리 낭비, 반복문 없이 코딩하기



[Tips] 파이썬 가상환경 만들기 2탄



[AI Lab.] 제3회 K-water AI 경진대회 성료

모든 실습 예제와 링크는 “AI연구센터” 누리집 (Homepage, GitHub)에 게시되어 활용가능합니다.

# 1. What's NEW in AI

#Weather Forecast

#Gemini

#Korean SAT evaluation

#BCI

## □ 구글의 생성모델 제미니(Gemini), OpenAI의 ChatGPT를 뛰어넘을까?

- 구글에서는 멀티모달 생성모델 제미니를 12월 6일에 발표하였습니다. 인간 전문가를 능가한다는 평가받고 있는 최초의 AI모델 제미니는 이미지, 비디오, 오디오 다중 모드에서는 GPT-4V와 Whisper를 능가했다고 합니다.
- 제미니는 모델의 크기에 따라서 Ultra(복잡한 작업을 위한 가장 크고 뛰어난 모델), Pro(광범위한 작업에 걸쳐 확장 가능), Nano(온디바이스 작업을 위한 가장 효율적인 모델) 세가지로 최적활용이 가능하며,
- 특히, Ultra 모델은 언어모델 벤치마크 32개 항목 중 30개 항목에서 GPT-4V를 능가했고, 특히 수학, 역사, 의학 등 다양한 분야의 정보를 이해하고 문제를 해결하는 태스크인 MMLU에서 처음으로 사람의 점수를 뛰어넘었다고 합니다.

※ MMLU: Massive Multitask Language Understanding, 인공지능 모델이 획득한 지식을 측정하는 벤치마크  
Gemini 소개 글: <https://blog.google/technology/ai/google-gemini-ai/#introducing-gemini>  
Gemini 소개 영상: <https://youtu.be/jV1vkHv4zq8>

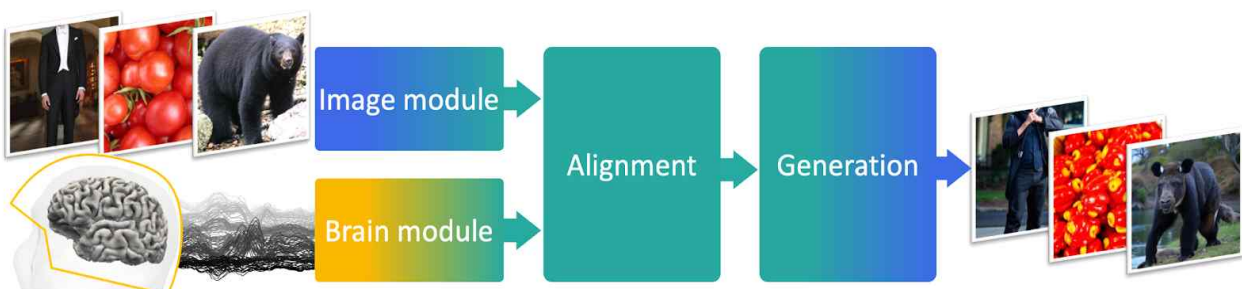
## □ 사람의 뇌파를 통해 생각을 읽는 AI가 개발되고 있다고?!!

- 사람의 뇌파를 통해 생각을 읽는 기술을 BCI(Brain Computer Interface) 기술이며, 예를 들면 사람에게 사진을 보여줬을 때 발생하는 뇌신호를 측정해서 이미지를 다시 생성할 수 있는 기술입니다.
- 금년 10월에 Meta에서는 그동안의 fMRI\*(기능적 자기 공명 영상)의 낮은 해상도를 생성형 AI모델을 이용하여 극복함으로써 뇌의 신호를 좀 더 잘 측정하고 학습할 수 있는 기술을 개발하고 논문\*\*을 발표하였습니다.

\* fMRI: functional magnetic resonance imaging

\*\* Brain Decoding: Toward Real-Time Reconstruction of Visual Perception

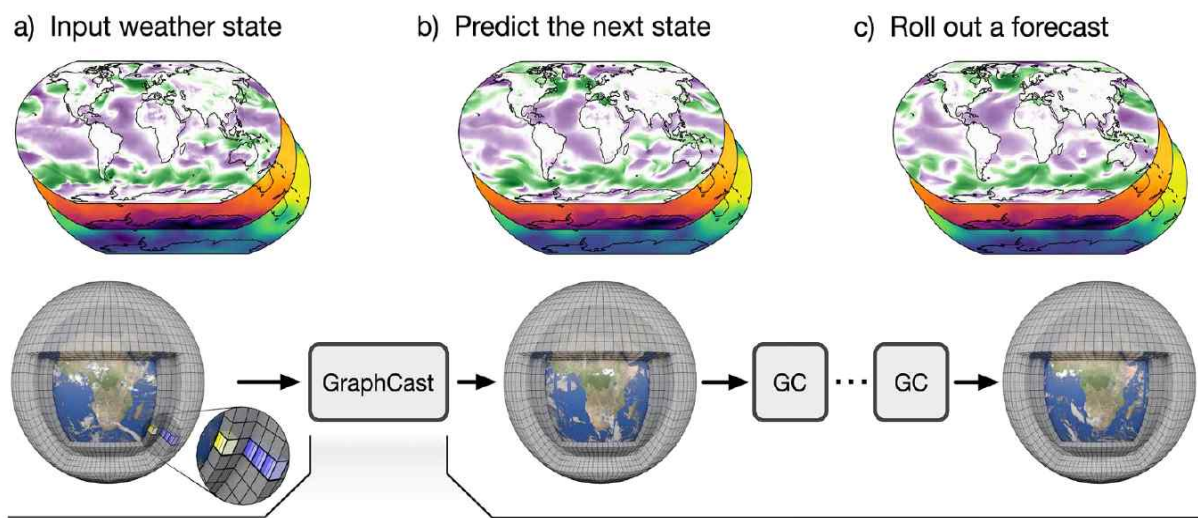
- 이러한 기술은 몸을 제어하기 어려운 장애인들과 의사소통을 하거나 생각만으로 운전을 한다든지 인간의 생활에 많은 도움을 줄 수 있는 기술로 기대가 되지만 뇌 해킹 등의 부작용에 대한 우려도 제기되고 있습니다.



## □ 1분안에 10일 기상예측 “구글 딥마인드의 GraphCast”

- 구글 딥마인드에서 올해 11월 사이언스지를 통해 기상예측 AI모델 GraphCast\*를 발표했습니다. 그동안의 기상예측은 기상전문가들의 전문적인 지식과 슈퍼컴퓨터를 이용한 엄청난 양의 물리적 과정에 대한 계산을 통해 예측이 이루어졌습니다.
- 이번에 발표한 구글 딥마인드의 GraphCast는 지구 표면 전체를 위·경도 0.25도의 영역으로 나누고, GNN(Graph Neural Network) 기술을 이용하여 각 영역의 기온, 풍속, 기압 등 총 227개의 변수로 분석하였습니다. GNN은 10일 동안의 날씨를 예측하는데 1분 이하의 시간이 걸릴 정도로 빠른 계산이 가능하며, 정확도 측면에서도 뛰어난 성능을 보인다고 합니다.

※ Learning skillful medium-range global weather forecasting (Science, November 01, 2023)



## □ GPT-4의 수능 1등급 도전 프로젝트

- 본 프로젝트는 수능 국어과목에 프롬프트 엔지니어링 없이 GPT-3.5와 GPT-4를 적용시 각각 8등급과 3등급이라는 예상 밖의 낮은 성능을 확인하고, 성능을 개선할 수 있는 방법을 찾기 위해서 시작된 프로젝트입니다.
- 이 프로젝트는 GPT모델을 시간, 돈, 많은 데이터를 필요로 하는 Fine-Tuning 방법보다는 패턴 학습과 Zero-Shot 성능이 뛰어난 GPT-4의 강점을 활용하여 국어 수능문제를 어떻게 푸는지 알려주는 형태로 성능을 개선하였습니다.
- 객관식 문제를 푸는데 적합하도록 지문을 이해하고 선택지 중에 정답을 고르는 방법에 대해서 CoT\*기반의 간단한 프롬프트를 사용한 결과 GPT-4의 수능 국어 성적이 2등급(상위 5%, 94점)까지 상승했고, 특정 문제 유형에 최적화된 프롬프트를 추가로 사용하면 1등급(상위 4%, 96점)까지도 올랐다고 합니다.

※ CoT (Chain of Thought): 복잡한 문제를 풀기 위해 단계별로 생각하면서 차례대로 접근하는 방법

- 관련 GitHub: [https://github.com/NomaDamas/KICE\\_slayer\\_AI\\_Korean](https://github.com/NomaDamas/KICE_slayer_AI_Korean)



## 2. Meanwhile, in K-water AI Lab.

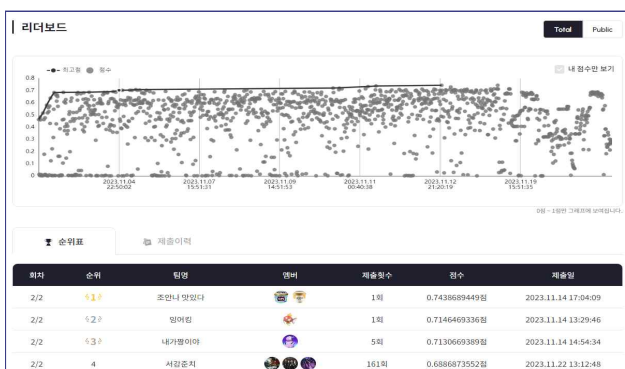
# 제3회 AI경진대회

# 자연어처리 세미나

# AI홍수 심포지엄

### □ (홍보) 제3회 「K-water 대국민 AI 경진대회」 성료

- 물고기 종류를 식별하고 분류하는 알고리즘을 개발하는 AI 경진대회가 금년에 성황리에 종료되었습니다. 3주 동안 전 세계에서 400명이 넘는 인원이 참가하였으며, 종료 후에도 연습 리더보드가 열리는 등 그 어느 때보다 많은 관심을 받았습니다.
- 2주간의 치열한 경쟁 끝에 세종대학교 팀이 1등을 차지했으며, 2위와 3위는 롯데정보통신 팀이 차지했습니다. 수상자들이 출품한 code는 추후 고도화 작업을 거쳐 낙동강 하굿둑 “실시간 어도 이용 어류 인식 및 분석 시스템”에 활용될 예정입니다.
- 내년에도 새롭고 흥미로운 주제로 돌아올 테니 많은 참여 부탁드립니다! :)



## 2023 제3회 K-water AI 경진대회

어종(魚種) 식별 및 분류 알고리즘 개발

#### 대회개요

~ 제시된 낙동강 하굿둑 물고기 영상에서 어종을 식별하고 분류하는 AI모델 개발

#### 참가 대상

- 인공지능을 통한 문제해결에 관심있는 14세 이상 개인 및 팀 (최대 4인)
- 팀 대표자는 상급 수경 가능한 국내 거주를 보유하고 있어야 하며 내외국인 모두 참여 가능
- 개인이 두 개 이상의 팀에 중복 참여 불가

#### 일정

참가모집 및 팀빌딩	대회 기간	시상식(오프라인)
2023.10.18(수) ~ 2023.11.8(수)	2023.10.30(월) ~ 2023.11.13(월)	2023년 11월 하순 경

\*상기 일정은 변동될 수 있으며, 공지사항을 통해 안내

#### 시상 및 혜택

총 상금 1000만원

구분	시상인수	상금
대상	1	500만원
최우수상	1	300만원
우수상	1	200만원

#### 문의처

한국수자원공사 K-water연구원 AI연구센터  
 ~ 042-870-7340  
 ~ kwjoo@kwwater.or.kr

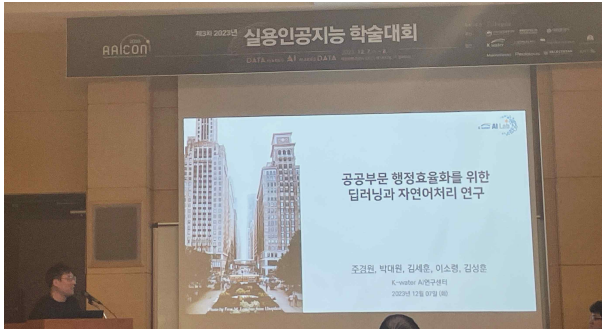
#### 참가접수



## □ (학회) 제3회 실용인공지능학회(AAiCON) 참석 (12월 7일~8일)

- AI연구센터에서는 AI 기술 발전과 관련된 응용 연구 및 개발 구축 사례 등 실용적인 연구를 공유하는 실용인공지능학회에 참석하여 자체 수행한 연구 결과 2건을 발표하였습니다.

- \* 데이터 모델과 수치모델 하이브리드 방식의 취수원 수질예측 시뮬레이션 (김성훈 등)
- \* 공공부문 행정 효율화를 위한 딥러닝과 자연어처리 활용 사례 연구 (주경원 등)

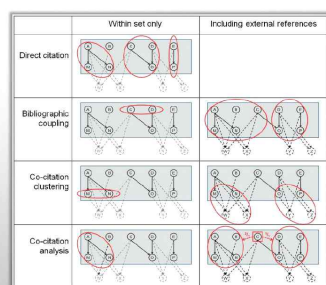


## □ (행사) 「자연어처리 및 미래기술 예측」 전문가 세미나 개최

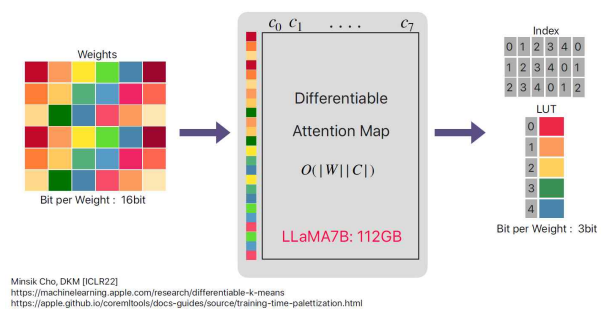
- AI연구센터에서는 12월 5일(월) 자연어처리 및 거대언어모델 기반 기술정보 분석 분야의 최신연구 이해를 위한 전문가 초청 세미나를 진행했습니다.
- 첫 번째 발표로는 KISTI(한국과학기술정보연구원)의 미래기술분석센터에서 매년 미래기술 시그널 탐지 연구를 수행하고 계신 양혜영 박사님이 발표가 있었으며, 두 번째 발표로는 거대언어모델 효율적 추론 분야에서 활발히 연구하고 계신 Apple AI/ML 부문의 조민식 박사님의 발표가 진행되었습니다.



### 미래기술 조기 탐지



### Differentiable K-Means Weight Clustering (DKM) [ICLR22]





## □ (행사) 「인공지능 홍수분석 기술 현황 및 미래」 심포지엄 개최

- AI연구센터에서는 한국수자원학회 AI응용연구분과와 공동으로 「인공지능 홍수 분석 기술 현황 및 미래」라는 주제로 심포지엄을 개최(12/19)하였습니다. 추운 날씨에도 불구하고 50여 명이 참석하는 등 성황리에 진행되었습니다.
- 정부에서는 신속한 의사결정이 요구되는 홍수관리 업무에 AI 기술을 도입하기 위한 연구개발을 수행 중으로 내년까지 빠르게 완료할 계획입니다.
- 따라서 이번 심포지엄에서 K-water와 수자원학회 전문가들뿐 아니라 환경부 AI홍수 담당자도 참석하여 관련 기술개발 동향을 공유하고, 예상되는 문제와 해결방안에 대해서도 심도 있게 논의하였습니다.
- 주제발표는 산·학·연의 전문가 네 명이 댐 및 하천 홍수, 산사태 등 다양한 물 재해에 AI를 적용한 사례를 발표하는 것으로 진행되었습니다.

- |                                      |                           |
|--------------------------------------|---------------------------|
| - AI 기반 재해분석 적용사례 소개 /               | 이기하 교수(경북대학교 건설방재공학부)     |
| - 하천범람, 도시침수에서 인공지능 활용과 가상센서 구성 방안 / | 장준석 대표(㈜팀제파)              |
| - AI를 활용한 실시간 댐 홍수분석기술 개발 /          | 최영돈 박사(K-water연구원 AI연구센터) |
| - 영상분석 기술을 활용한 하천범람 예측 방법론 /         | 변성준 박사(국제도시물정보과학연구원)      |

- 발표에 이어 AI 홍수분석의 현황과 미래 발전방안에 대해 지정토론자 여섯 명과 함께 객석의 전문가들이 열띤 토론을 이어갔습니다.
- ※ 토론자 : 김성훈 AI연구센터장(K-water), 차준호 연구관(환경부), 이을래 소장(K-water), 강부식 교수(단국대), 권문혁 부장(K-water), 김연주 교수(연세대)
- 토론자들은 현 기술 수준에서 AI 실용화를 위해서는 물리모형의 설명력과 AI의 신속·정확한 장점을 융합하는 것이 바람직하며, 미래기술의 방향성을 고려할 때 AI를 적극적으로 활용하려는 노력이 필요하다고 강조했습니다.



### 3. “10줄 코딩” Hands-on AI Project

#OpenCV 사용법

#실시간 영상(CCTV) 활용

#Frame 저장

#### □ OpenCV를 활용한 실시간 영상(CCTV) 활용 (초급)

※ cv2.imshow() 함수는 기술적인 이유로 google colab에서 실행되지 않습니다.

관련내용: <https://github.com/jupyter/notebook/issues/3935>

- 컴퓨터 비전(CV)을 목적으로 개발된 오픈소스 라이브러리 OpenCV를 활용한 실시간 영상 프로세싱 방법에 대해 알아보겠습니다.

- OpenCV는 이미지나 동영상에서 물체검출, 추적, 특징추출, 패턴인식 등의 작업을 어렵지 않게 수행할 수 있습니다.

- 우선, 실습 작업을 위해 OpenCV 패키지를 설치하고, 라이브러리를 불러옵니다.

```
pip install opencv-python      # 패키지를 설치합니다.  
import cv2                    # 라이브러리를 불러옵니다.
```

- 웹캠 또는 CCTV 영상을 캡처하기 위한 객체를 생성합니다.

- 노트북 cam 또는 Webcam 사용시 환경 : argument = 0

- CCTV(실시간 스트리밍) 사용시 환경 : argument = RTSP\* 주소

- \* RTSP(Real Time Streaming Protocol) 스트리밍 미디어 서버를 컨트롤 하기 위해 고안된 네트워크 프로토콜

```
# 웹캠 영상 캡처 샘플(1)  
webcam = cv2.VideoCapture(0) # VideoCapture 객체를 생성합니다. (0 : webcam)  
  
# CCTV 영상 캡처 샘플(2)  
webcam = cv2.VideoCapture('rtsp://210.99.70.120:1935/live/cctv007.stream')  
  
# 천안시 교통정보 CCTV 공개 (https://www.data.go.kr/data/15063717/fileData.do) 천안역
```

- 본격적으로 영상과 연결하여 프레임(Frame) 단위로 표출합니다.

- isOpened() 메소드는 VideoCapture 객체가 정상적으로 연결되었다면 True, 그렇지 않다면 False를 반환합니다.

```
# 영상이 제대로 연결되었는지 확인합니다. 정상 연결 시 True를 반환하여 조건문을 통과시킵니다.  
if webcam.isOpened():  
    # <영상이 연결되어 있는 동안 프레임을 읽어서 영상을 표출하는 순환문 삽입>  
else:  
    print("Can't Open Video")
```

- 영상이 연결되어 있는 동안 read() 메소드를 통해 frame을 읽어(캡처)내고, cv2.imshow()\* 함수를 활용하여 캡처된 frame을 보여줍니다. 사용자가 키보드 'q'를 입력해서 반복문을 탈출하기 전까지 계속해서 보이게 됩니다.

- \* imshow(winname: 영상을 출력할 대상 창 이름, mat: 표출할 영상자료)

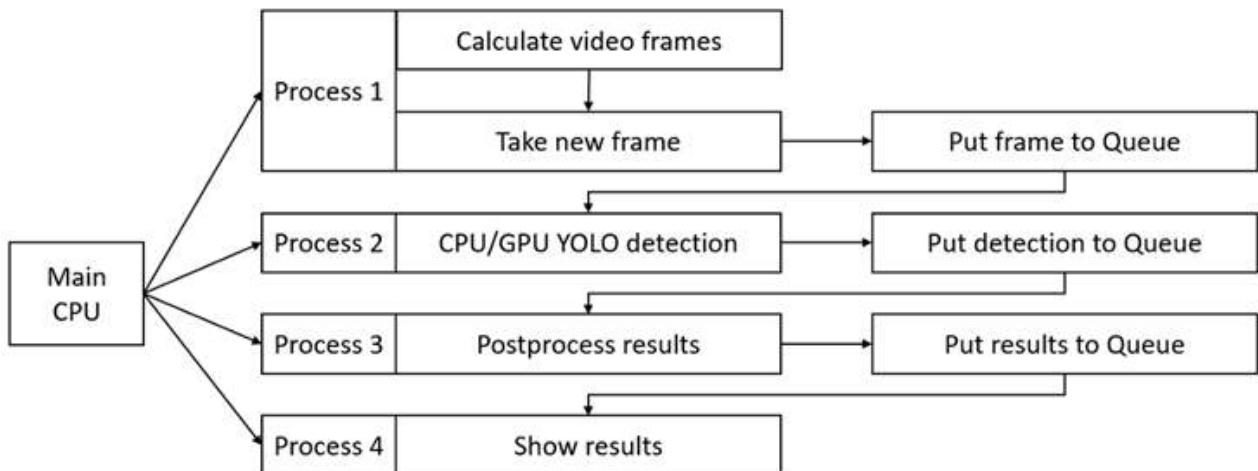
```

# 정상 연결 시 프레임을 읽어서 영상을 표출합니다.
while True:
    status, frame = webcam.read()    # read() 메소드로 영상자료(frame)을 캡처합니다.
    if status:
        cv2.imshow("test", frame) # 캡처된 영상자료를 cv2.imshow() 함수를 이용해 표출
                                   # 합니다. cv2.waitKey와 같이 써주어야 작동합니다.
        if cv2.waitKey(1) & 0xFF == ord('q'): # 'q'입력 시 반복문을 탈출합니다.
            break
    else:
        print('No Frame')
        break
else:
    print("Can't Open Video")

# 영상과 연결을 해제하고 생성했던 창을 닫습니다.
webcam.release()
cv2.destroyAllWindows()

```

- 추가 작업(Yolo 적용, 물체검출, 추적, 특징추출, 패턴인식 등) 적용 시에는,
  - 기본 작업은 Process1 → Process3 → Process4 과정으로 간단히 구현할 수 있습니다.
  - 추가 작업은 중급 이상의 단계로 기본 작업에서 추출된 Frame을 활용하여 Process2 과정을 추가하여 Yolo나 OpenCV 등의 작업을 적용하여 구현할 수 있습니다.





## 4. TIPS

# join함수      # 2차원 전치행렬      # 최빈값 구하기      # 가상환경 2탄

### □ (파이썬) 리스트로 메모리 낭비 없이 문자열 만들기

- '+'연산자를 사용하면 'a'+b'+c'처럼 문자열을 추가할 때마다 문자열 객체를 생성하기 때문에 메모리 관리 측면에서 효율적이지 않습니다.

```
a = ['Python', 'is', 'awesome'] # 리스트를 만듭니다.  
print(a[0]+' '+a[1]+' '+a[2]) # +연산자와 공백(' ')을 사용해 문자열을 합칩니다.
```

출력: Python is awesome

- 그러나, join함수를 사용하면 이러한 메모리의 과소비 없이 손쉽게 문자열을 합칠 수 있습니다.

```
a = ['Python', 'is', 'awesome'] # 리스트를 만듭니다.  
print(' '.join(a)) # 리스트 각 요소 사이를 띄우려면 공백(' ')을 이용합니다.
```

출력: Python is awesome

- 숫자형 포함 시 문자열 변환을 위해 list comprehension을 사용합니다.

```
a = [7, '+', 5, '=', 12] # 리스트를 만듭니다.  
print(' '.join(str(i) for i in a)) # list comprehension 사용해 차례로 문자형으로 변환합니다.
```

출력: 7 + 5 = 12

### □ (파이썬) 반복문 없이 리스트 내 요소들의 최빈값 구하기

- 리스트나 문자열 요소에 대한 개수를 구할 때 반복문을 사용할 수도 있지만, 파이썬 내장모듈인 collection의 Counter 함수를 사용하면 편리합니다.
- Counter 함수의 most\_common 메소드는 인자값으로 상위 몇 번째 요소의 개수를 구할지 정할 수 있고, 생략하면 모든 요소를 반환합니다.

```
a = [1, 2, 3, 1, 2, 3, 1, 1, 1, 1] # 리스트를 만듭니다.  
from collections import Counter # collections 모듈에서 Counter 함수를 import합니다.  
cnt = Counter(a) # Counter 함수에 list a를 적용합니다.  
print(cnt.most_common(1)) # most_common 메소드를 이용해 [(빈도값, 빈도수)]를  
print(cnt.most_common()) # 구합니다(인자값 1이면 최빈값, 2이면 최빈값과 2위 값).
```

출력: [(1, 6)]  
[(1, 6), (2, 2), (3, 2)] # 인자값을 생략하면 모든 요소를 반환합니다.

### □ (파이썬) 2차원 행렬의 전치행렬 구하기

- 전치행렬(轉置行列; transposed matrix)은 행렬을 역슬래시 방향으로 행과 열을 뒤바꾼 행렬입니다. 즉, 다음과 같이 주대각선을 축으로 반사 대칭을 가하여 얻는 행렬입니다.

$$A = (1, 2) \quad A^T = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad B = \begin{pmatrix} 1, 2 \\ 3, 4 \\ 5, 6 \end{pmatrix} \quad B^T = \begin{pmatrix} 1, 3, 5 \\ 2, 4, 6 \end{pmatrix}$$

- 이를 구하기 위하여, zip 함수를 사용하여 동일한 위치의 요소를 묶어내면, 전치행렬을 구하는 것과 같은 결과를 얻을 수 있습니다.
- 리스트나 튜플 앞에 '\*'를 붙이면 요소들을 하나씩 꺼집어낼 수 있습니다. 즉, 리스트의 포장을 풀어주는(unpacking) 효과입니다.

```
array2d = [[1,2], [3,4], [5,6]])          # 리스트로 2x3행렬을 만듭니다.
print(array2d)
```

출력: [[1, 2], [3, 4], [5, 6]]

```
array2d = [[1,2], [3,4], [5,6]])          # 리스트로 2x3행렬을 만듭니다.
# 바깥쪽 리스트를 풀고 zip함수로 안쪽 리스트를 인덱스 순서대로 묶습니다.
transposed = list(zip(*array2d))
print(transposed)
```

출력: [(1, 3, 5), (2, 4, 6)]

## □ (가상환경 시리즈) 2탄. 가상환경을 만들어보자

- 1탄에서는 가상환경이 무엇인지 그리고 왜 사용하는지 알아보았습니다. 복습하자면, 가상환경은 독립적인 실행환경이기 때문에 가상환경별로 버전과 패키지를 설치할 수 있어 프로젝트별 관리가 쉽다는 장점이 있었습니다.
- 이번 호에서는 가상환경 만드는 법을 알아보겠습니다.

### 1. 아나콘다 설치(무료)

[www.anaconda.com/download](http://www.anaconda.com/download)

### 2. 아나콘다 프롬프트 열기

윈도우키 → anaconda prompt

### 3. 가상환경 만들기

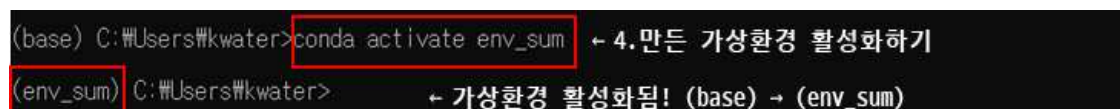


관리자: Anaconda Prompt

(base) C:\Users\kwater>conda create env\_sum python=3.9 ← 3.가상환경 만들기

conda create env\_sum python=3.9  
콘다라는 명령어로 // 만들겠다 // 가상환경이름은 env\_sum // 사용할 파이썬 버전=3.9

### 4. 가상환경 활성화하기



(base) C:\Users\kwater>conda activate env\_sum ← 4. 만든 가상환경 활성화하기

(env\_sum) C:\Users\kwater> ← 가상환경 활성화됨! (base) → (env\_sum)

### 5. 필요한 패키지 설치하기

pip install plotly 등등