

덕성여자대학교 & 해성여자고등학교

2025학년도 꿈이룸 창의융합인재 5주차

# 캡스톤 1 일차

## 디지털 영상을

## 다루는 OpenCV



하윤종 (hayunjong83@gmail.com)

# 컴퓨터가 디지털 이미지를 보는 방법



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0


## What Computer Sees

0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

컴퓨터는 **이미지**를 **숫자(number)**로 인식한다.

참고 [CNN Series Part 1 : How do computers see images?](#)

# = 비트와 바이트 =

비트 (binary digit, bit)	바이트 (Byte)
0과 1, 두 가지 값만 가질 수 있는 측정 단위	여덟 개의 비트로 구성된 데이터의 양을 나타내는 단위
 0 OFF FALSE  1 ON TRUE	

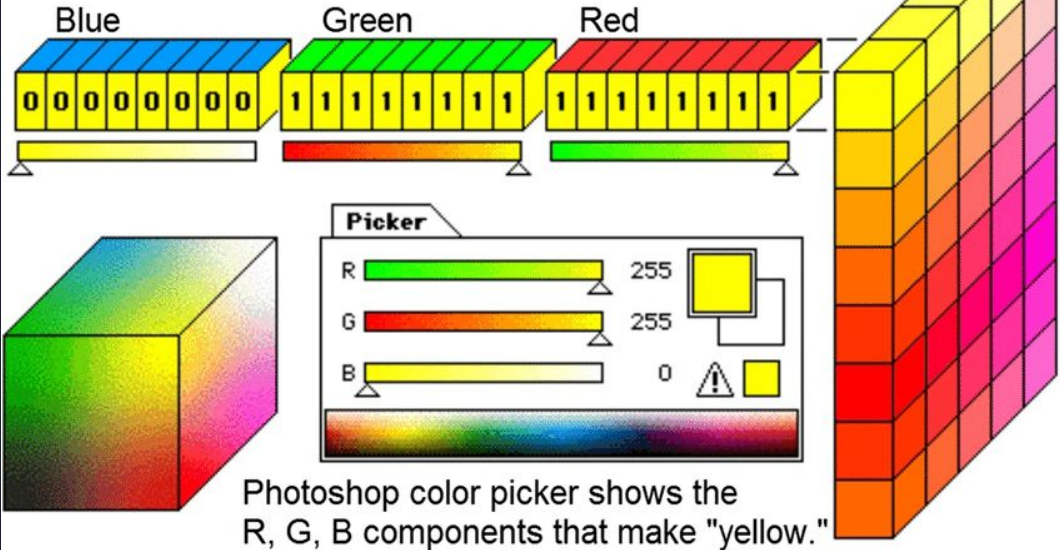
삼성반도체이야기  
samsungsemiconstory.com

신호를 나타내는 최소 단위 '비트(bit)'  
정보의 기본단위 '바이트(byte)'

참고) [삼성반도체 - 바이트\(Byte\)](#)

## 24-bit "true color" displays

Each screen pixel is represented by three groups of eight bits, for a total of 24 bits.

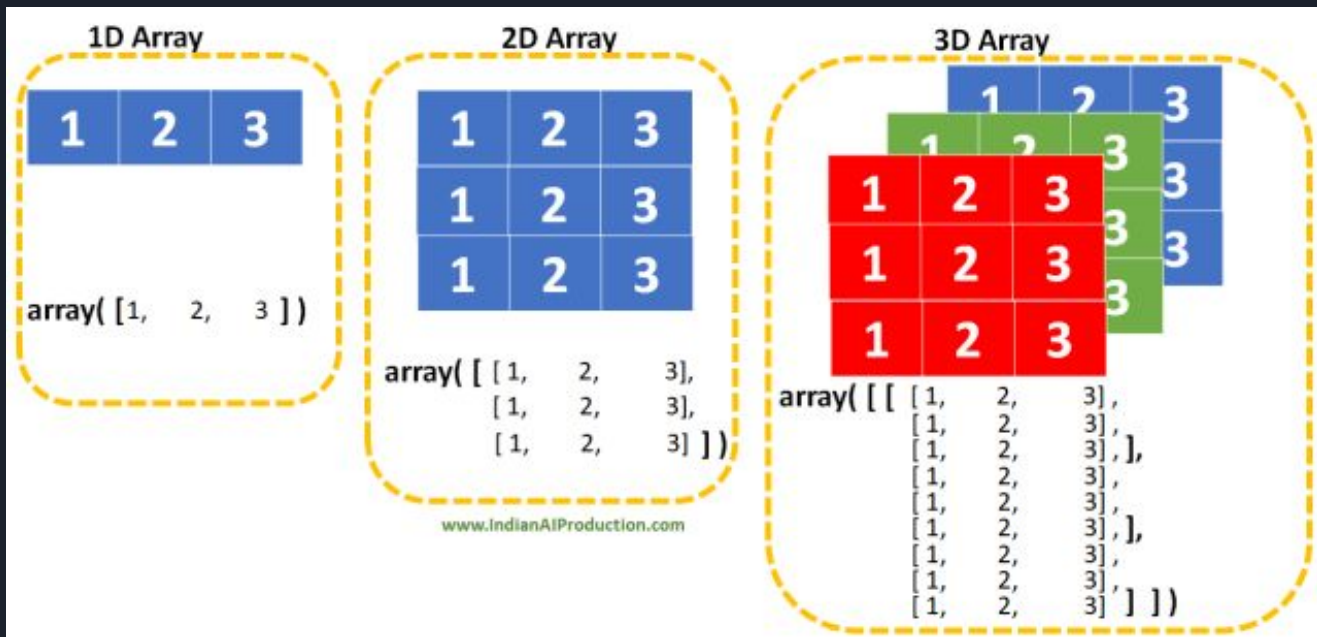


참고) [Digital Image Processing with MATLAB](#)

컬러 영상/이미지의 한 픽셀(pixel)은 0부터 255 사이의 R(red), G(green), B(blue) 세 색상의 조합으로 나타난다.

참고) [A new high capacity and secure image realization steganography based on ASCII code matching](#)

# 파이썬(python)에서 행렬을 쉽게 다루게 해주는 NumPy



참고) [What is the NumPy array?](#)

이미지를 숫자가 나열된 배열(행렬)로 생각한다면, 이미지를 NumPy를 이용하여 다룰 수 있다.

## NumPy로 풀어보는 행렬 문제

Anaconda PowerShell Prompt

```
(base) PS C:\Users\hayunjong83> conda info --envs
```

```
# conda environments:
```

```
#
```

```
base * C:\Users\hayunjong83\miniconda3
```

```
duksung C:\Users\hayunjong83\miniconda3\envs\duksung
```

```
haesung C:\Users\hayunjong83\miniconda3\envs\haesung
```

```
(base) PS C:\Users\hayunjong83> conda activate haesung 가상환경 활성화
```

```
(haesung) PS C:\Users\hayunjong83> pip install numpy 라이브러리 설치
```

```
Collecting numpy
```

```
Using cached numpy-2.3.1-cp312-cp312-win_amd64.whl.metadata (60 kB)
```

```
Using cached numpy-2.3.1-cp312-cp312-win_amd64.whl (12.7 MB)
```

```
Installing collected packages: numpy
```

```
Successfully installed numpy-2.3.1
```

```
(haesung) PS C:\Users\hayunjong83>
```



두 행렬  $A = \begin{pmatrix} -3 & 1 \\ 2 & 4 \end{pmatrix}$ ,  $B = \begin{pmatrix} 2 & 3 \\ 4 & -1 \end{pmatrix}$ 에 대하여 행렬  $2A + B$ 는?

①  $\begin{pmatrix} -4 & 5 \\ 8 & 7 \end{pmatrix}$

②  $\begin{pmatrix} -1 & 5 \\ 6 & 7 \end{pmatrix}$

③  $\begin{pmatrix} -4 & 4 \\ 8 & 3 \end{pmatrix}$

④  $\begin{pmatrix} -1 & 4 \\ 6 & 3 \end{pmatrix}$

⑤  $\begin{pmatrix} -2 & 5 \\ 6 & 2 \end{pmatrix}$

Anaconda PowerShell Prompt

```
(haesung) PS C:\Users\hayunjong83> python
Python 3.12.11 | packaged by Anaconda, Inc. | (main, Jun 5 2025, 12:58:53)
[MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> A = np.array([ [-3, 1], [2, 4] ])
>>> B = np.array([ [ 2, 3], [4, -1] ])
>>> 2*A + B
array([[ -4,  5],
       [ 8,  7]])
>>> _
```

두 행렬  $A, B$ 에 대하여  $A+B=\begin{pmatrix} 2 & 2 \\ 3 & 1 \end{pmatrix}$ ,  $B=\begin{pmatrix} 0 & 1 \\ 3 & 2 \end{pmatrix}$ 일 때,

행렬  $AB+B^2$ 은? [3점]

①  $\begin{pmatrix} 3 & 5 \\ 6 & 6 \end{pmatrix}$

②  $\begin{pmatrix} 3 & 1 \\ 12 & 8 \end{pmatrix}$

③  $\begin{pmatrix} 6 & 6 \\ 3 & 5 \end{pmatrix}$

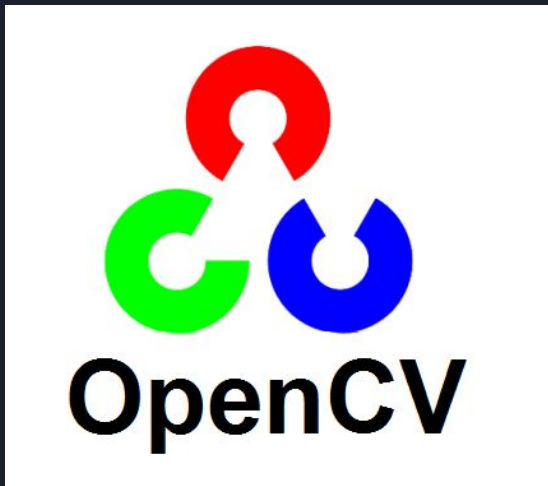
④  $\begin{pmatrix} 6 & 8 \\ 3 & 1 \end{pmatrix}$

⑤  $\begin{pmatrix} 12 & 8 \\ 3 & 1 \end{pmatrix}$

```
Anaconda PowerShell Prompt
(haesung) PS C:\Users\Whayunjong83> python
Python 3.12.11 | packaged by Anaconda, Inc. | (main, Jun 5 2025, 12:58:53)
[MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> X = np.array([[2, 2], [3, 1]])
>>> Y = np.array([[0, 1], [3, 2]])
>>> X@Y
array([[6, 6],
       [3, 5]])
>>> _
```



## OpenCV를 이용한 디지털 이미지/영상 다루기



- **OpenCV(Open Source Computer Vision)**  
인텔이 시작한 실시간 컴퓨터 비전 목적의 라이브러리
- 강력한 이미지 처리 알고리즘 제공
- 머신러닝과 딥러닝 기반의 알고리즘도 제공

Anaconda PowerShell Prompt

```
(base) PS C:\Users\hayunjong83> conda activate haesung  
(haesung) PS C:\Users\hayunjong83> pip install opencv-python
```

- 가상환경 활성화
- 필요한 라이브러리 ( **opencv-python** ) 설치

## (0) 실습 준비

```
Anaconda PowerShell Prompt
(haesung) PS C:\Users\hayunjong83> cd Desktop
(haesung) PS C:\Users\hayunjong83\Desktop> mkdir summer_capstone

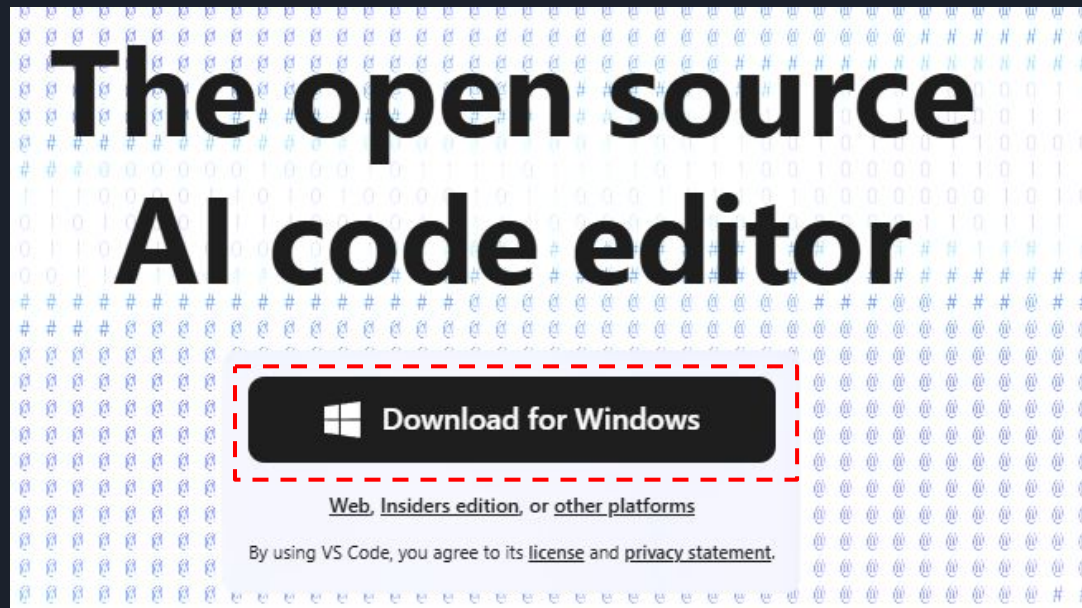
Directory: C:\Users\hayunjong83\Desktop

Mode                LastWriteTime         Length Name
----                -
d-----          2025-07-21 오후 12:09                summer_capstone

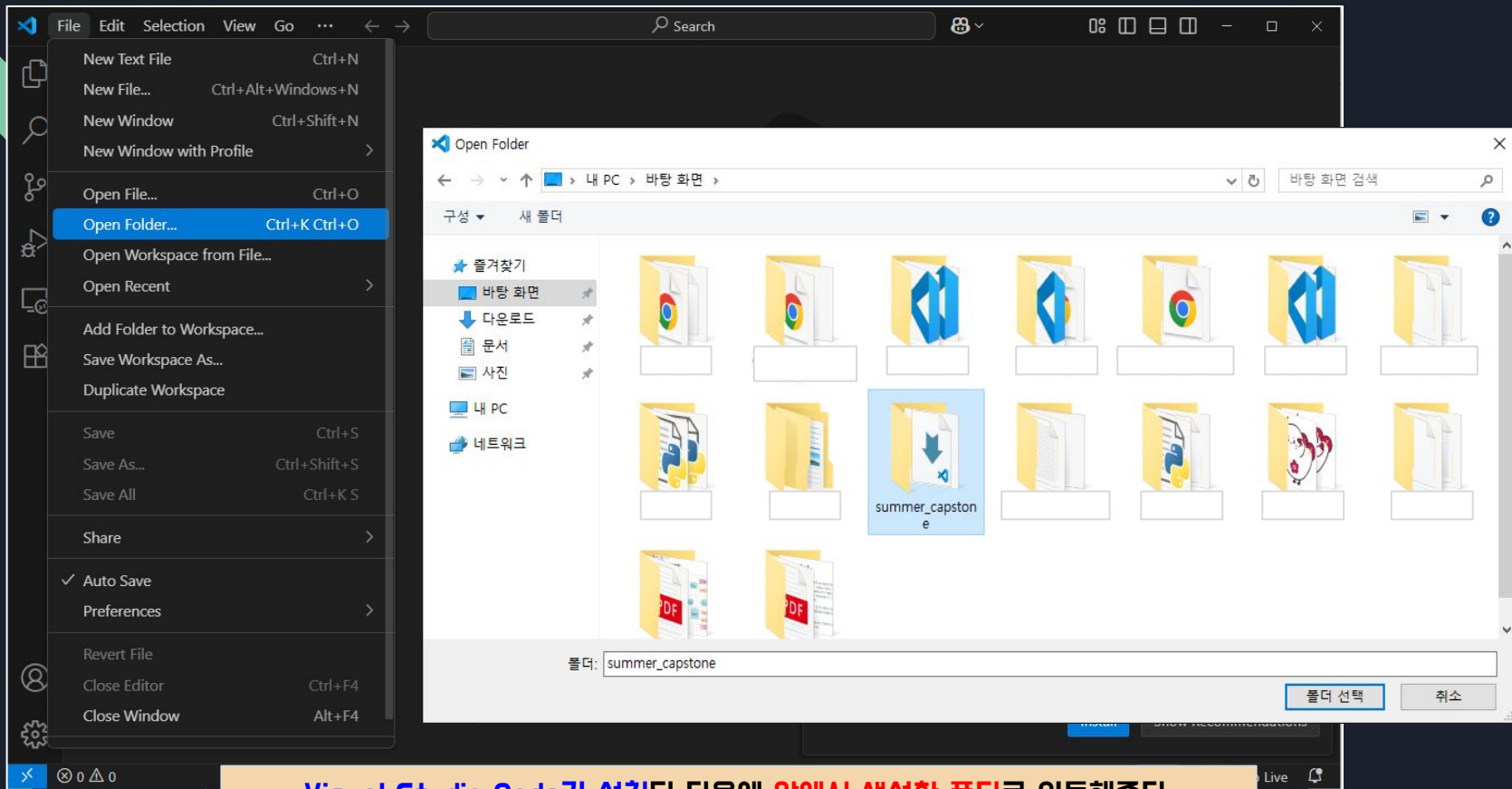
(haesung) PS C:\Users\hayunjong83\Desktop> cd summer_capstone
(haesung) PS C:\Users\hayunjong83\Desktop\summer_capstone> _
```

바탕화면에 **summer\_capstone** 이라는 이름의 폴더를 만들고, **생성된 폴더로 이동한다.**

앞으로 직접 코드를 작성하고 수정할 수 있도록, 코드 편집기인 Visual Studio Code를 설치해준다.



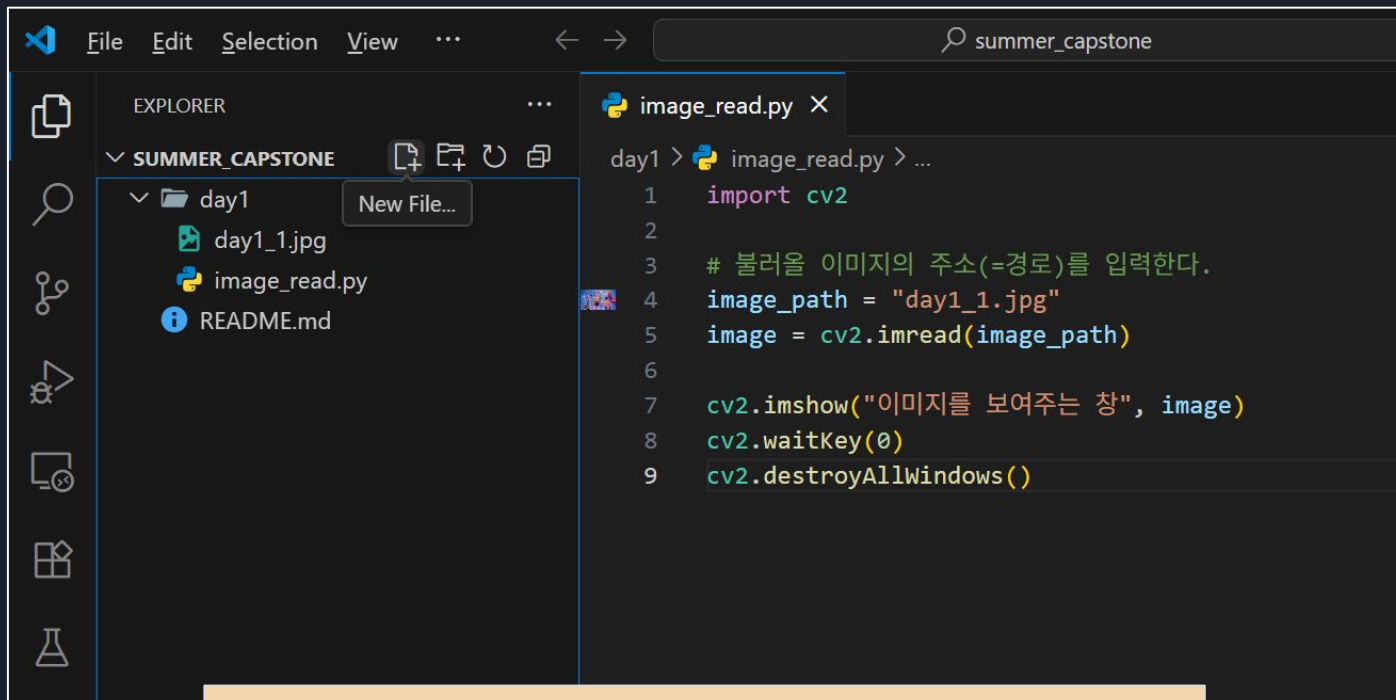
<https://code.visualstudio.com/> 에서 설치파일을 받아서 다운로드 해준다.



Visual Studio Code가 설치된 다음에 앞에서 생성한 폴더로 이동해준다.

## (1) 이미지 읽기

- 사용한 코드 : `image_read.py`
- 사용한 이미지 : `day1_1.jpg`

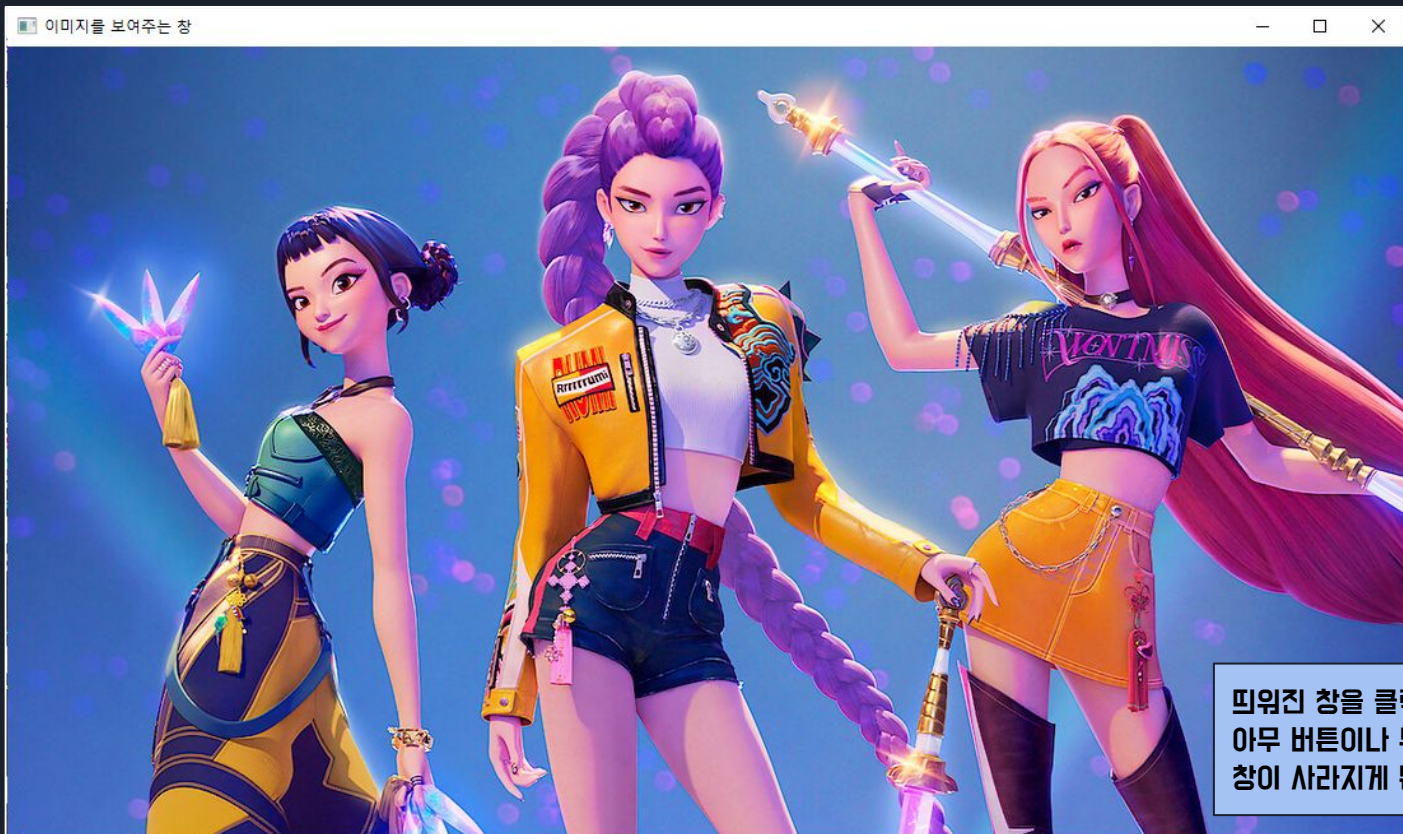


`image_read.py` 라는 이름의 파일에 위의 코드를 작성해준다.

파일명은 자신이 원하는 대로 수정할 수 있다.

Anaconda PowerShell Prompt

```
(haesung) PS C:\Users\hayunjong83\Desktop\summer_capstone\day1> python image_read.py
```





## (2) 동영상 읽기

- 사용한 코드 : [video\\_read.py](#)
- 사용한 영상 : [day1\\_2.mp4](#)

video\_read.py X

day1 > video\_read.py > ...

```
1 import cv2
2
3 # 불러올 동영상의 주소(=경로)를 입력한다.
4 video_path = "day1_2.mp4"
5
6 cap = cv2.VideoCapture(video_path)
7 if cap.isOpened():
8     while True:
9         ret, img = cap.read()
10        if ret:
11            cv2.imshow("영상을 보여주는 창", img)
12        else:
13            break
14
15        if cv2.waitKey(30) == ord('q'):
16            break
17
18 cap.release()
19 cv2.destroyAllWindows()
```

주어진 경로의 비디오 파일을 읽는다.

비디오에서 읽은 이미지를 화면에 보여준다.

화면에 더 이상 보여줄 영상이 없거나  
'q'를 입력하면 영상 재생을 종료한다.

**video\_read.py** 라는 이름의 파일에 위의 코드를 작성해준다.





```
g83\Desktop\summer_capstone\day1> python video_read.py
```

영상 재생이 끝나거나. 'q' 버튼을 누르면 재생창이 사라지게 된다.

### (3) 웹캠으로부터 영상 입력 받기

webcam\_read.py U X

day1 > webcam\_read.py > ...

```
1  import cv2
2
3  cap = cv2.VideoCapture(0)
4  if cap.isOpened():
5      while True:
6          ret, img = cap.read()
7          if ret:
8              cv2.imshow("웹캠을 보여주는 창", img)
9          else:
10             print("카메라에서 영상을 읽을 수 없습니다.")
11             break
12
13         if cv2.waitKey(1) == ord('q'):
14             break
15     else:
16         print("웹캠 카메라를 열 수 없습니다.")
17
18 cap.release()
19 cv2.destroyAllWindows()
```

→ 컴퓨터에 연결된 비디오장치 입력을 받는다.

webcam\_read.py 라는 이름의 파일에 위의 코드를 작성해준다.

#### (4) OpenCV를 이용하여 도형 그리기

- 사용한 코드 : [image\\_write.py](#)
- 결과 이미지 : [test.jpg](#)

##### - 직사각형 그리기

```
cv2.rectangle( 도형이 그려질 이미지,  
              (왼쪽 상단 꼭지점 좌표),  
              (오른쪽 하단 꼭지점 좌표),  
              색상 정보, 테두리 선 두께  
              )
```

##### - 원 그리기

```
cv2.circle( 도형이 그려질 이미지,  
            (원의 중심 좌표),  
            반지름 , 색상 정보, 테두리 선  
            두께  
            )
```

##### - 직선 그리기

```
cv2.line( 도형이 그려질 이미지,  
          (직선의 시작 좌표),  
          (직선의 끝 좌표),  
          색상정보, 테두리 선 두께  
          )
```

image\_write.py U X

day1 > image\_write.py > ...

```
1 import cv2  
2 import numpy as np  
3  
4 # 높이가 400픽셀, 너비가 600픽셀인 흰 배경 이미지를 만든다.  
5 bg = np.ones((400, 600, 3), dtype=np.uint8) * 255  
6  
7 # 직사각형 그리기  
8 # 시작점 => 화면 좌측상단에서 왼쪽으로 100px, 아래로 100px  
9 # 직사각형의 높이와 너비는 각각 50px  
10 # 파란색 (255, 0, 0)  
11 cv2.rectangle(bg, (100, 100), (150, 150), (255, 0, 0), 2)  
12  
13 # 원 그리기  
14 # 중심점 => 화면 좌측상단에서 왼쪽으로 400px, 아래로 100px  
15 # 반지름은 50px  
16 # 초록색 (0, 255, 0)  
17 cv2.circle(bg, (400, 100), 50, (0, 255, 0), -1)  
18  
19 cv2.imshow("결과를 보여주는 창", bg)  
20 cv2.waitKey(0)  
21 cv2.destroyAllWindows()  
22  
23 cv2.imwrite("test.jpg", bg)
```

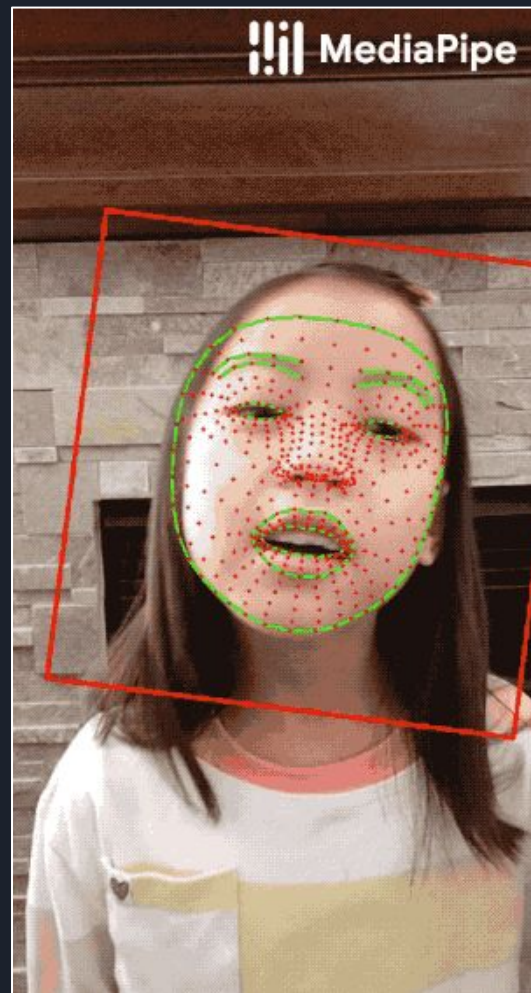
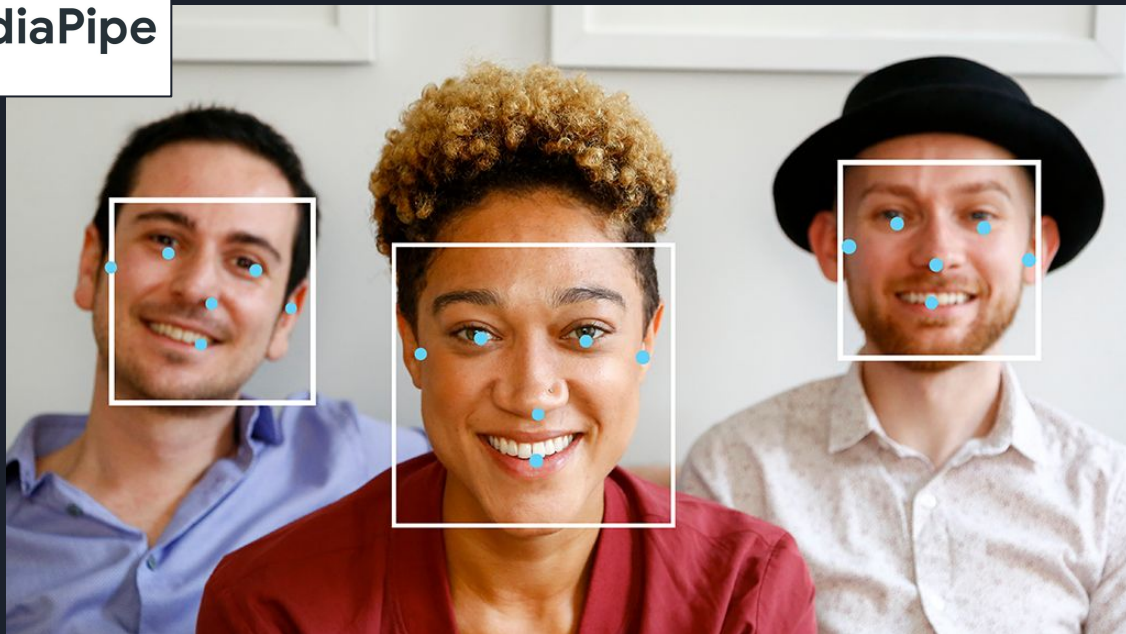
결과 이미지를 저장할 때

(5) 얼굴 인식 결과를 OpenCV를 이용하여 표현하기



MediaPipe

구글에서 제공하는 미디어파이프(Mediapipe)를 이용하면  
얼굴 인식 등 다양한 영상 처리를 간단하게 수행할 수 있다.







참고) [MediaPipe - The Ultimate Guide to Video Processing](#)

솔루션

LLM Inference API

객체 감지

이미지 분류

이미지 세분화

대화형 세분화

손 랜드마크 감지

동작 인식

이미지 임베딩

얼굴 인식

얼굴 특징 감지

얼굴 스타일 지정

자세 랜드마크 감지

이미지 생성

텍스트 분류

텍스트 임베딩

언어 감지기

오디오 분류

## 0. 미디어파이프 라이브러리 설치

```
pip install mediapipe
```

Anaconda PowerShell Prompt

```
(haesung) PS C:\Users\hayunjong83\Desktop\summer_capstone\day1> pip install mediapipe
```

### 1. 이미지에 있는 사람 얼굴을 인식하기

- 사용한 코드 : [face\\_detect\\_image.py](#)
- 사용한 이미지 : [day1\\_4.jpg](#), [day1\\_5.jpg](#)

```
import cv2
import mediapipe as mp

# 이미지를 읽기 & 높이/너비 알아내기
img_path = "day1_5.jpg"
img = cv2.imread(img_path)
h, w, c = img.shape
```



# 미디어파이프

```
mp_face = mp.solutions.face_detection
face_detection = mp_face.FaceDetection(
    model_selection = 0,
    min_detection_confidence = 0.5)
```

# 이미지를 RGB 순서로 바꿔서 입력

```
src = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

# 얼굴 탐색 하기

```
results = face_detection.process(src)
```

얼굴인식을 위한 미디어파이프 객체를 불러와 선언한다.



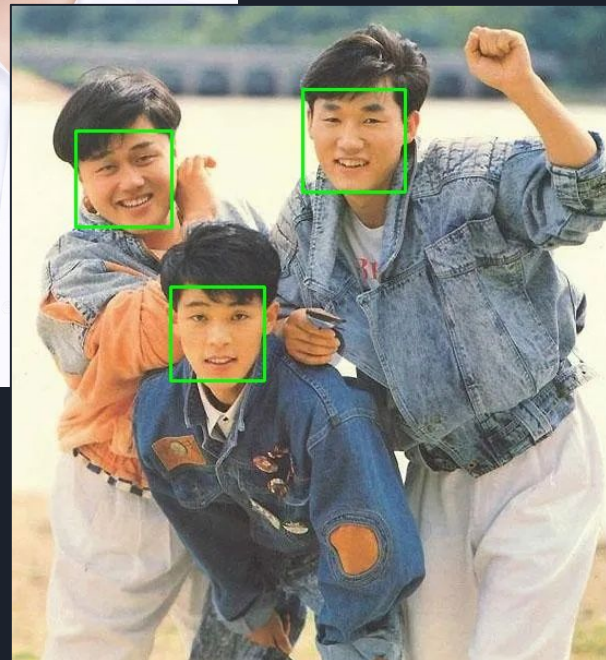
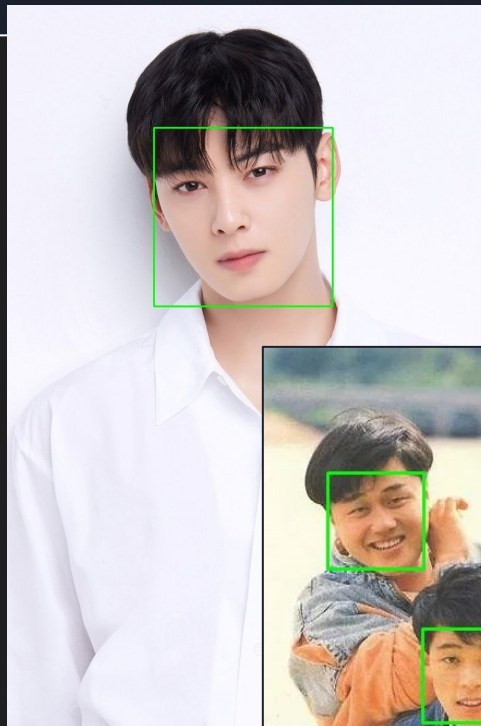
미디어파이프 객체에 처리할 이미지를 넘겨준다.

```
# 탐색한 얼굴 표현하기
if results.detections:
    for det in results.detections:
        bbox = det.location_data.relative_bounding_box
        x = int(bbox.xmin * w)
        y = int(bbox.ymin * h)
        bbox_w = int(bbox.width * w)
        bbox_h = int(bbox.height * h)

        cv2.rectangle(img,
                       (x, y), (x + bbox_w, y + bbox_h),
                       (0, 255, 0), 2)

# 결과 보여주기
cv2.imshow("얼굴 인식 결과", img)
cv2.waitKey(0)
cv2.destroyAllWindows()

# 결과 저장
cv2.imwrite("test3.jpg", img)
```





## 2. 영상에 있는 사람 얼굴을 인식하기

- 사용한 코드 : [face\\_detect\\_video.py](#)
- 사용한 영상 : [day1\\_2.mp4](#)

```
import cv2
import mediapipe as mp

video_path = "day1_2.mp4" → 읽을 영상의 경로
result_path = "test4.mp4" → 결과를 저장할 경로

# 미디어파이프
mp_face = mp.solutions.face_detection
face_detection = mp_face.FaceDetection(
    model_selection = 1,
    min_detection_confidence = 0.2)

cap = cv2.VideoCapture(video_path)

# 영상정보를 얻는다.
fps = cap.get(cv2.CAP_PROP_FPS)
w = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
h = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

# 저장을 위한 설정
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(result_path, fourcc, fps, (w, h))
```

저장되는 이미지 크기가 제대로 설정되지 않으면  
결과 영상이 기록되지 않을 수 있다.

```

while cap.isOpened():
    ret, img = cap.read()

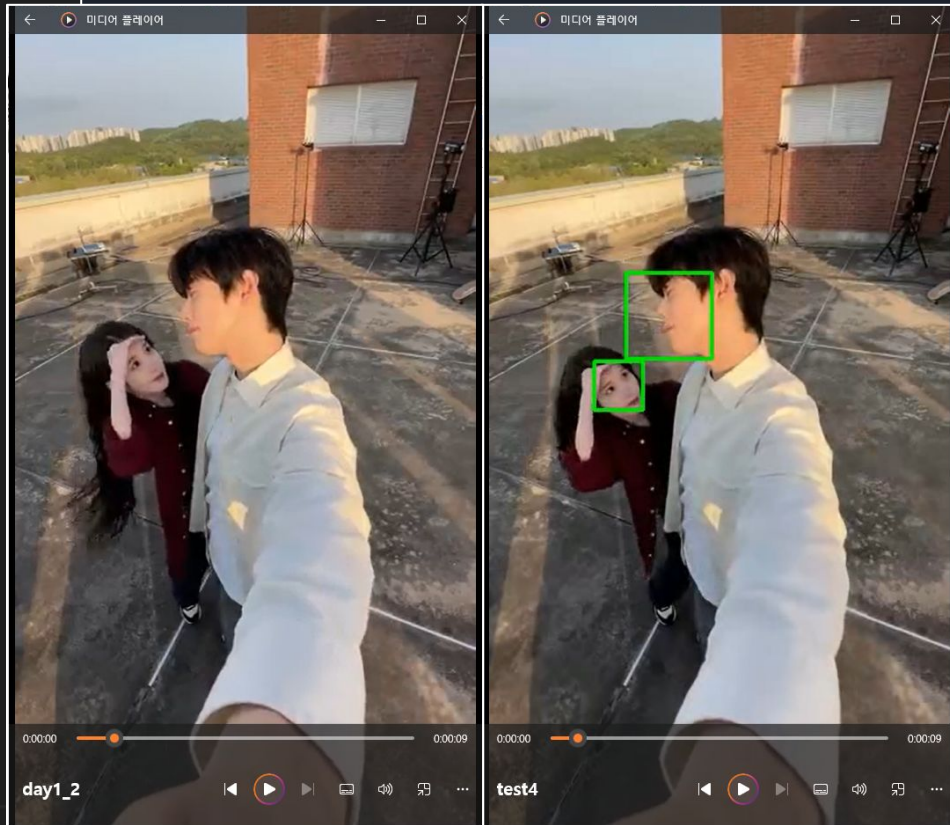
    if ret:
        src = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        # 얼굴 탐지
        results = face_detection.process(src)
        if results.detections:
            for det in results.detections:
                bbox = det.location_data.relative_bounding_box
                x = int(bbox.xmin * w)
                y = int(bbox.ymin * h)
                bbox_w = int(bbox.width * w)
                bbox_h = int(bbox.height * h)

                cv2.rectangle(img,
                               (x, y), (x + bbox_w, y + bbox_h),
                               (0, 255, 0), 2)

        # 영상을 기록
        out.write(img)
        cv2.imshow("얼굴인식결과", img)
        if cv2.waitKey(30) == ord('q'):
            break
    else:
        break

```

영상에서 이미지 한장. 한장을 차례로 처리한다고 생각하면  
비디오에서의 영상처리를 쉽게 이해할 수 있다.

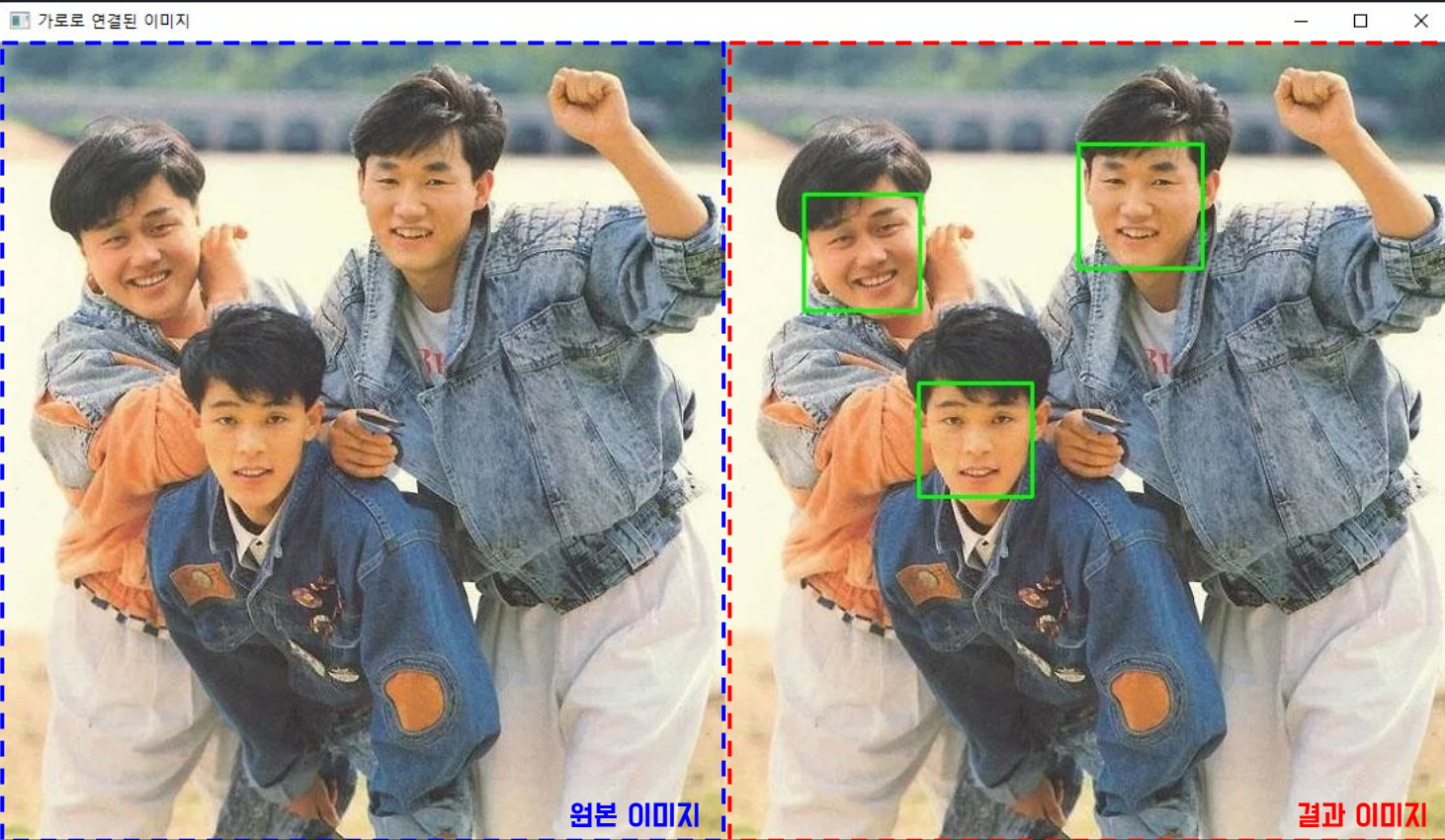


## (6) 원본영상과 결과영상을 나란히 붙여서 비교하기

- 사용한 코드 : [image\\_concat.py](#)
- 결과 이미지 : [test5.jpg](#)

```
image_concat.py U X
day1 > image_concat.py > ...
1  import cv2
2  import numpy as np
3
4  # 원본 이미지
5  original = cv2.imread('day1_4.jpg')
6  # 처리된 이미지
7  processed = cv2.imread('test2.jpg')
8
9  # 가로로 이미지를 연결
10 result = np.hstack((original, processed))
11
12 cv2.imshow("가로로 연결된 이미지", result)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
15
16 # 이미지 저장
17 cv2.imwrite("test5.jpg", result)
```

연결할 두 이미지의 높이가 같을 경우에  
np.hstack()을 이용해서 두 이미지를 연결할 수 있다.







동일한 방법을 이용해서  
두 개의 비디오 영상을 연결할 수도 있다.

- 사용한 코드 : [video\\_concat.py](#)
- 결과 영상 : [test6.mp4](#)



**Q & A**

**감사합니다**