

Computer Vision

# L01. OpenCV

동아대학교 컴퓨터AI공학부  
서정일 교수



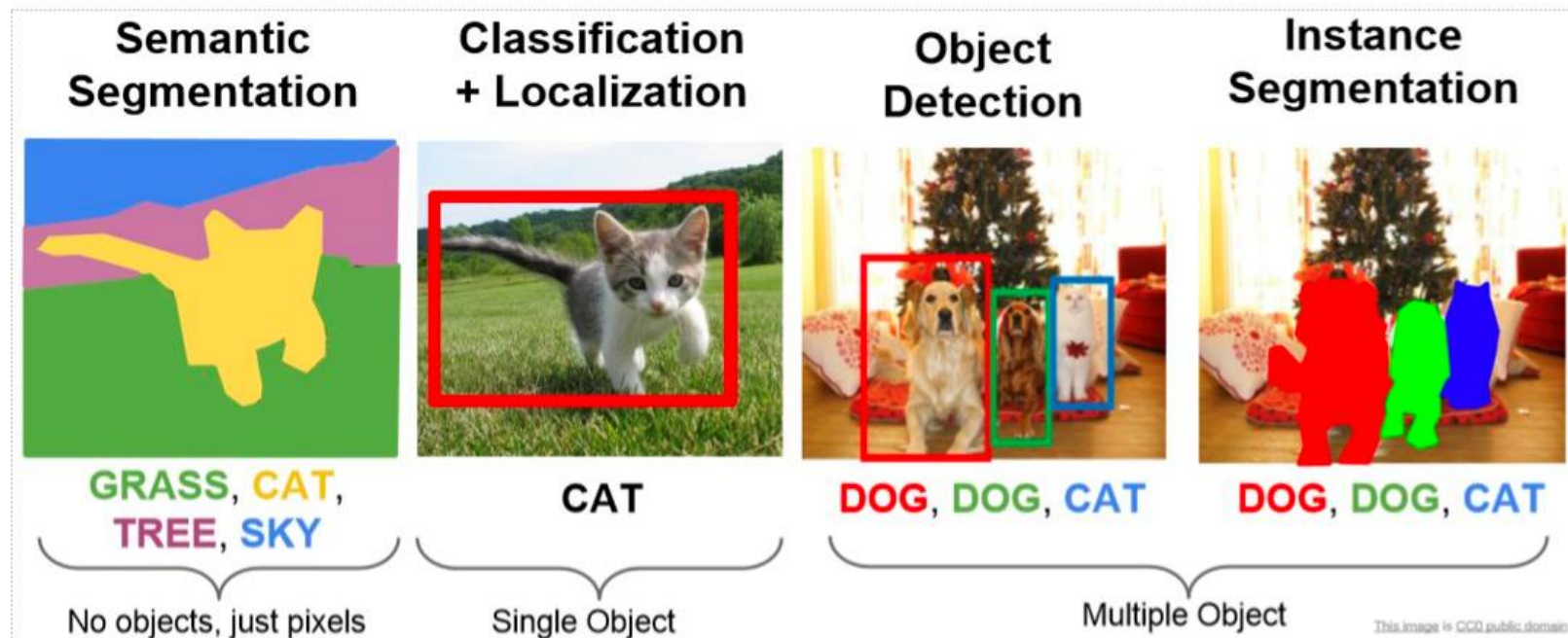
# Context

- OpenCV 소개
- 객체지향 잘 활용하기
- 영상을 읽고 표시하기
- 영상 형태 변환하고 크기 축소하기
- 웹 캠에서 비디오 읽기
- 그래픽 기능과 사용자 인터페이스 만들기
- 페인팅 기능 만들기

# 1. OpenCV

## 컴퓨터비전

- 이미지 분류
- Detection(객체 감지)
- Tracking(객체 추적)
- 콘텐츠 기반 이미지 검색



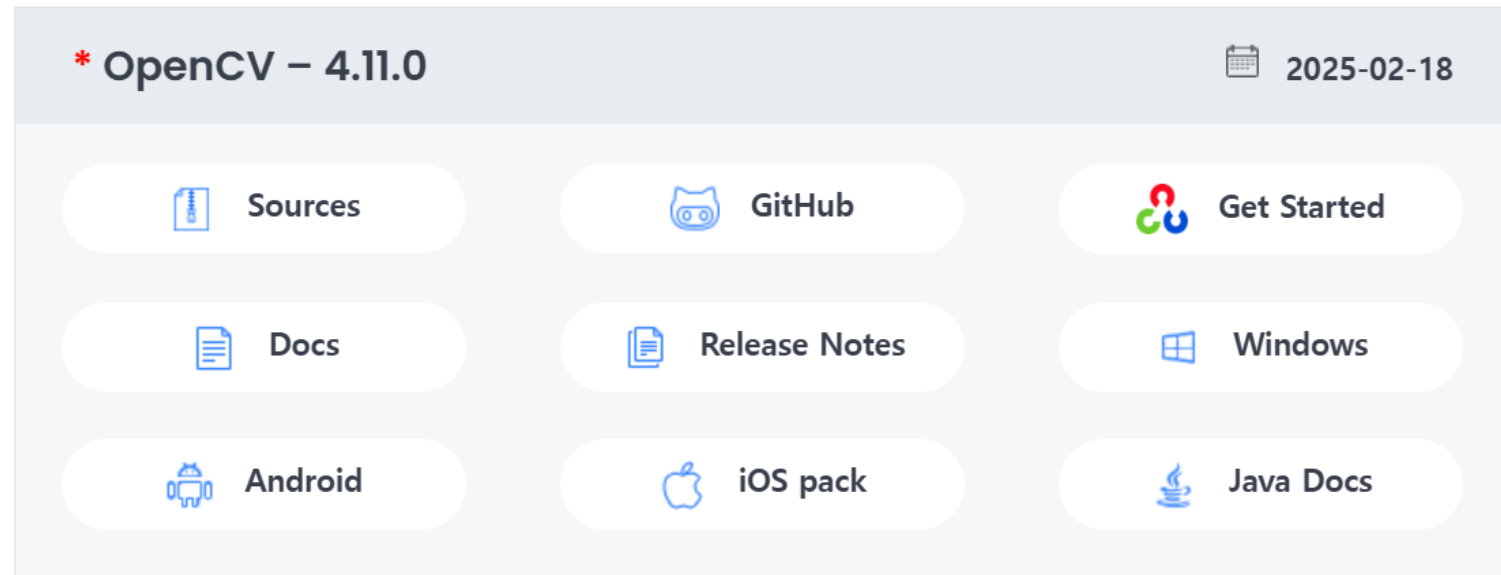
# 1. OpenCV

## 컴퓨터비전의 방법 분류(IBM)

- 다중 신경망 방식(MLP)  
주로 이미지 분류(Image Classification)에서 사용  
충분한 양의 데이터를 가지고 훈련하여 이미지를 분류함
- 컨볼루션 방식(CNN)  
컨볼루션 신경망을 통한 학습을 이용  
Detection, Tracking 등에서 사용  
CNN은 단일 이미지에 주로 적용 / RNN은 비디오에 주로 적용

# 1. OpenCV

- **OpenCV (Open Computer Vision Library)**
  - Intel의 오픈소스 컴퓨터 비전 라이브러리
- **C/C++로 작성되었음**
- **현재 OpenCV 4.11.0 버전까지 릴리즈됨**



## 2. 객체지향 잘 활용하기

OpenCV : C/C++로 작성된 오픈소스 컴퓨터 비전 라이브러리

>> OpenCV-Python는 OpenCV를 Python으로 바인딩한 버전

C++, Python으로 OpenCV를 사용하는 이유?

>> 컴퓨터 비전 프로그래밍에선 객체지향 언어가 유용하기 때문

## 2. 객체지향 잘 활용하기

객체지향 언어가 CV Programming에서 유리한 이유?

>> OOP의 특징을 살펴보면 답이 나옴.

1. **모듈화와 재사용성** : 여러 작업들을 재사용 가능한 객체로 나누어 개발
2. **추상화** : 이미지/영상과 관련된 여러 추상적 개념과 연산을 개발 가능
3. **상속과 다형성** : 객체 상속 및 함수 오버로딩 등을 활용해 유연한 개발 가능
4. **유지보수 용이성** :  
CV 개발에선 알고리즘 변경 및 기능 추가가 빈번 → 객체 사용 시 쉽게 개발 가능
5. **다중 스레딩 및 병렬 처리** :  
CV에선 병렬 처리를 통한 빠른 계산이 중요함 → OOP에서 쉽게 구현 가능

### 3. 영상을 읽고 표시하기

이 챕터에서 다루는 내용은 기본적인 **Image Read & Display**.

```
CODE > 0917_2.py > ...
1  import cv2 as cv
2  import sys
3
4  img = cv.imread('soccer.jpg')
5
6  if img is None :
7      sys.exit('파일이 존재하지 않습니다.')
8
9  cv.imshow('Image Display', img)
10
11  cv.waitKey()
12  cv.destroyAllWindows()
13
14  print(type(img))
15  print(img.shape)
```

OpenCV 모듈 import.  
sys 모듈 import

soccer.jpg file

[https://drive.google.com/file/d/1mRyLwFJHzXqRwXCJOXhyhmGwtCTGDkKx/view?usp=drive\\_link](https://drive.google.com/file/d/1mRyLwFJHzXqRwXCJOXhyhmGwtCTGDkKx/view?usp=drive_link)



### 3. 영상을 읽고 표시하기

이 챕터에서 다루는 내용은 기본적인 **Image Read & Display**.

CODE > 0917\_2.py > ...

```
1  import cv2 as cv
2  import sys
3
4  img = cv.imread('soccer.jpg')
5
6  if img is None :
7      sys.exit('파일이 존재하지 않습니다.')
8
9  cv.imshow('Image Display', img)
10
11  cv.waitKey()
12  cv.destroyAllWindows()
13
14  print(type(img))
15  print(img.shape)
```




soccer.jpg 라는 이미지 읽기. (Read)

### 3. 영상을 읽고 표시하기

이 챕터에서 다루는 내용은 기본적인 Image Read & Display.

```
CODE > 0917_2.py > ...
1  import cv2 as cv
2  import sys
3
4  img = cv.imread('soccer.jpg')
5
6  if img is None :
7      sys.exit('파일이 존재하지 않습니다.')
8
9  cv.imshow('Image Display', img)
10
11  cv.waitKey()
12  cv.destroyAllWindows()
13
14  print(type(img))
15  print(img.shape)
```



soccer.jpg가 존재하지 않는다면  
sys 모듈 이용 >> 프로그램 종료.

### 3. 영상을 읽고 표시하기

이 챕터에서 다루는 내용은 기본적인 **Image Read & Display**.

```
CODE > 0917_2.py > ...
1  import cv2 as cv
2  import sys
3
4  img = cv.imread('soccer.jpg')
5
6  if img is None :
7      sys.exit('파일이 존재하지 않습니다.')
8
9  cv.imshow('Image Display', img)
10
11  cv.waitKey()
12  cv.destroyAllWindows()
13
14  print(type(img))
15  print(img.shape)
```

Image Display 라는 창에  
img(soccer.jpg) 출력 (Display).

### 3. 영상을 읽고 표시하기

이 챕터에서 다루는 내용은 기본적인 Image Read & Display.

```
CODE > 0917_2.py > ...
1  import cv2 as cv
2  import sys
3
4  img = cv.imread('soccer.jpg')
5
6  if img is None :
7      sys.exit('파일이 존재하지 않습니다.')
8
9  cv.imshow('Image Display', img)
10
11  cv.waitKey()
12  cv.destroyAllWindows()
13
14  print(type(img))
15  print(img.shape)
```




Image Display 라는 창을  
키보드 입력이 들어올때까지 켜놓  
기.

### 3. 영상을 읽고 표시하기

Image Display



### 3. 영상을 읽고 표시하기

이 챕터에서 다루는 내용은 기본적인 Image Read & Display.

```
CODE > 0917_2.py > ...
1  import cv2 as cv
2  import sys
3
4  img = cv.imread('soccer.jpg')
5
6  if img is None :
7      sys.exit('파일이 존재하지 않습니다.')
8
9  cv.imshow('Image Display', img)
10
11  cv.waitKey()
12  cv.destroyAllWindows()
13
14  print(type(img))
15  print(img.shape)
```

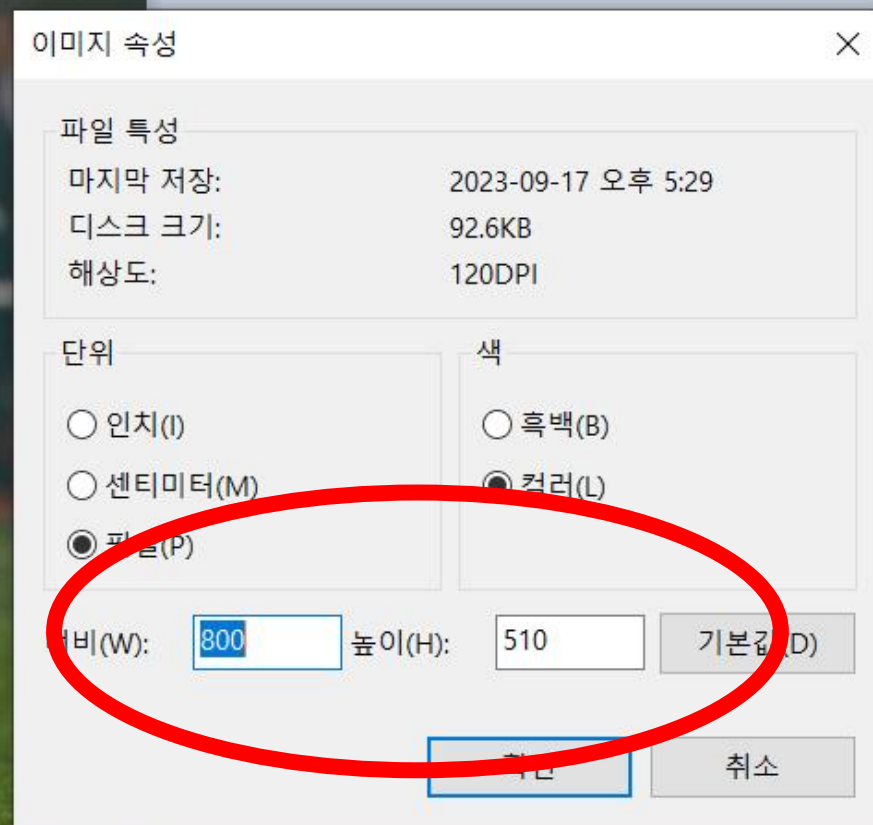
Image Display 창이 꺼지면  
img의 type과 shape를 출력함.



### 3. 영상을 읽고 표시하기

```
<class 'numpy.ndarray'>  
(510, 800, 3)
```

← img는 numpy.ndarray 타입으로 저장됨. << 이미지를 행렬로 저장.  
행이 510개, 열이 800개, 색상값이 3개(RGB 값)라는 걸 확인.



### 3. 영상을 읽고 표시하기

CODE > 0917\_3.py > ...

```
1  import cv2 as cv
2  import sys
3
4  img = cv.imread('soccer.jpg')
5
6  cv.imshow('Image Display', img)
7  print(img[0,0,0], img[0,0,1],img[0,0,2])
8
9  cv.waitKey()
10 cv.destroyAllWindows()
```

Image Display라는 창에 img 띄우고,  
(0,0)의 B,G,R값 출력.

R,G,B 가 아니라 B,G,R 임에 유의!



### 3. 영상을 읽고 표시하기

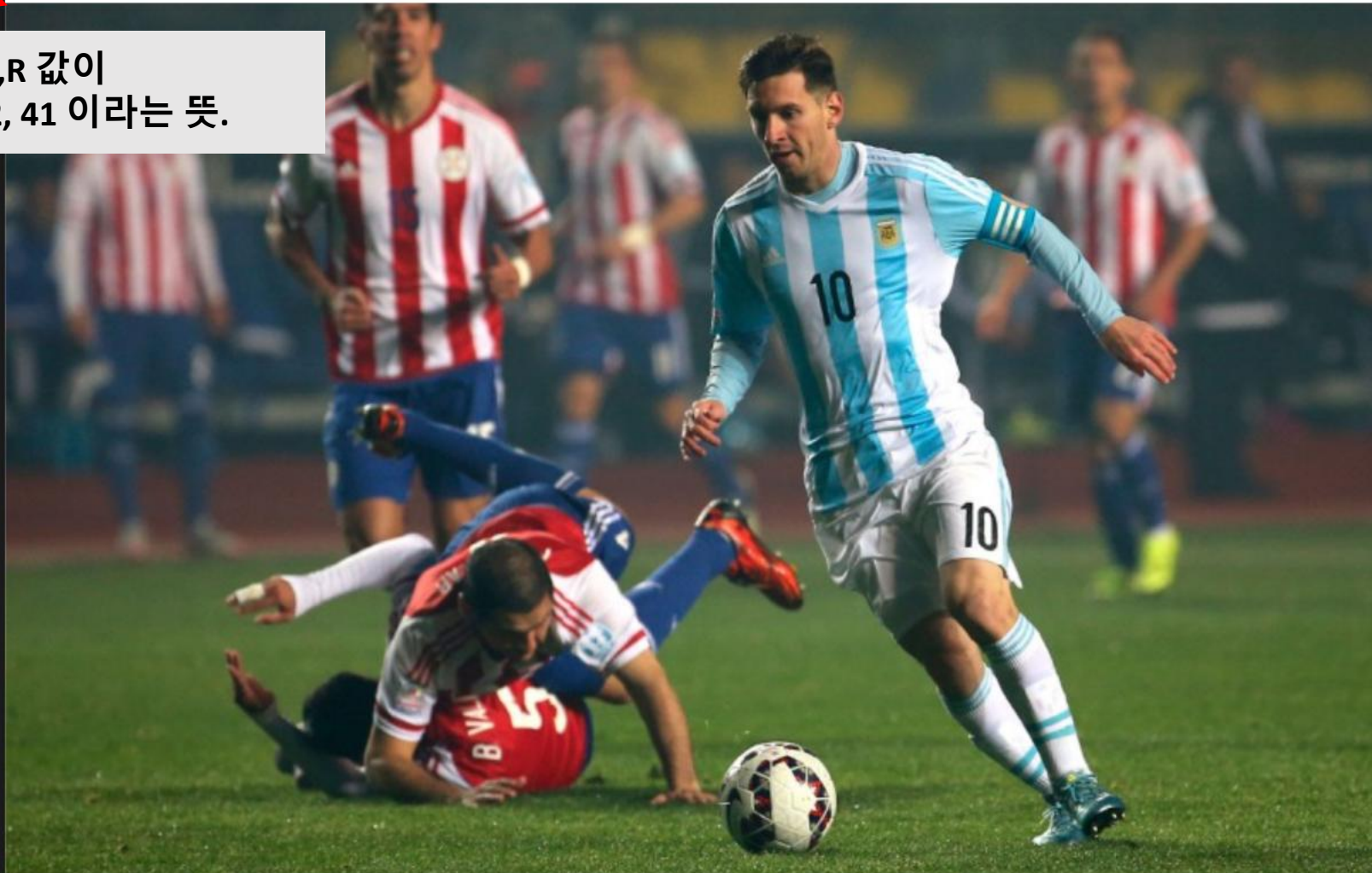
```
(cv) D:\학술\연구실\2학년 2학기\CV&DL Study\CODE>python 0917_3.py
```

54 52 41



Image Display

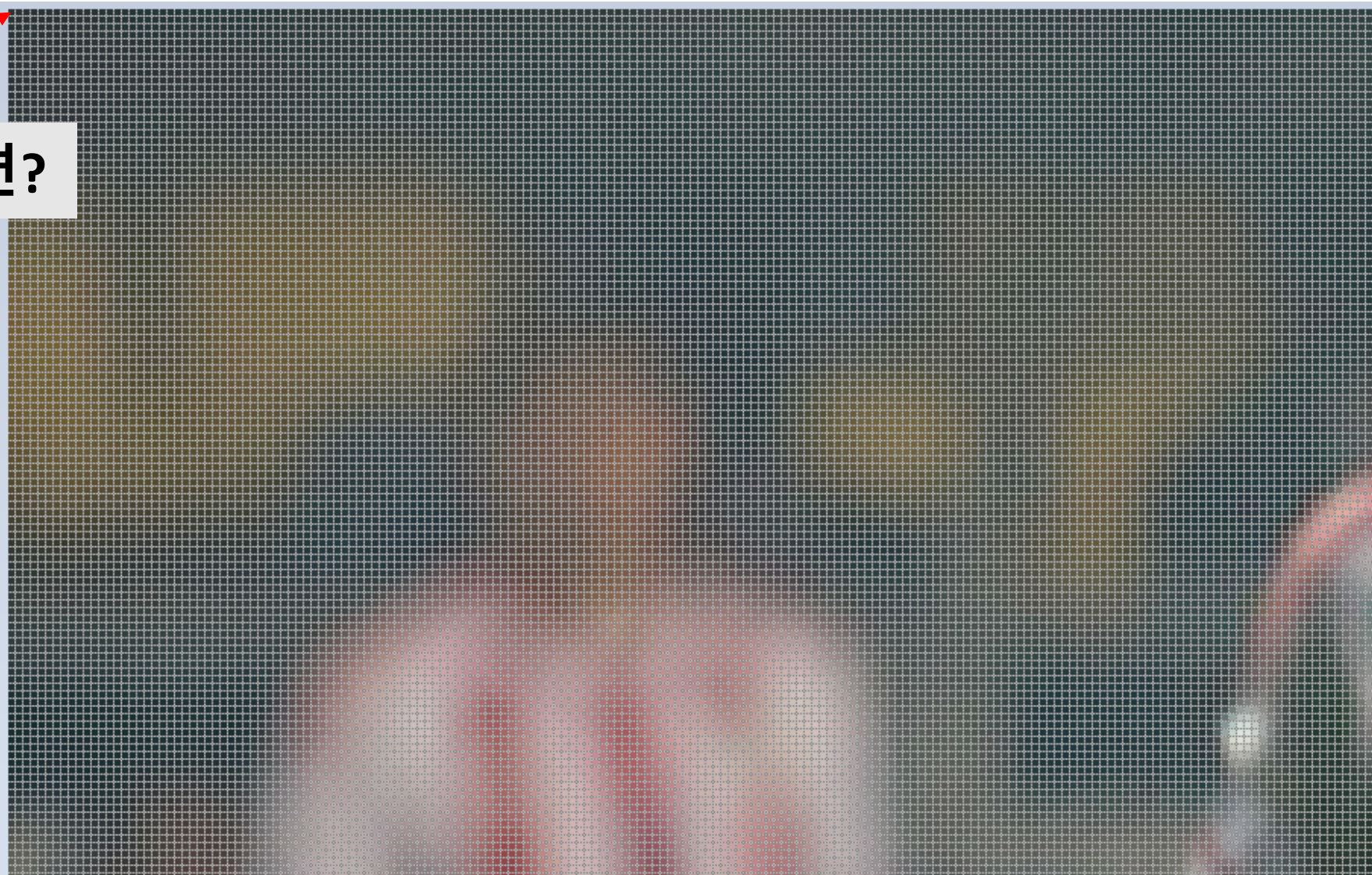
(0,0)의 B,G,R 값이  
각각 54, 52, 41 이라는 뜻.



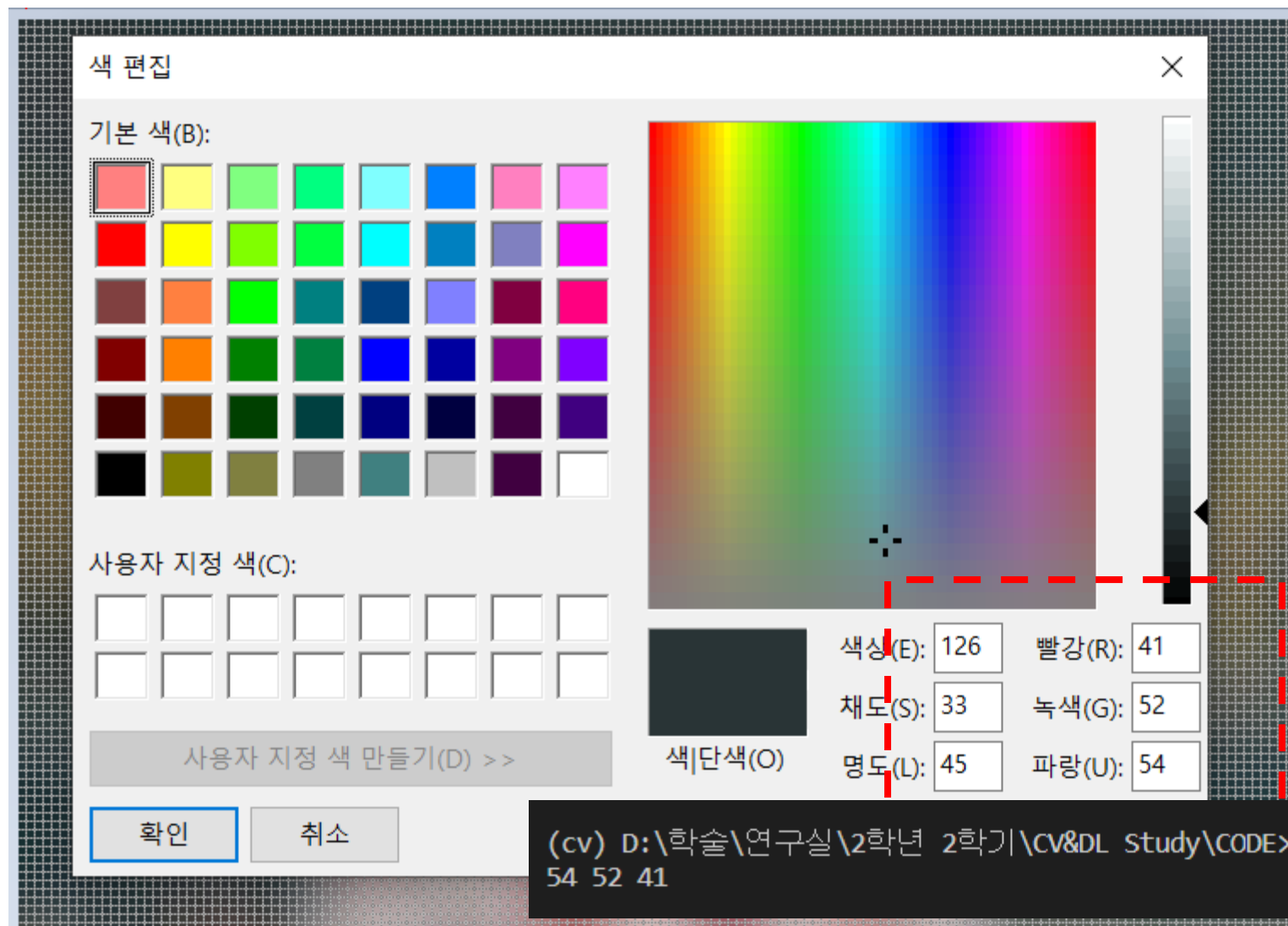


### 3. 영상을 읽고 표시하기

확인해보면?



### 3. 영상을 읽고 표시하기



## 4. 영상 형태 변환하고 크기 축소하기

09행 : cvtColor 함수 -> 명암 영상 변환

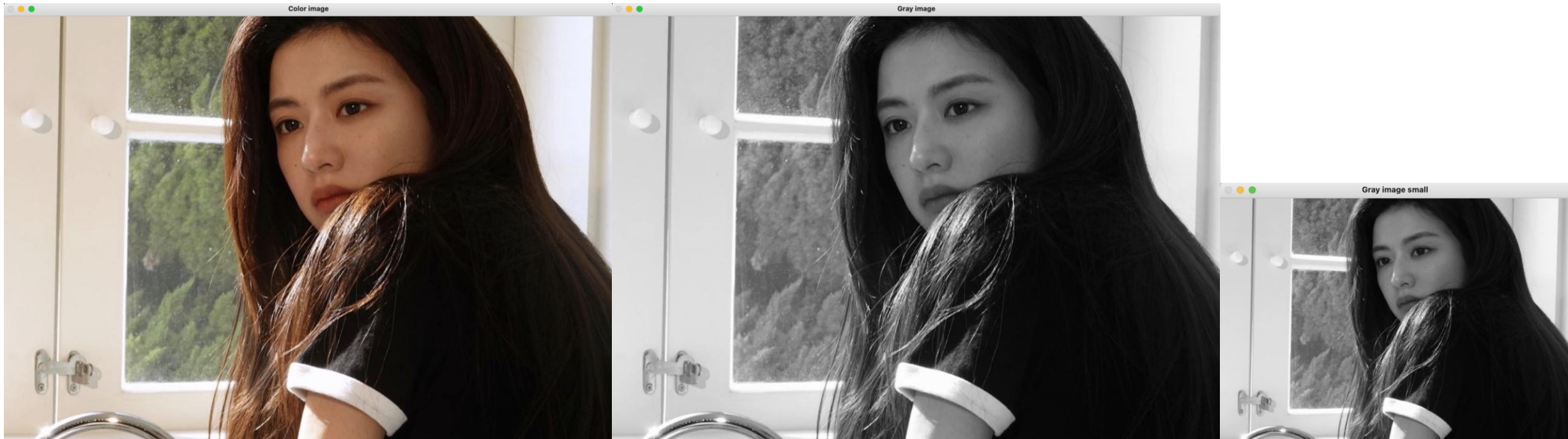
10행 : resize 함수 -> 영상 크기 변환



```
1  import cv2 as cv
2  import sys
3
4  img=cv.imread('source_4548_1/source/ch2/IMG_2362.jpg')
5
6  if img is None:
7      sys.exit('파일을 찾을 수 없습니다.')
8
9  gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY) # BGR 컬러 영상을 명암 영상으로 변환
10 gray_small=cv.resize(gray,dsize=(0,0),fx=0.5,fy=0.5) # 반으로 축소
11
12 cv.imwrite('soccer_gray.jpg',gray) # 영상을 파일에 저장
13 cv.imwrite('soccer_gray_small.jpg',gray_small)
14
15 cv.imshow('Color image',img)
16 cv.imshow('Gray image',gray)
17 cv.imshow('Gray image small',gray_small)
18
19 cv.waitKey()
20 cv.destroyAllWindows()
```



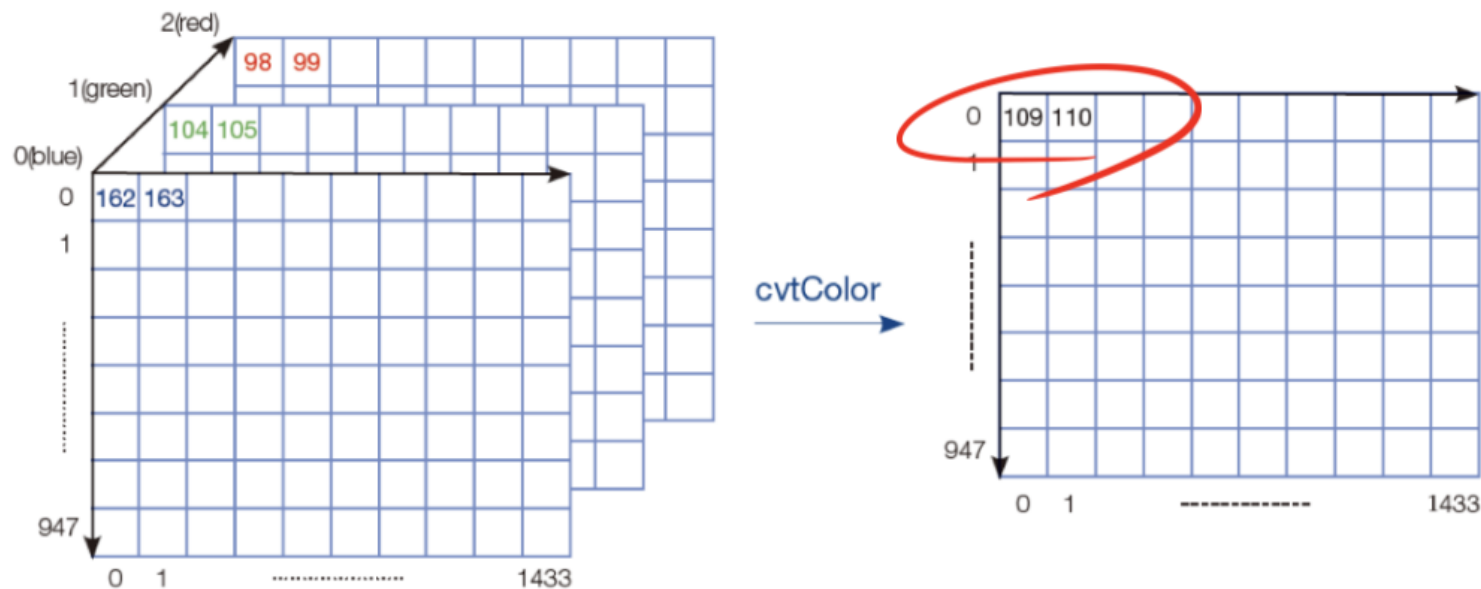
## 4. 영상 형태 변환하고 크기 축소하기



## 4. 영상 형태 변환하고 크기 축소하기

RGB[A] to Gray:  $Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$

$$I = \text{round}(0.299 \times R + 0.587 \times G + 0.114 \times B) \quad (2.1)$$



RGB에서 그레이스케일로 변환할 때 각 색상 채널(Red, Green, Blue)에 곱해지는 계수가 다른 이유는 인간의 눈이 다른 색상에 대해 서로 다른 민감도를 가지기 때문이며, 인간의 시각 시스템은 녹색에 가장 민감하고, 파란색에 가장 덜 민감함

## 5. 웹 캠에서 비디오 읽기

```
1  import cv2 as cv
2  import sys
3
4  cap=cv.VideoCapture(0,cv.CAP_DSHOW) # 카메라와 연결 시도
5
6  if not cap.isOpened():
7      sys.exit('카메라 연결 실패')
8
9  while True:
10     ret,frame=cap.read()           # 비디오를 구성하는 프레임 획득
11
12     if not ret:
13         print('프레임 획득에 실패하여 루프를 나갑니다.')
14         break
15
16     cv.imshow('Video display',frame)
17
18     key=cv.waitKey(1) # 1밀리초 동안 키보드 입력 기다림
19     if key==ord('q'): # 'q' 키가 들어오면 루프를 빠져나감
20         break
21
22 cap.release() # 카메라와 연결을 끊음
23 cv.destroyAllWindows()
```

## 5. 웹 캠에서 비디오 읽기

- `cv.VideoCapture` : 웹 캠과 연결을 시도
- `cv.CAP_DSHOW` : 비디오가 화면에 바로 나타나게 함

```
1  import cv2 as cv
2  import sys
3
4  cap=cv.VideoCapture(0,cv.CAP_DSHOW) # 카메라와 연결 시도
5
6  if not cap.isOpened():
7      sys.exit('카메라 연결 실패')
```



## 5. 웹 캠에서 비디오 읽기

- **ret** : 카메라로부터 프레임을 성공적으로 읽었는지를 나타내는 변수

```
9  ∨ while True:
10      ret,frame=cap.read()          # 비디오를 구성하는 프레임 획득
11  ∨  if not ret:
12      |    print('프레임 획득에 실패하여 루프를 나갑니다.')
13      |    break
14
15      cv.imshow('Video display',frame)
16
17      key=cv.waitKey(1)    # 1밀리초 동안 키보드 입력 기다림
18  ∨  if key==ord('q'):    # 'q' 키가 들어오면 루프를 빠져나감
19      |    break
```

## 5. 웹 캠에서 비디오 읽기

- `cap.release()` : 카메라 연결 해제
- `cv.destroyAllWindows` : 윈도우창 종료

```
21     cap.release()           # 카메라와 연결을 끊음
22     cv.destroyAllWindows()
```

## 5. 웹 캠에서 비디오 읽기



## 6. 비디오에서 수집한 영상 이어 붙이기

```

1  import cv2 as cv
2  import numpy as np
3  import sys
4
5  cap=cv.VideoCapture(0,cv.CAP_DSHOW) # 카메라와 연결 시도
6
7  if not cap.isOpened():
8      sys.exit('카메라 연결 실패')
9
10 frames=[]
11 while True:
12     ret,frame=cap.read()          # 비디오를 구성하는 프레임 획득
13
14     if not ret:
15         print('프레임 획득에 실패하여 루프를 나갑니다.')
16         break
17
18     cv.imshow('Video display',frame)
19
20     key=cv.waitKey(1) # 1밀리초 동안 키보드 입력 기다림
21     if key==ord('c'): # 'c' 키가 들어오면 프레임을 리스트에 추가
22         frames.append(frame)
23
24     elif key==ord('q'): # 'q' 키가 들어오면 루프를 빠져나감
25         break

```

```

27 cap.release()          # 카메라와 연결을 끊음
28 cv.destroyAllWindows()
29
30 if len(frames)>0:      # 수집된 영상이 있으면
31     imgs=frames[0]
32     for i in range(1,min(3,len(frames))): # 최대 3개까지 이어 붙임
33         imgs=np.hstack((imgs,frames[i]))
34
35     cv.imshow('collected images',imgs)
36
37     cv.waitKey()
38     cv.destroyAllWindows()
39
40 print(len(frames))
41 print(frames[0].shape)
42 print(type(imgs))
43 print(imgs.shape)

```

## 6. 비디오에서 수집한 영상 이어 붙이기

- numpy : np.hstack()를 사용하기 위한 라이브러리

```
1  import cv2 as cv
2  import numpy as np
3  import sys
4
5  cap=cv.VideoCapture(0,cv.CAP_DSHOW) # 카메라와 연결 시도
6
7  if not cap.isOpened():
8      sys.exit('카메라 연결 실패')
```

## 6. 비디오에서 수집한 영상 이어 붙이기

- frames : 이어 붙일 프레임들이 들어있는 리스트

```
10 frames=[]
11 while True:
12     ret,frame=cap.read()          # 비디오를 구성하는 프레임 획득
13
14     if not ret:
15         print('프레임 획득에 실패하여 루프를 나갑니다.')
16         break
17
18     cv.imshow('Video display',frame)
19
20     key=cv.waitKey(1)  # 1밀리초 동안 키보드 입력 기다림
21     if key==ord('c'):  # 'c' 키가 들어오면 프레임을 리스트에 추가
22         frames.append(frame)
23
24     elif key==ord('q'): # 'q' 키가 들어오면 루프를 빠져나감
25         break
```

## 6. 비디오에서 수집한 영상 이어 붙이기

- `np.hstack()` : 입력 배열들의 데이터를 가로로 연결

```
27 cap.release()          # 카메라와 연결을 끊음
28 cv.destroyAllWindows()
29
30 ✓ if len(frames)>0:      # 수집된 영상이 있으면
31     imgs=frames[0]
32 ✓     for i in range(1,min(3,len(frames))): # 최대 3개까지 이어 붙임
33         imgs=np.hstack((imgs,frames[i]))
34
35     cv.imshow('collected images',imgs)
36
37     cv.waitKey()
38     cv.destroyAllWindows()
```

## 6. 비디오에서 수집한 영상 이어 붙이기

```
40 print(len(frames))
41 print(frames[0].shape)
42 print(type(imgs))
43 print(imgs.shape)
```

```
In [1]: len(frames)
        7
In [2]: frames[0].shape
        (480,640,3)
In [3]: type(imgs)
        numpy.ndarray
In [4]: imgs.shape
        (480,1920,3)
```

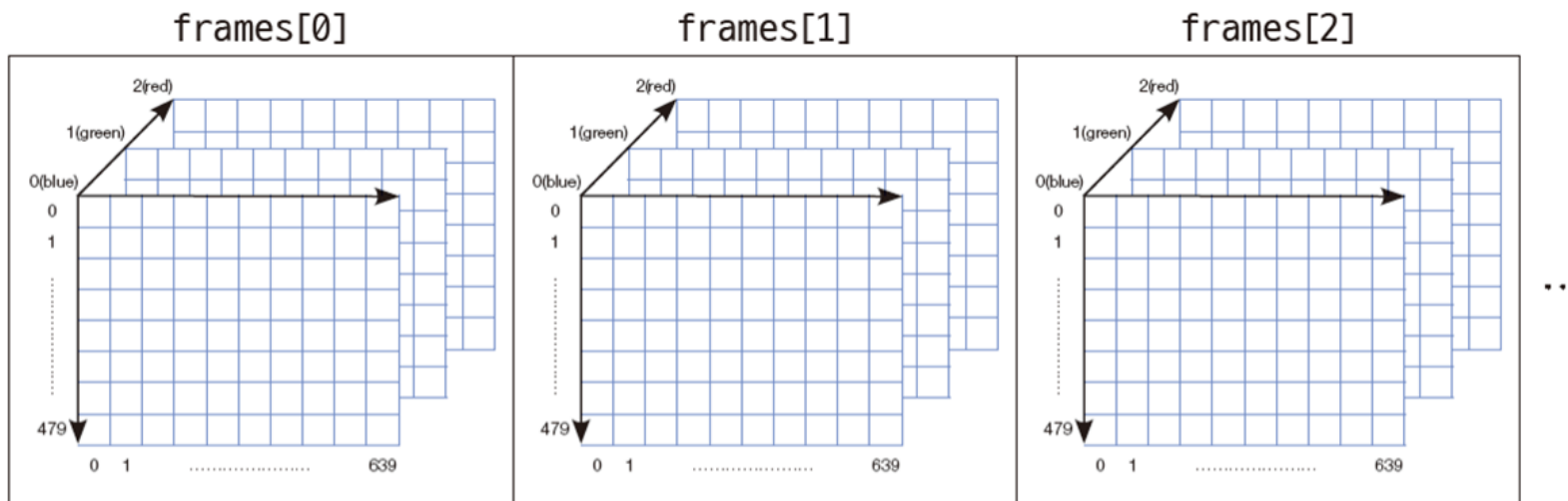


## 6. 비디오에서 수집한 영상 이어 붙이기

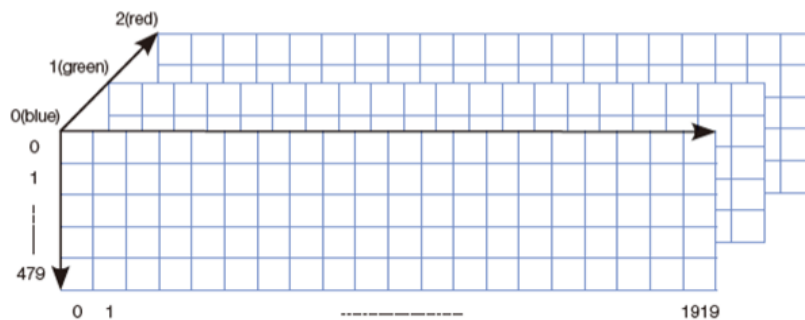


## 6. 비디오에서 수집한 영상 이어 붙이기

### ■ 영상 이어 붙이기 프로그램 자료 구조



(a) frames 리스트



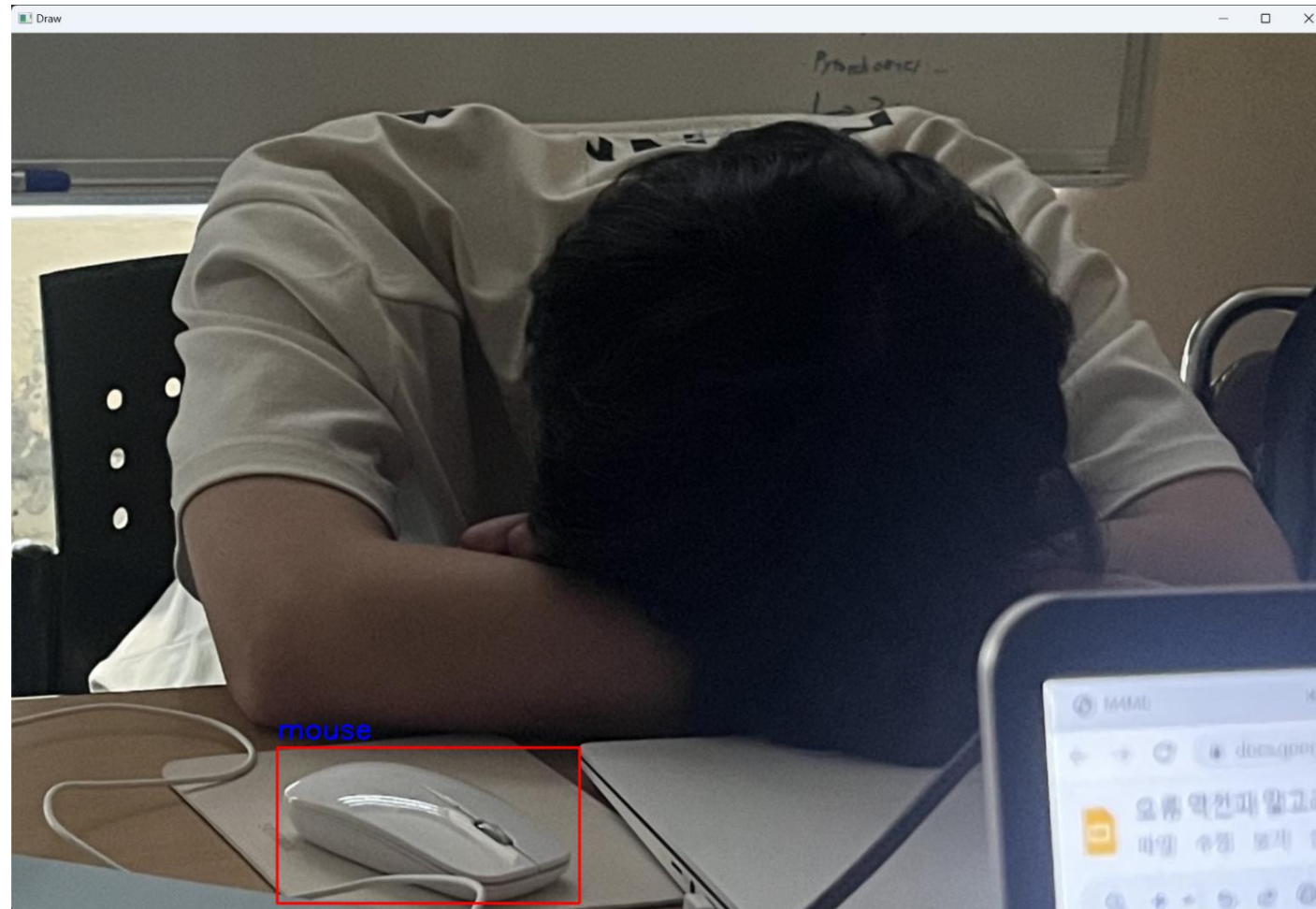
(b) imgs 배열

## 7. 영상에 도형을 그리고 글씨 쓰기

- `cv.rectangle(영상, 왼쪽 위 구석점 좌표, 오른쪽 밑 구석점 좌표, 색(B,G,E), 선의 두께)`
- `cv.putText(영상, 써넣을 문자열, 문자열 왼쪽 아래 구석점 좌표, 폰트 종류, 글자 크기, 글자 색, 글자 두께)`

```
1  import cv2 as cv
2  import sys
3
4  img=cv.imread('ch2\movemin.jpg')
5
6  if img is None:
7      sys.exit('파일을 찾을 수 없습니다.')
8
9  cv.rectangle(img,(290,780),(620,950),(0,0,255),2) # 직사각형 그리기
10 cv.putText(img,'mouse',(290,770),cv.FONT_HERSHEY_SIMPLEX,1,(255,0,0),2) # 글씨 쓰기
11
12 cv.imshow('Draw',img)
13
14 cv.waitKey()
15 cv.destroyAllWindows()
```

## 7. 영상에 도형을 그리고 글씨 쓰기



## 7. 영상에 도형을 그리고 글씨 쓰기

- 마우스로 클릭한 곳에 직사각형 그리기

```
1  import cv2 as cv
2  import sys
3
4  img=cv.imread('ch2\movemin.jpg')
5
6  if img is None:
7      sys.exit('파일을 찾을 수 없습니다.')
8
9  def draw(event,x,y,flags,param):          # 콜백 함수
10     if event==cv.EVENT_LBUTTONDOWN: # 마우스 왼쪽 버튼 클릭했을 때
11         cv.rectangle(img,(x,y),(x+200,y+200),(0,0,255),2)
12     elif event==cv.EVENT_RBUTTONDOWN: # 마우스 오른쪽 버튼 클릭했을 때
13         cv.rectangle(img,(x,y),(x+100,y+100),(255,0,0),2)
14
15     cv.imshow('Drawing',img)
16
17 cv.namedWindow('Drawing')
18 cv.imshow('Drawing',img)
19
20 cv.setMouseCallback('Drawing',draw) # Drawing 윈도우에 draw 콜백 함수 지정
21
22 while(True):          # 마우스 이벤트가 언제 발생할지 모르므로 무한 반복
23     if cv.waitKey(1)==ord('q'):
24         cv.destroyAllWindows()
25     break
```



## 7. 영상에 도형을 그리고 글씨 쓰기

- 마우스로 클릭한 곳에 직사각형 그리기

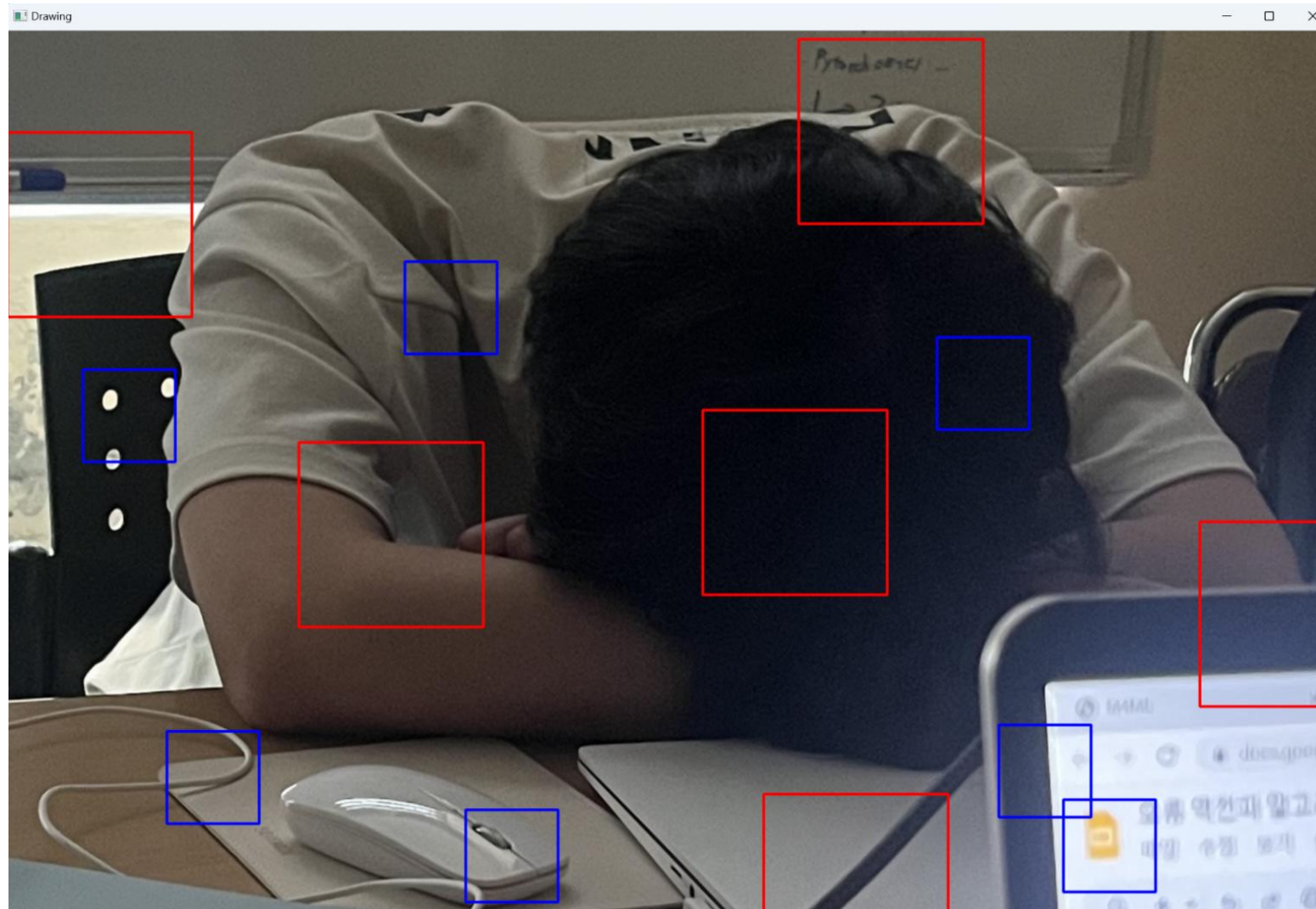
```
1  import cv2 as cv
2  import sys
3
4  img=cv.imread('ch2\movemin.jpg')
5
6  if img is None:
7      sys.exit('파일을 찾을 수 없습니다.')
8
9  def draw(event,x,y,flags,param):      # 콜백 함수
10     if event==cv.EVENT_LBUTTONDOWN: # 마우스 왼쪽 버튼 클릭했을 때
11         cv.rectangle(img,(x,y),(x+200,y+200),(0,0,255),2)
12     elif event==cv.EVENT_RBUTTONDOWN: # 마우스 오른쪽 버튼 클릭했을 때
13         cv.rectangle(img,(x,y),(x+100,y+100),(255,0,0),2)
14
15     cv.imshow('Drawing',img)
```

## 7. 영상에 도형을 그리고 글씨 쓰기

- `cv.setMouseCallback()` : 마우스를 다루는 프로그램에서는 클릭이나 커서 이동 같은 이벤트가 언제 발생할지 알 수 없기 때문에 콜백 함수가 필요함

```
17 cv.namedWindow('Drawing')
18 cv.imshow('Drawing',img)
19
20 cv.setMouseCallback('Drawing',draw) # Drawing 윈도우에 draw 콜백 함수 지정
21
22 while(True):          # 마우스 이벤트가 언제 발생할지 모르므로 무한 반복
23     if cv.waitKey(1)==ord('q'):
24         cv.destroyAllWindows()
25         break
```

## 7. 영상에 도형을 그리고 글씨 쓰기





## 8. 마우스 드래그로 도형 크기 조절하기

```
1  import cv2 as cv
2  import sys
3
4  img=cv.imread('ch2\movemin.jpg')
5
6  if img is None:
7      sys.exit('파일을 찾을 수 없습니다.')
8
9  def draw(event,x,y,flags,param):
10     global ix,iy
11
12     if event==cv.EVENT_LBUTTONDOWN: # 마우스 왼쪽 버튼 클릭했을 때 초기 위치 저장
13         ix,iy=x,y
14     elif event==cv.EVENT_LBUTTONUP: # 마우스 왼쪽 버튼 클릭했을 때 직사각형 그리기
15         cv.rectangle(img,(ix,iy),(x,y),(0,0,255),2)
16
17     cv.imshow('Drawing',img)
18
19 cv.namedWindow('Drawing')
20 cv.imshow('Drawing',img)
21
22 cv.setMouseCallback('Drawing',draw)
23
24 while(True):
25     if cv.waitKey(1)==ord('q'):
26         cv.destroyAllWindows()
27     break
```

## 8. 마우스 드래그로 도형 크기 조절하기

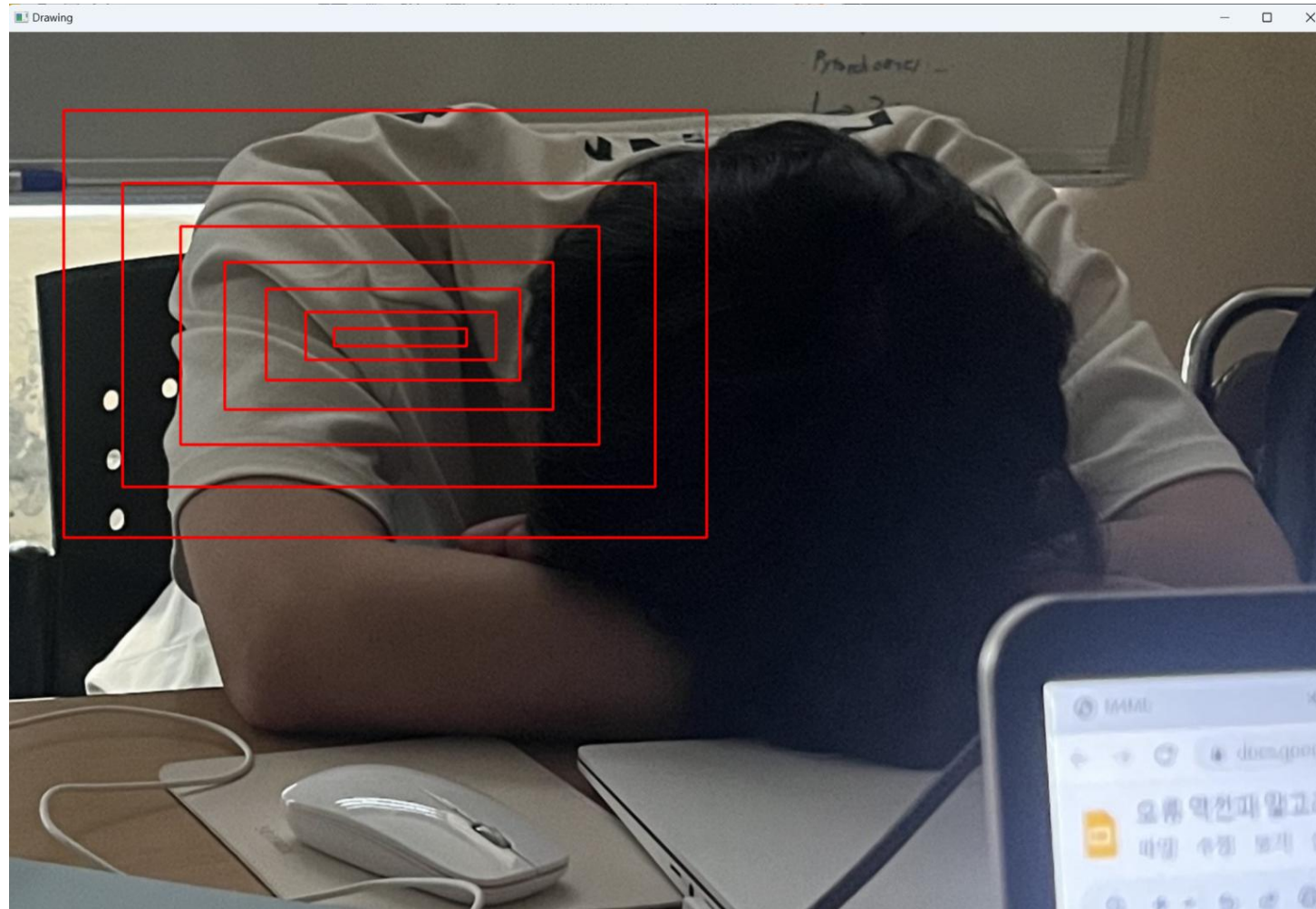
global ix, iy : 10행이 없으면 xi와 iy는 함수가 시작할 때 생겼다가 끝날 때 소멸하는 지역 변수로 작용해 드래그 동안 발생하는 여러번의 함수 호출에서 생성과 소멸을 반복하므로 좌푯값 유지 못함

```
1  import cv2 as cv
2  import sys
3
4  img=cv.imread('ch2\movemin.jpg')
5
6  if img is None:
7      sys.exit('파일을 찾을 수 없습니다.')
8
9  def draw(event,x,y,flags,param):
10     global ix,iy
11
12     if event==cv.EVENT_LBUTTONDOWN: # 마우스 왼쪽 버튼 클릭했을 때 초기 위치 저장
13         ix,iy=x,y
14     elif event==cv.EVENT_LBUTTONUP: # 마우스 왼쪽 버튼 클릭했을 때 직사각형 그리기
15         cv.rectangle(img,(ix,iy),(x,y),(0,0,255),2)
16
17     cv.imshow('Drawing',img)
```

## 8. 마우스 드래그로 도형 크기 조절하기

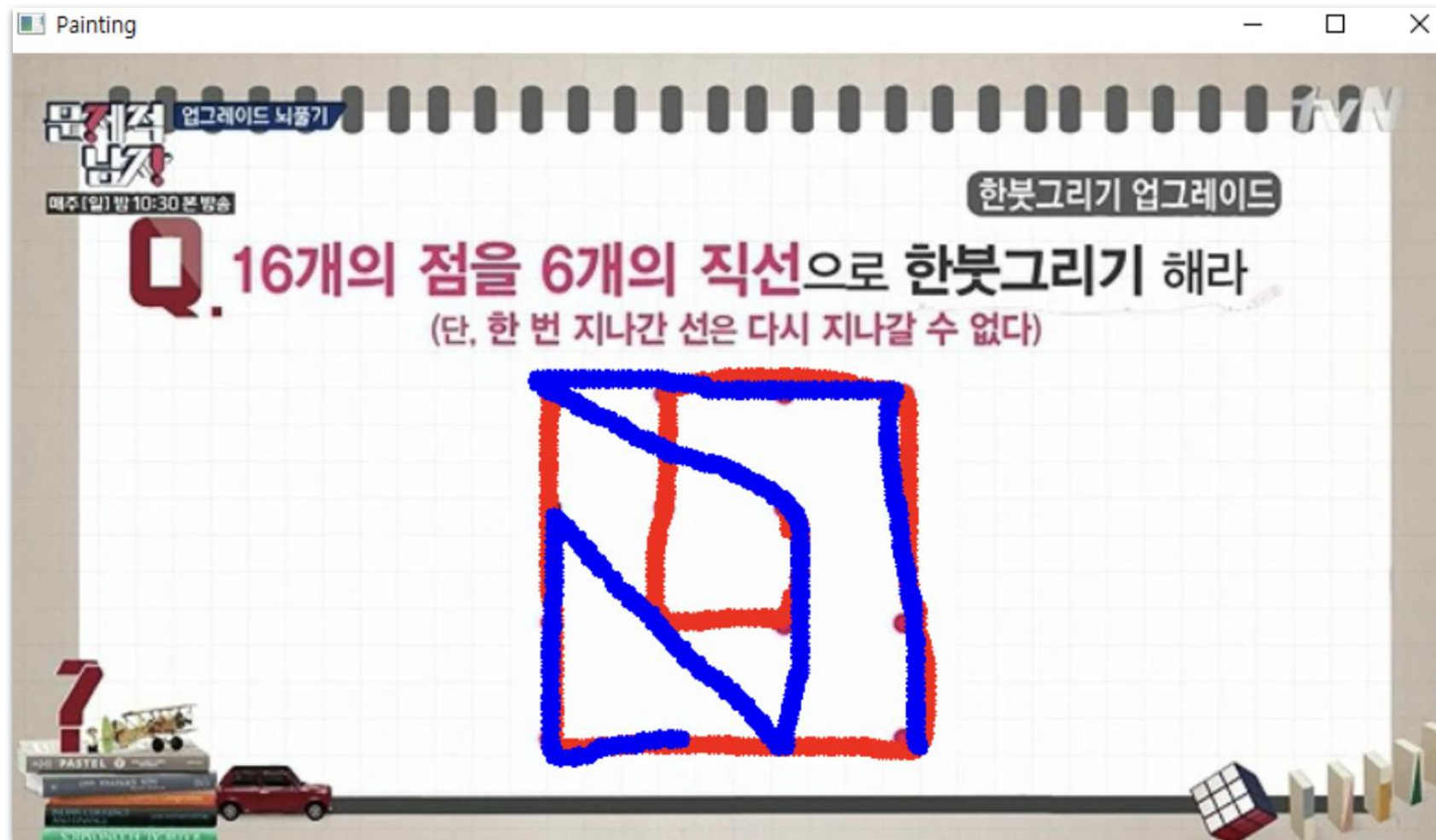
```
19 cv.namedWindow('Drawing')
20 cv.imshow('Drawing',img)
21
22 cv.setMouseCallback('Drawing',draw)
23
24 while(True):
25     if cv.waitKey(1)==ord('q'):
26         cv.destroyAllWindows()
27         break
```

## 8. 마우스 드래그로 도형 크기 조절하기



## 9. 페인팅 기능 만들기

- OpenCV의 이벤트 핸들링 기능을 사용하여 그림 그리기 코드를 작성함



## 9. 페인팅 기능 만들기

### ▪ HyperParameter 설정

- LColor : 마우스 왼쪽 키 이벤트 발생 시 색깔 - Red
- RColor : 마우스 오른쪽 키 이벤트 발생 시 색깔 - Blue



```
1  import cv2
2  import sys
3
4  BrushSize = 5
5  LColor, RColor = (255, 0, 0), (0, 0, 255)
```

## 9. 페인팅 기능 만들기

### ■ painting 함수 구축

- 각 이벤트별로 이벤트 발생 시 원을 그리는 형태로 구축
- flag 를 사용하여 마우스 클릭 유지 시 계속 원을 그리도록 구축



```
1  def painting(event, x, y, flags, param):
2      if event == cv2.EVENT_LBUTTONDOWN:
3          cv2.circle(img, (x, y), BrushSize, LColor, -1)
4      elif event == cv2.EVENT_RBUTTONDOWN:
5          cv2.circle(img, (x, y), BrushSize, RColor, -1)
6      elif event == cv2.EVENT_MOUSEMOVE and flags == cv2.EVENT_FLAG_LBUTTON:
7          cv2.circle(img, (x, y), BrushSize, LColor, -1)
8      elif event == cv2.EVENT_MOUSEMOVE and flags == cv2.EVENT_FLAG_RBUTTON:
9          cv2.circle(img, (x, y), BrushSize, RColor, -1)
10
11  cv2.imshow('Painting', img)
```



## 9. 페인팅 기능 만들기

### ■ 화면에 출력하기

- 미리 준비한 배경화면을 불러 화면에 출력
- 콜백 함수를 사용해 이벤트 발생 시에 특정 행동을 하도록 지정
  - 마우스 이벤트 발생 시 painting 함수를 불러오도록 지정



```
1  img = cv2.imread("paint.png")
2
3  if img is None:
4      sys.exit("File not exists.")
5
6  cv2.namedWindow('Painting')
7  cv2.imshow('Painting', img)
8  cv2.setMouseCallback('Painting', painting)
```



## 9. 페인팅 기능 만들기

### ■ 반복 출력

- while 문을 사용해 무한 반복을 함
- 사용자 입력 키가 'q'일 때 destroyAllWindows()를 사용하여 창이 닫히게 함
  - 만약 사용자가 입력하지 않고 UI 상에서 화면을 꺼버리면 강제종료를 해야 함



```
1  while(True):  
2      if cv2.waitKey(1) == ord('q'):  
3          cv2.destroyAllWindows()  
4          break
```



# 감사합니다.



**동아대학교**  
DONG-A UNIVERSITY

**M** Media for Machine  
Laboratory



**담당자**

서정일

**E-Mail**

jeongilseo@dau.ac.kr

**Homepage**

m4ml.re.kr