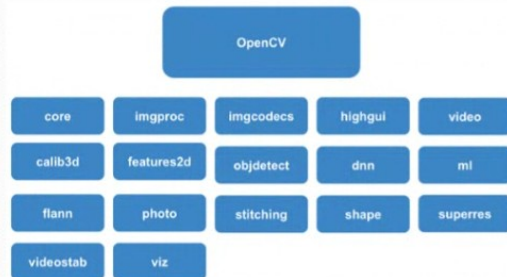


# 1. OpenCV 란?

## OpenCV Modules



### •Main modules:

- core. [Core functionality](#)
- imgproc. [Image Processing](#)
- imgcodecs. [Image file reading and writing](#)
- videoio. [Video I/O](#)
- highgui. [High-level GUI](#)
- video. [Video Analysis](#)
- calib3d. [Camera Calibration and 3D Reconstruction](#)
- features2d. [2D Features Framework](#)
- objdetect. [Object Detection](#)
- dnn. [Deep Neural Network module](#)
- ml. [Machine Learning](#)
- flann. [Clustering and Search in Multi-Dimensional Spaces](#)
- photo. [Computational Photography](#)
- stitching. [Images stitching](#)
- gapi. [Graph API](#)

### •Extra modules:

- alphamat. [Alpha Matting](#)
- aruco. [Aruco markers, module functionality was moved to objdetect module](#)
- bgsegm. [Improved Background-Foreground Segmentation Methods](#)
- bioinspired. [Biologically inspired vision models and derived tools](#)
- cgalib. [Custom Calibration Pattern for 3D reconstruction](#)
- cudaarithm. [Operations on Matrices](#)
- cudabgsegm. [Background Segmentation](#)
- cudacodec. [Video Encoding/Decoding](#)
- cudafeatures2d. [Feature Detection and Description](#)

⋮  
skip...

<https://docs.opencv.org/4.x/>

opencv: open-source computer vision  
컴퓨터 비전 관련 작업을 위한 라이브러리 집합

## 2. 개발 환경 설정 - Ubuntu

1. Ubuntu 22.04 사용 시 default 로 Python 3.10 이 기 설치 되어 있어 별도 Python 설치 필요없음.

2. 첫째 날 수업의 Ubuntu 설치 과정에서 이미 python 가상환경 설정을 완료했으므로 해당 환경 사용.

- Python3 가상환경 activation
- Python OpenCV 설치 (아래 2가지 중 택일)

```
(.env) $ pip install opencv-python # only contains main modules
```

```
(.env) $ pip install opencv-contrib-python # contains both main and contrib modules
```

- 설치 검증 - np.\_\_version\_\_ / cv2.\_\_version\_\_ 의 버전명이 출력되는지 확인한다.

```
(.env) $ python3
>>> import cv2
>>> import numpy as np
>>> print("cv2 ver:", cv2.__version__, "np ver:", np.__version__)
cv2 ver: 4.8.1, np ver: 1.26.1 # your version might be different
```

```
(.env) syc@syc-27V70Q-GA70K:~/opencv$ python3
Python 3.10.12 (main, May 27 2025, 17:12:29) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> import numpy as np
>>> print("cv2ver:", cv2.__version__, "np ver", np.__version__)
cv2ver: 4.11.0 np ver 2.2.6
```

## RGB Calculator



`rgb(255, 0, 0)`

`#ff0000`

`hsl(0, 100%, 50%)`

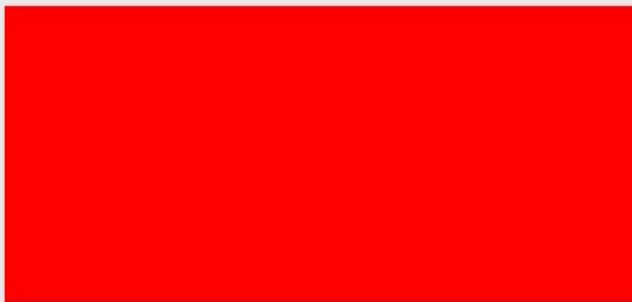
R:  

G:  

B:  

[Use this color in our Color Picker](#)

## HSL Calculator



`hsl(0, 100%, 50%)`

`rgb(255, 0, 0)`

`#ff0000`

H:  

S:  

L:  

[Use this color in our Color Picker](#)

=> HSL 색 검출에 용이함. H에 대해서만 if 문을 사용하면 되기 때문

### 3. 이미지 - Basic Operation

#### ▪ RGB/HSV Color Space (색 공간)

```
# 이미지 파일을 Read 하고 Color space 정보 출력
color = cv2.imread("strawberry.jpg", cv2.IMREAD_COLOR)
#color = cv2.imread("strawberry_dark.jpg", cv2.IMREAD_COLOR)
print(color.shape)

height,width,channels = color.shape
cv2.imshow("Original Image",color)

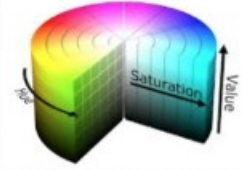
# Color channel 을 B,G,R 로 분할하여 출력
b,g,r = cv2.split(color)
rgb_split = np.concatenate((b,g,r),axis=1)
cv2.imshow("BGR Channels",rgb_split)

# 색공간을 BGR 에서 HSV 로 변환
hsv = cv2.cvtColor(color, cv2.COLOR_BGR2HSV)

# Channel 을 H,S,V 로 분할하여 출력
h,s,v = cv2.split(hsv)
hsv_split = np.concatenate((h,s,v),axis=1)
cv2.imshow("Split HSV",hsv_split)
```



RGB color space



HSV color space

#### Quiz

1. 위 색공간 이미지의 링크로 이동해서 각 색 공간의 표현 방법을 이해해 보자.
2. HSV color space 가 어떤 경우에 효과적으로 사용될까?
3. HSV 로 변환된 이미지를 BGR 이 아닌 RGB 로 다시 변환해서 출력해 보자.
4. COLOR\_RGB2GRAY 를 사용해서 흑백으로 변환해 출력해 보자.

원본사진



bgr



hsv





### 3. 이미지 - Basic Operation

#### ▪ Crop / Resize (자르기 / 크기 조정)

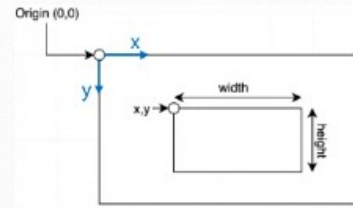
```
# 이미지 파일을 Read
img = cv2.imread("my_input.jpg")

# Crop 300x400 from original image from (100, 50)=(x,y)
cropped = img[50:450, 100:400]

# Resize cropped image from 300x400 to 400x200
resized = cv2.resize(cropped, (400,200))

# Display all
cv2.imshow("Original", img)
cv2.imshow("Cropped image", cropped)
cv2.imshow("Resized image", resized)

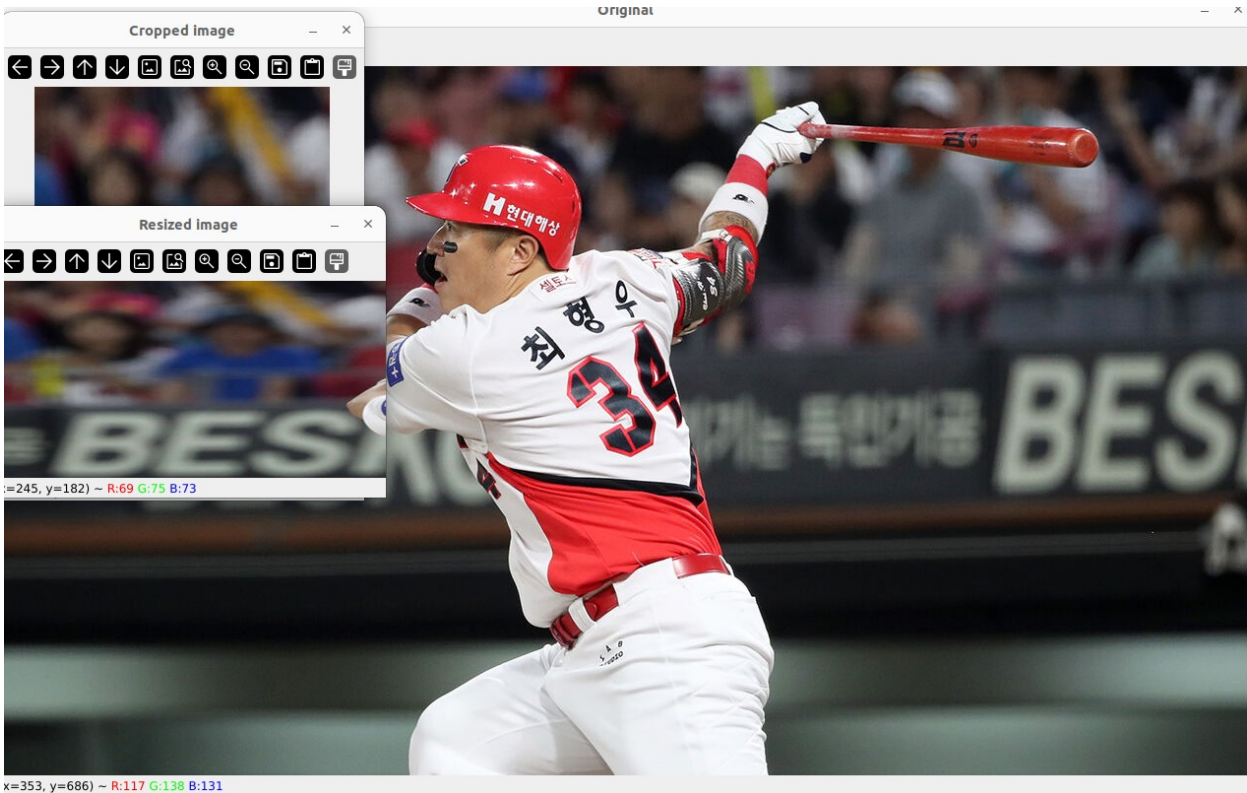
cv2.waitKey(0)
cv2.destroyAllWindows()
```



< OpenCV 에서 2D 이미지 좌표 개념 >

#### Quiz

1. Input image 를 본인이 좋아하는 인물 사진으로 변경해서 적용하자. 그리고 본인이 사용한 input image 의 size 를 확인해 보자.
2. 본인이 사용한 이미지의 얼굴 영역만 crop 해서 display 해 보자.
3. 원본 이미지의 정확히 1.5배만큼 이미지를 확대해서 파일로 저장해 보자.
4. openCV 의 rotate API 를 사용해서 우측으로 90도만큼 회전된 이미지를 출력해 보자.



### 3. 이미지 - Basic Operation

- 역상 (Reverse Image)

```
import cv2

src = cv2.imread("my_input.jpg",
cv2.IMREAD_COLOR)
dst = cv2.bitwise_not(src)

cv2.imshow("src", src)
cv2.imshow("dst", dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

## Quiz

1. AND, OR, XOR 연산에 대해서 확인해 보자.



src



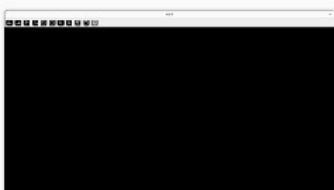
NOT



AND



OR



XOR

### 3. 이미지 - Basic Operation

- 이진화 (Binary)

```
import cv2

src = cv2.imread("my_input.jpg",
cv2.IMREAD_COLOR)

gray = cv2.cvtColor(src,
cv2.COLOR_BGR2GRAY)
ret, dst = cv2.threshold(gray, 100, 255,
cv2.THRESH_BINARY)

cv2.imshow("dst", dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

#### Quiz

1. 임계값을 변화시켜 보자.

-cv2.threshold(gray,100,255,cv2.THRESH\_BINARY)



-cv2.threshold(gray,100,200,cv2.THRESH\_BINARY)



### 3. 이미지 - Basic Operation

- 가장자리 검출 (Edge)

```
import cv2

src = cv2.imread("Image/wheat.jpg", cv2.IMREAD_COLOR)
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)

sobel = cv2.Sobel(gray, cv2.CV_8U, 1, 0, 3)

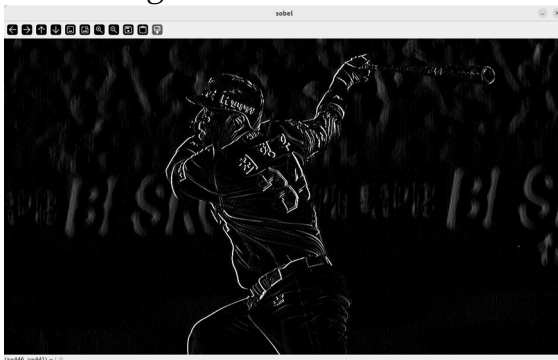
cv2.imshow("sobel", sobel)

cv2.waitKey()
cv2.destroyAllWindows()
```

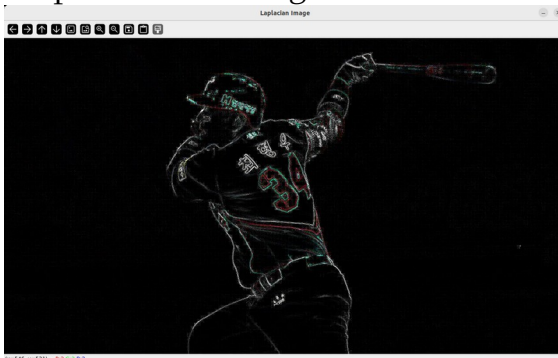
#### Quiz

1. Laplacian 변환을 적용해 보자.
2. Canny Edge Detection을 적용해 보자

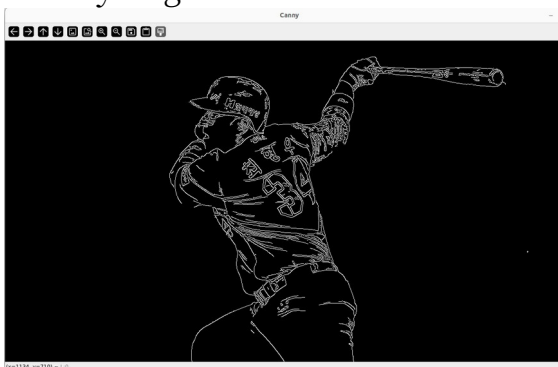
#### -Sobel edge detect



#### -Laplacian 변환 Edge detect



#### -Canny Edge Detection



=> 가장 뚜렷한 edge 검출



### 3. 이미지 - Basic Operation

- 배열 병합 (add Weighted)

```
import cv2

src = cv2.imread("tomato.jpg", cv2.IMREAD_COLOR)
hsv = cv2.cvtColor(src, cv2.COLOR_BGR2HSV)
h, s, v = cv2.split(hsv)

lower_red = cv2.inRange(hsv, (0, 100, 100), (5, 255, 255))
upper_red = cv2.inRange(hsv, (170, 100, 100), (180, 255, 255))
added_red = cv2.addWeighted(lower_red, 1.0, upper_red, 1.0, 0.0)

red = cv2.bitwise_and(hsv, hsv, mask = added_red)
red = cv2.cvtColor(red, cv2.COLOR_HSV2BGR)

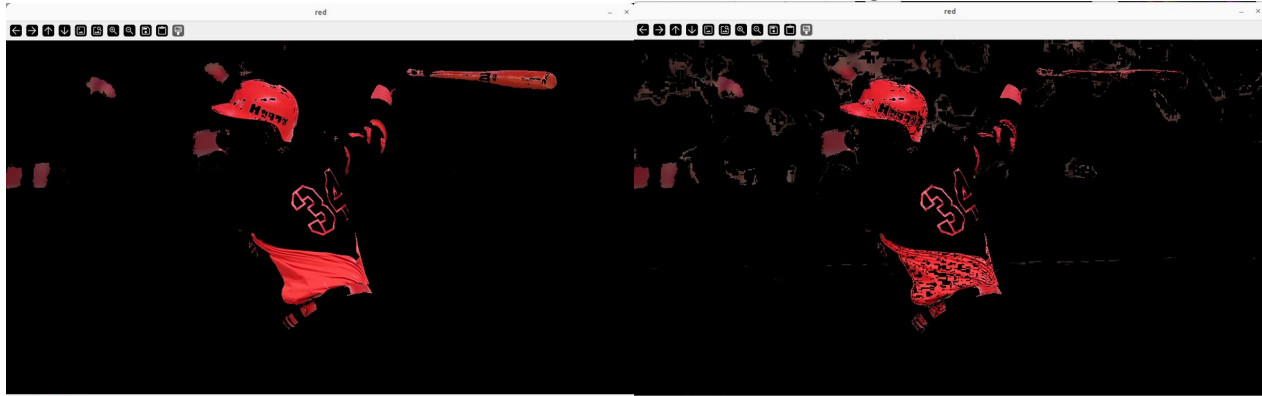
cv2.imshow("red", red)
cv2.waitKey()
cv2.destroyAllWindows()
```



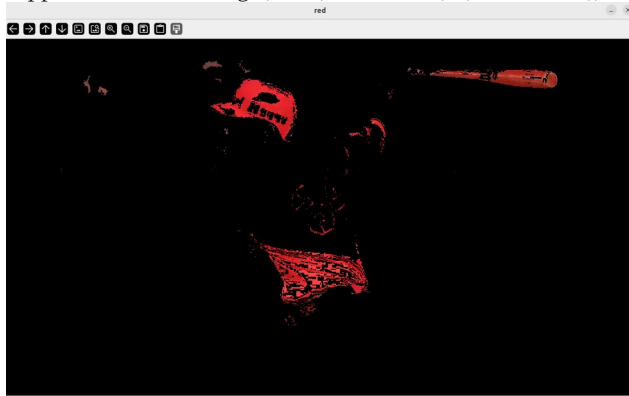
### Quiz

1. lower\_red 값의 범위를 변경해 보자.
2. upper\_red 값의 범위를 변경해 보자.
3. addWeighted의 gamma 값을 변경해 보자

-위의 코드 실행 결과



-upper\_red: cv2.inRange(hsv, (170,200,200), (180,255,255))





### 3. 이미지 - Basic Operation

- 채널 분리 및 병합

```
import cv2

src = cv2.imread("tomato.jpg",
cv2.IMREAD_COLOR)
b, g, r = cv2.split(src)
inverse = cv2.merge((r, g, b))

cv2.imshow("b", b)
cv2.imshow("g", g)
cv2.imshow("r", r)
cv2.imshow("inverse", inverse)
cv2.waitKey()
cv2.destroyAllWindows()
```



### Quiz

1. Numpy 형태의 채널 분리를 적용해 보자.

```
b = src[:, :, 0]
g = src[:, :, 1]
r = src[:, :, 2]
```

2. 빈 이미지를 적용해 보자.

```
height, width, channel = src.shape
zero = np.zeros((height, width, 1),
dtype=np.uint8)
bgz = cv2.merge((b, g, zero))
```

-inverse



-r



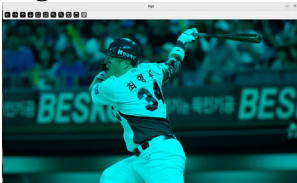
-g



-b



-bgz



## 4. 동영상 - Basic Operation

- 동영상 파일을 읽고 보여주기

```
import numpy as np
import cv2

# Read from the recorded video file
cap = cv2.VideoCapture("ronaldinho.mp4")

# 동영상 파일이 성공적으로 열렸으면 while 문 반복
while(cap.isOpened()):
    # 한 프레임을 읽어옴
    ret, frame = cap.read()

    if ret is False:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    # Display
    cv2.imshow("Frame", frame)

    # 1 ms 동안 대기하며 키 입력을 받고 'q' 입력 시 종료
    key = cv2.waitKey(1)
    if key & 0xFF == ord('q'):
        break

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

### Quiz



- 동영상이 너무 빠르게 재생된다. 이유를 찾아보고 정상적인 속도로 재생될 수 있도록 수정해 보자.
- 동영상이 끝까지 재생되면 더이상 frame 을 읽어오지 못해 종료된다. 동영상이 끝까지 재생되면 다시 처음부터 반복 될 수 있도록 수정해 보자.
- 동영상 크기를 반으로 resize 해서 출력해 보자.
- 동영상 재생 중 'c' 키 입력을 받으면 해당 프레임을 이미지 파일로 저장하게 코드를 수정해 보자. 파일 이름은 001.jpg, 002.jpg 등으로 overwrite 되지 않게 하자.

=> 문제점: 동영상이 빠르게 재생된 후, 종료. 해결방안: cv2.waitKey(55)로 변경



=> 'c'를 입력한 후 capture 된 사진. 파일명도 capture\_0.jpg 로 의도한 대로 동작한 것을 확인.

## 5. 카메라 - Basic Operation

- 카메라로부터 input 을 받아 보여주고 동영상 파일로 저장하기

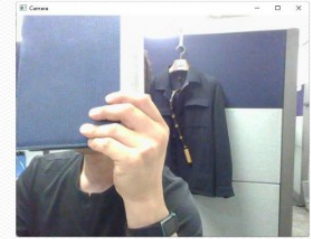
```
# Read from the first camera device
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)

w = 640#1280#1920
h = 480#720#1080
cap.set(cv2.CAP_PROP_FRAME_WIDTH, w)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, h)

# 성공적으로 video device 가 열렸으면 while 문 반복
while(cap.isOpened()):
    # 한 프레임을 읽어옴
    ret, frame = cap.read()
    if ret is False:
        print("Can't receive frame (stream end?). Exiting ...")
        break








    # Display
    cv2.imshow("Camera", frame)

    # 1 ms 동안 대기하며 키 입력을 받고 'q' 입력 시 종료
    key = cv2.waitKey(1)
    if key & 0xFF == ord('q'):
        break
```



### Quiz

- 가지고 있는 카메라의 지원 가능한 해상도를 확인 후 카메라 해상도를 변경해 보자.
- 카메라 Input 을 "output.mp4" 동영상 파일로 저장하도록 코드를 추가해 보자.

	250634_color_space.py	755 바이트	13 : 59	☆
	201796967_1280.jpg	120.7 kB	12 : 29	☆
	Garden_strawberry_(Fragaria_x_ananassa)_single2.jpg	25.3 kB	11 : 52	☆
	U44VTm_3QB4enZmdd_QJhHQeiBB5y12O7Loms5yLitBXzw8...	79.4 kB	12 : 12	☆
	basic_opeartion.py	518 바이트	11 : 55	☆
	output.mp4	169.3 kB	18 : 11	☆
	son.mp4	"output.mp4"을(를) 선택했습니다 (169.3 kB)		

=> Quiz 2



## 6. OpenCV 에서 제공하는 GUI Input 활용하기

### ▪ Text / Line / Rectangle

```
# Read from the first camera device
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)

topLeft = (50, 50)
bottomRight = (300, 300)

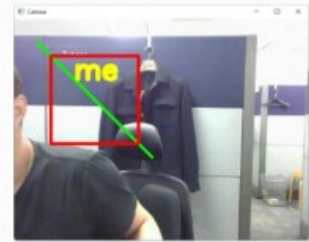
# 성공적으로 video device 가 열렸으면 while 문 반복
while(cap.isOpened()):
    # 한 프레임을 읽어옴
    ret, frame = cap.read()

    # Line
    cv2.line(frame, topLeft, bottomRight, (0, 255, 0), 5)

    # Rectangle
    cv2.rectangle(frame,
                  [pt+30 for pt in topLeft], [pt-30 for pt in bottomRight], (0, 0, 255), 5)

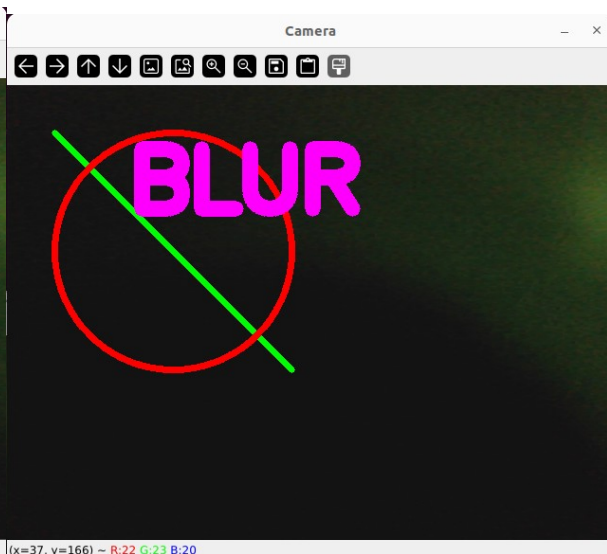
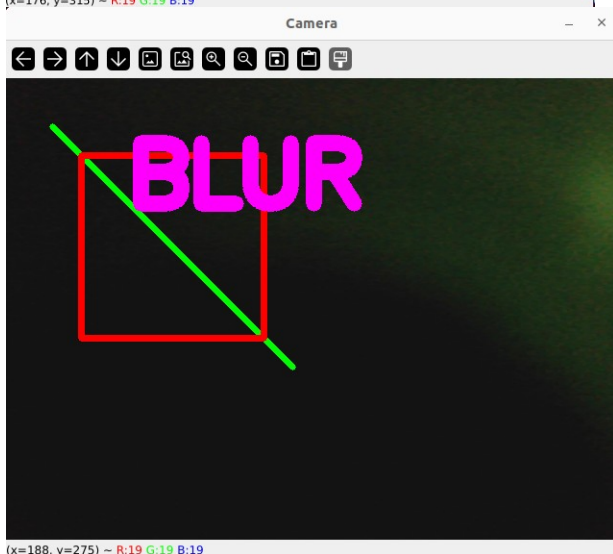
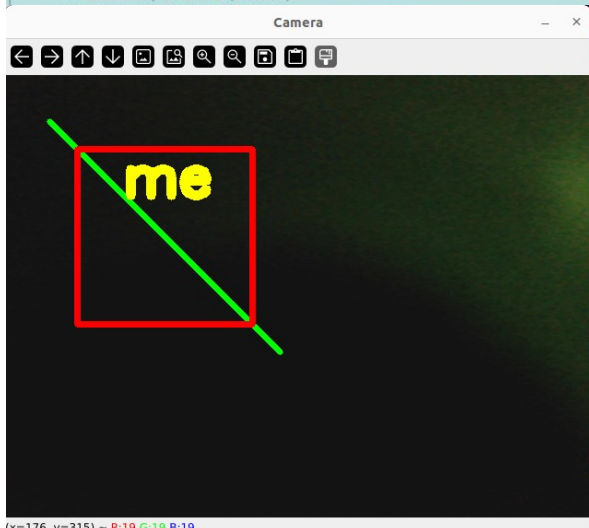
    # Text
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(frame, 'me',
                [pt+80 for pt in topLeft], font, 2, (0, 255, 255), 10)

    # Display
    cv2.imshow("Camera", frame)
```



### Quiz

1. Text 문구 / Font / 색상 / 크기 / 굵기 / 출력위치 등 모든 값을 변경해 보자.
2. 동그라미를 그리는 함수를 찾아서 적용해 보자.
3. 마우스 왼쪽 버튼을 click 하면 해당 위치에 동그라미가 그려지도록 코드를 추가해 보자.  
(Reference : cv2.EVENT\_LBUTTONDOWN)



=> Quiz 1, 2 해



```

import cv2

# 마우스 클릭 위치를 저장할 리스트
circle_centers = []

# 마우스 클릭 이벤트 처리 함수
def mouse_callback(event, x, y, flags, param):
    global circle_centers
    if event == cv2.EVENT_LBUTTONDOWN: # 왼쪽 버튼 클릭 시
        circle_centers.append((x, y)) # 클릭한 위치를 리스트에 추가

# 비디오 캡처 설정
cap = cv2.VideoCapture(0)

# 좌표 설정
topLeft = (50, 50)
bottomRight = (300, 300)

# 마우스 콜백 함수 등록
cv2.namedWindow("Camera")
cv2.setMouseCallback("Camera", mouse_callback)

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

    # 선 그리기
    cv2.line(frame, topLeft, bottomRight, (0, 255, 0), 5)

    # 직사각형 대신 동그라미 그리기
    center = ((topLeft[0] + bottomRight[0]) // 2, (topLeft[1] + bottomRight[1]) // 2)
    radius = min(bottomRight[0] - topLeft[0], bottomRight[1] - topLeft[1]) // 2
    cv2.circle(frame, center, radius, (0, 0, 255), 5)

    # 텍스트 추가
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(frame, 'BLUR', [pt + 80 for pt in topLeft], font, 3, (255, 0, 255), 15)

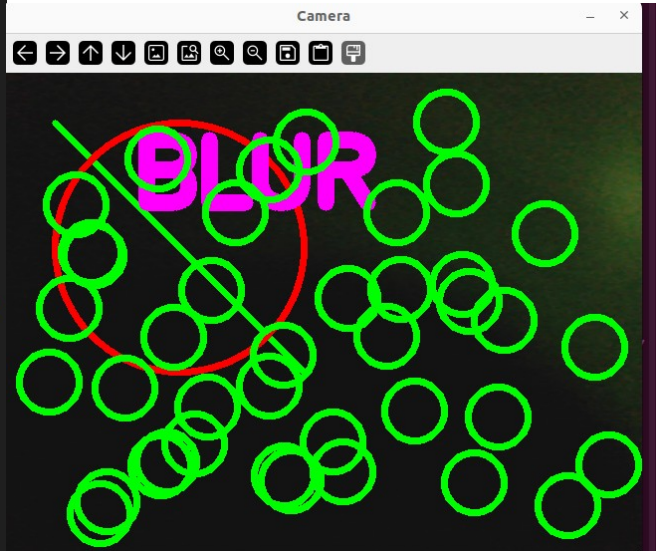
    # 마우스 클릭으로 추가된 동그라미 그리기
    for (cx, cy) in circle_centers:
        cv2.circle(frame, (cx, cy), 30, (0, 255, 0), 5) # 클릭한 위치에 동그라미

    # 비디오 창에 출력
    cv2.imshow("Camera", frame)

    # 종료 조건
    key = cv2.waitKey(30) & 0xFF
    if key == ord("q"):
        break

cap.release()
cv2.destroyAllWindows()

```



=> Quiz3 해결

## 6. OpenCV 에서 제공하는 GUI Input 활용하기

### ▪ Tracker

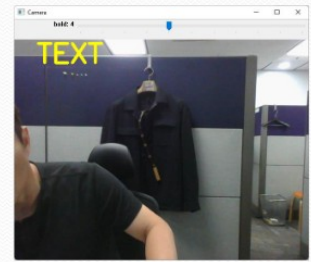
```
topLeft = (50, 50)
bold = 0
# Callback function for the tracker
def on_bold_tracker(value):
    #print("Tracker value:", value)
    global bold
    bold = value

cv2.namedWindow("Camera")
cv2.createTracker("bold", "Camera", bold, 10, on_bold_tracker)

# 성공적으로 video device 가 열렸으면 while 문 반복
while(cap.isOpened()):
    # 한 프레임을 읽어옴
    ret, frame = cap.read()
    if ret is False:
        print("Can't receive frame (stream end?). Exiting ...")
        break

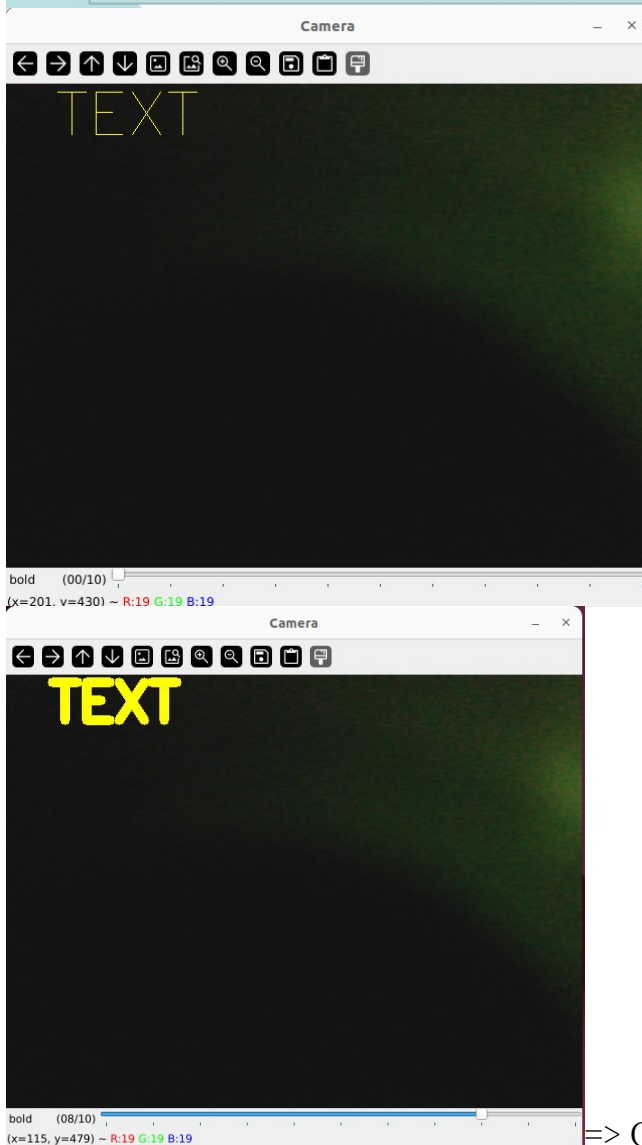
    # Text
    cv2.putText(frame, "TEXT",
                topLeft, cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 255, 255), 1 + bold)

    # Display
    cv2.imshow("Camera", frame)
```

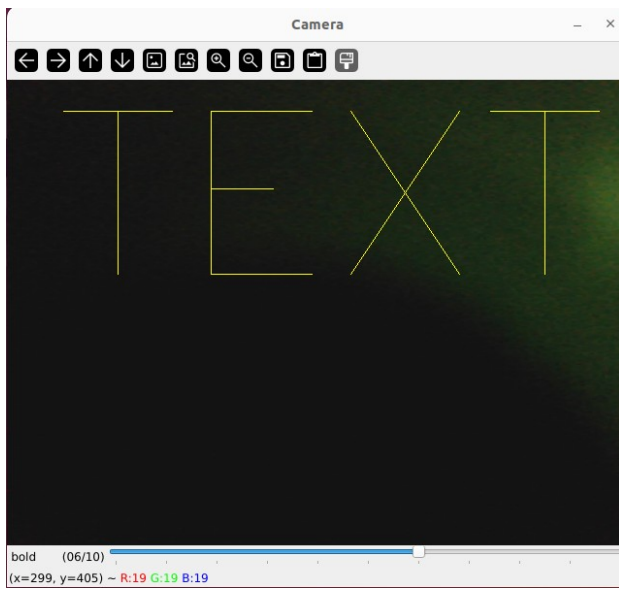


### Quiz

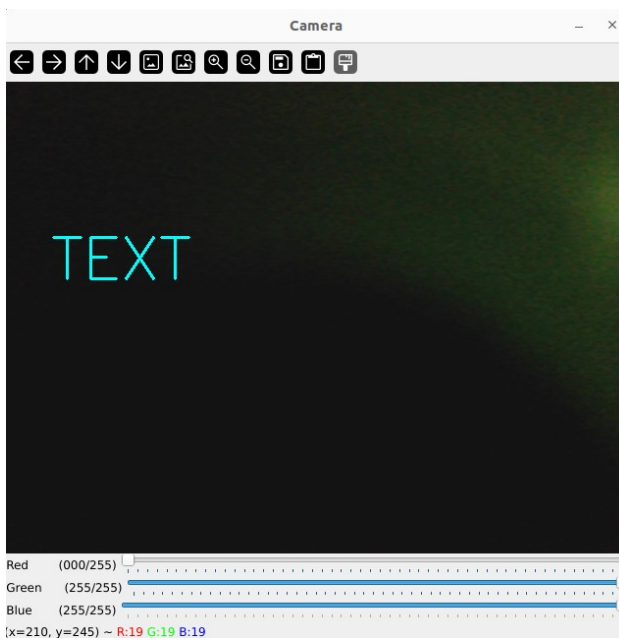
1. Tracker 를 control 해서 TEXT 의 굵기가 변하는 것을 확인해 보자.
2. Tracker 를 추가해서 font size 를 변경 / 적용해 보자.
3. R/G/B Tracker 를 각각 추가해서 글자의 font color 를 변경해 보자.



=> Quiz 1 해결



=>Quiz 2 해결



=>Quiz 3 해결

