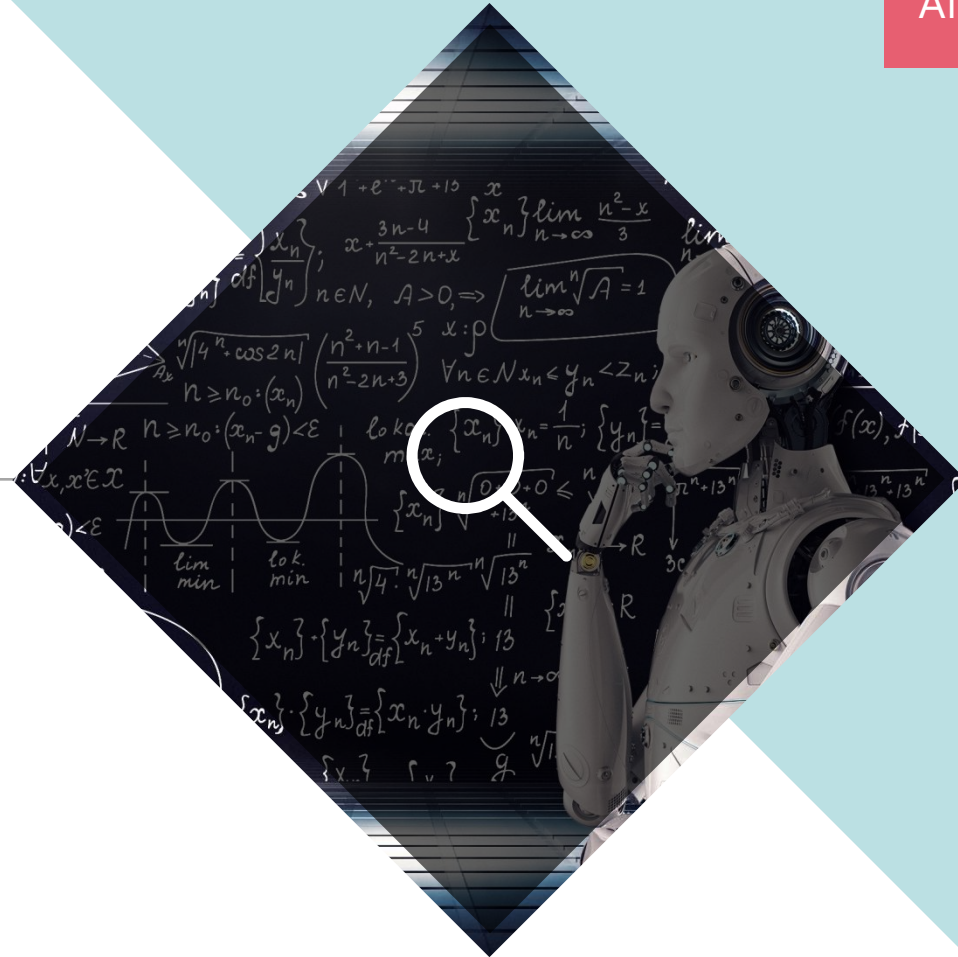


Day 09

ROS & Image Processing



Contents

I. Color Space (RGB/HSV/LAB)

II. Color Filter 구현하기

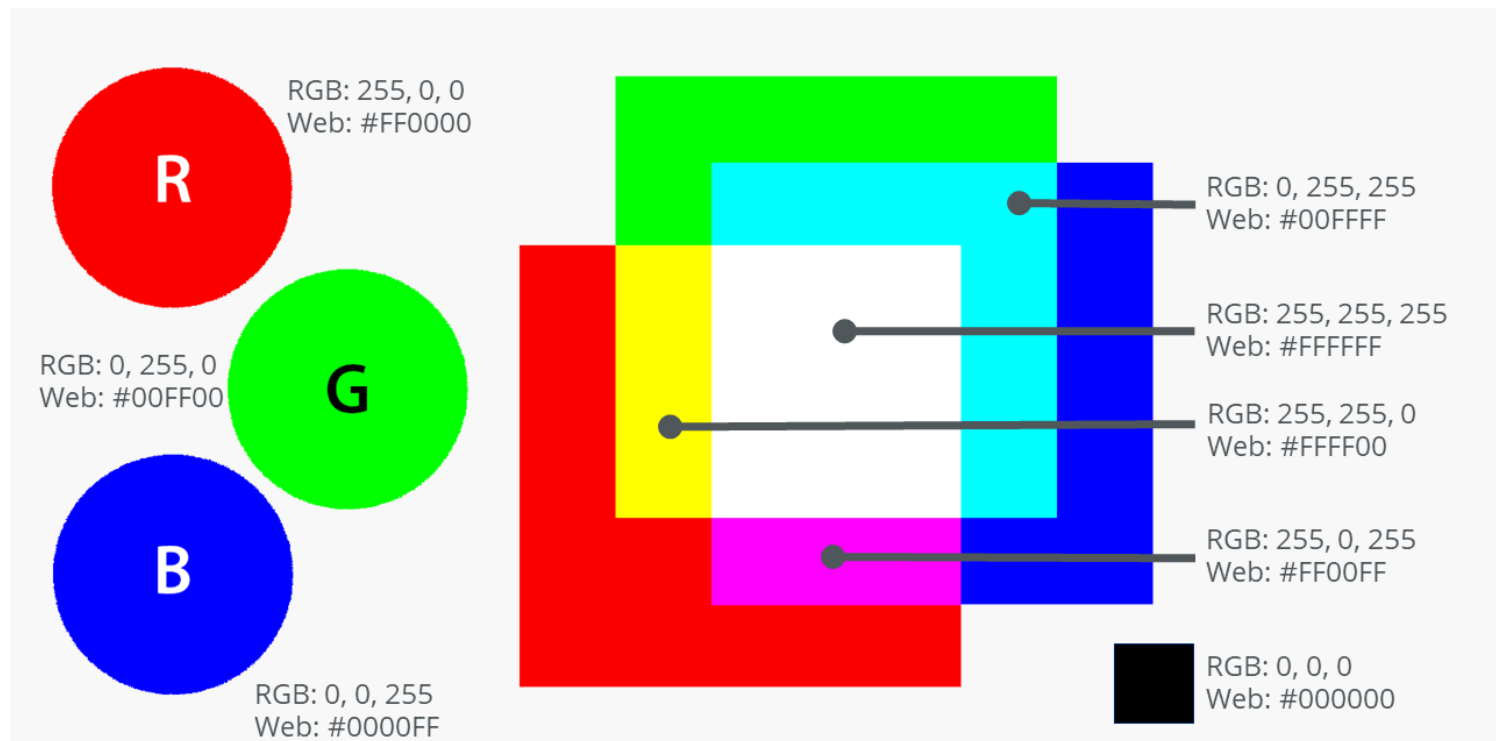
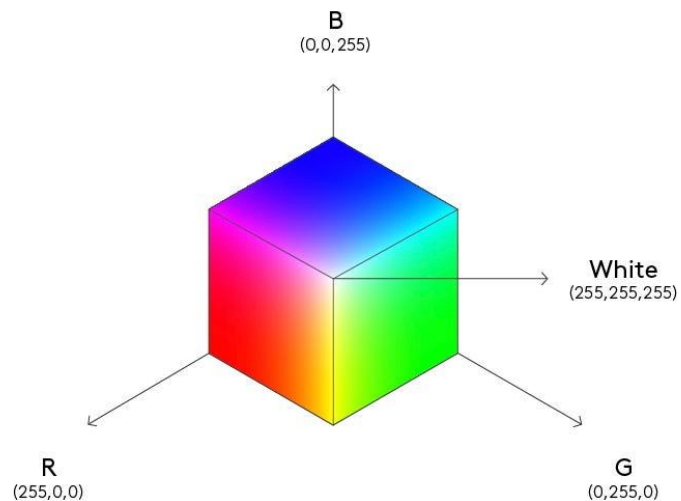
III. Tuning 한 Threshold 값 Save / Load

IV. Color Object Tracking 구현하기

I . Color Space

RGB Color Space

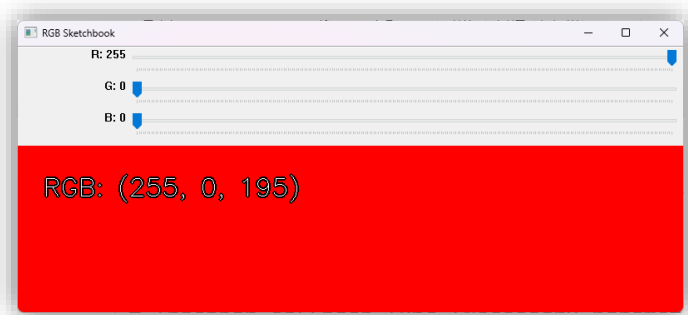
- 빨간색(Red), 초록색(Green), 파란색(Blue) 세 가지 색의 빛을 섞어서 다양한 색을 만드는 방식
- 각 색(R, G, B)은 0~255까지 밝기를 조절



I . Color Space

RGB Color Space (cont.)

- Tracker 를 사용해 R,G,B 값을 control 하여
변하는 RGB color 값을 확인하는 예제 코드



```
img[:] = [b, g, r]
```

- `img[:]` 이미지 전체 픽셀 (모든 행, 모든 열)
- `[b, g, r]` 각 픽셀의 색상 값을 BGR 순서로 지정

```
import cv2
import numpy as np

# Tracker callback func (Necessary parameter - do nothing here)
def nothing(x):
    pass

# 800x200 빈 스케치북 생성
img = np.zeros((200, 800, 3), np.uint8)

cv2.namedWindow('RGB Sketchbook')
cv2.createTrackbar('R', 'RGB Sketchbook', 0, 255, nothing)
cv2.createTrackbar('G', 'RGB Sketchbook', 0, 255, nothing)
cv2.createTrackbar('B', 'RGB Sketchbook', 0, 255, nothing)

while True:
    r = cv2.getTrackbarPos('R', 'RGB Sketchbook')
    g = cv2.getTrackbarPos('G', 'RGB Sketchbook')
    b = cv2.getTrackbarPos('B', 'RGB Sketchbook')
    img[:] = [b, g, r] # OpenCV는 BGR 순서

    cv2.imshow('RGB Sketchbook', img)

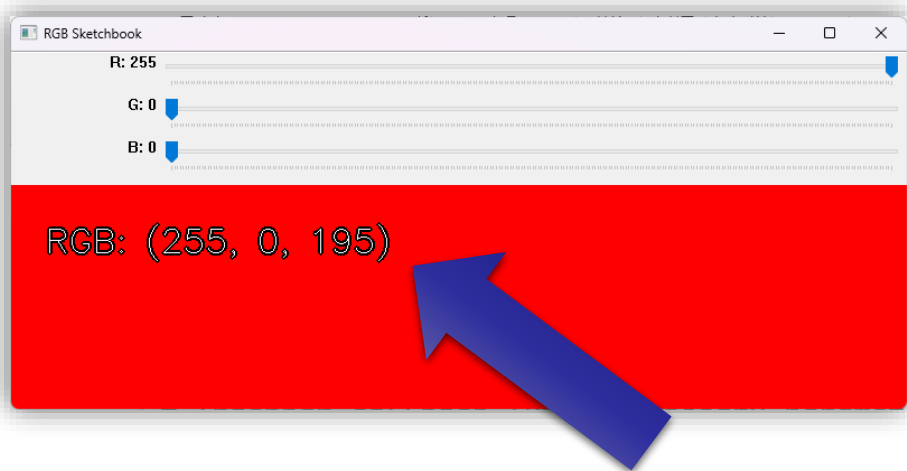
    if cv2.waitKey(1) & 0xFF == 27: # ESC 키로 종료
        break

cv2.destroyAllWindows()
```

I . Color Space

RGB Color Space (cont.)

- cv2.putText 함수를 2번 사용하여 RGB 값을 아래처럼 text 로 출력해 주도록 구현하기



```
import cv2
import numpy as np

# Tracker callback func (Necessary parameter - do nothing here)
def nothing(x):
    pass

# 800x200 빈 스케치북 생성
img = np.zeros((200, 800, 3), np.uint8)

cv2.namedWindow('RGB Sketchbook')
cv2.createTrackbar('R', 'RGB Sketchbook', 0, 255, nothing)
cv2.createTrackbar('G', 'RGB Sketchbook', 0, 255, nothing)
cv2.createTrackbar('B', 'RGB Sketchbook', 0, 255, nothing)

while True:
    r = cv2.getTrackbarPos('R', 'RGB Sketchbook')
    g = cv2.getTrackbarPos('G', 'RGB Sketchbook')
    b = cv2.getTrackbarPos('B', 'RGB Sketchbook')
    img[:] = [b, g, r] # OpenCV는 BGR 순서

    # 텍스트 표시
    text_rgb =

    # 텍스트 배경 처리용 (검정 테두리)
    cv2.putText
    cv2.putText

    cv2.imshow('RGB Sketchbook', img)

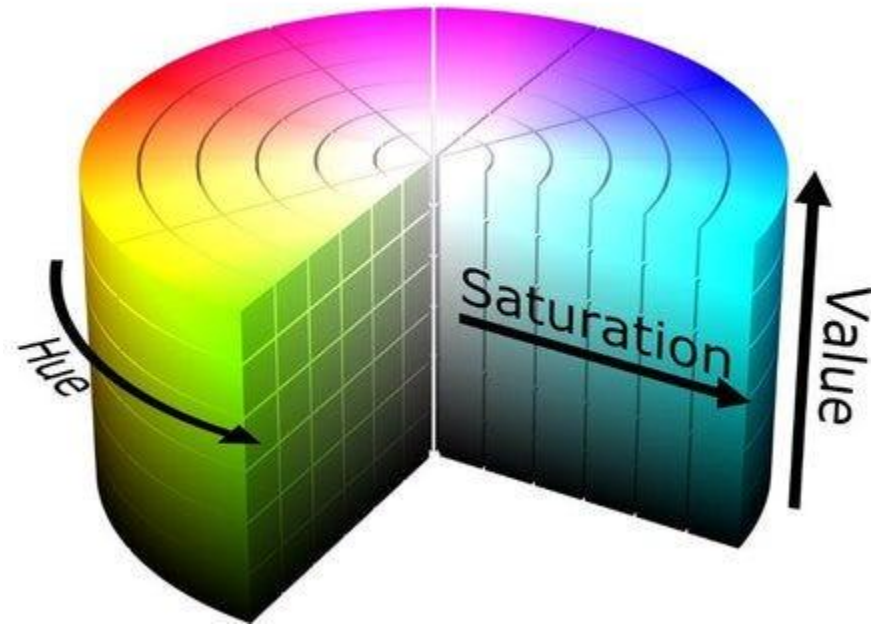
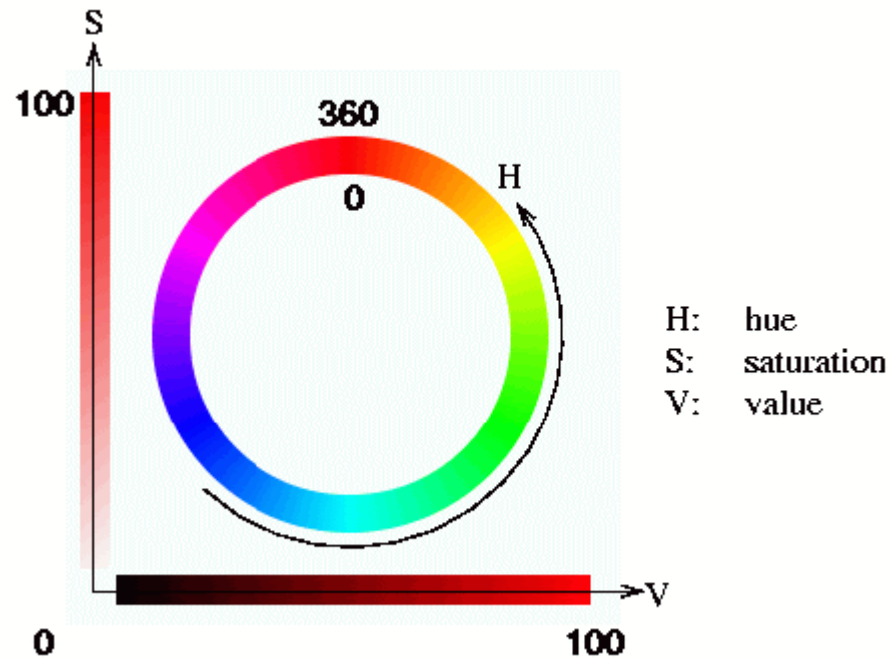
    if cv2.waitKey(1) & 0xFF == 27: # ESC 키로 종료
        break

cv2.destroyAllWindows()
```

I . Color Space

HSV Color Space

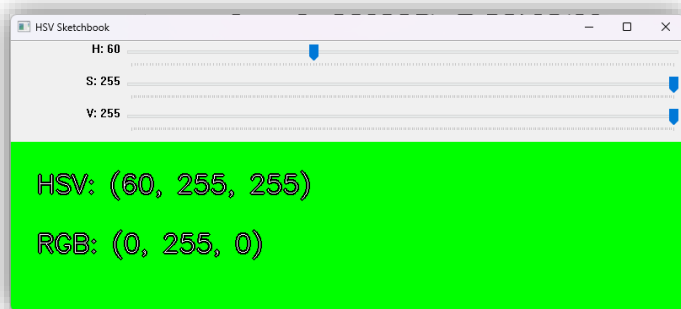
- HSV는 색깔을 사람의 감각처럼 표현한 색 모델
- H (Hue, 색조), (Saturation, 채도), V (Value, 명도)



I . Color Space

HSV Color Space (cont.)

- Tracker 를 사용해 H,S,V 값을 control 하여 변하는 HSV color 값을 확인하는 예제 코드
- H 값은 0 ~ 359 까지로 표현되나 OpenCV 에서는 표현 범위의 한계(8bit: 0 - 255) 로 인해 H 값만 2로 나눈 값을 사용함 ($0 \sim 359 / 2 = 0 \sim 179$)



OpenCV 에서 기본적으로 사용하는 색공간은 BGR 이기 때문에 Display 출력을 위해 HSV 를 BGR 로 변경이 필요함

```
import cv2
import numpy as np

def nothing(x): # Tracker callback func (Necessary parameter - do nothing here)
    pass

# Create a blank image (sketchbook)
img = np.zeros((200, 800, 3), np.uint8)

# Create HSV Sketchbook window and add trackers
cv2.namedWindow('HSV Sketchbook')
cv2.createTrackbar('H', 'HSV Sketchbook', 0, 179, nothing) # Hue: 0~179 (OpenCV range)
cv2.createTrackbar('S', 'HSV Sketchbook', 0, 255, nothing) # Saturation: 0~255
cv2.createTrackbar('V', 'HSV Sketchbook', 0, 255, nothing) # Value: 0~255

while True:
    # Get current tracker values
    h = cv2.getTrackbarPos('H', 'HSV Sketchbook')
    s = cv2.getTrackbarPos('S', 'HSV Sketchbook')
    v = cv2.getTrackbarPos('V', 'HSV Sketchbook')

    # Convert HSV to BGR (OpenCV uses BGR)
    hsv_color = np.uint8([[h, s, v]])
    b, g, r = cv2.cvtColor(hsv_color, cv2.COLOR_HSV2BGR)[0][0]

    # Apply background color
    img[:] = [b, g, r]

    # Show image
    cv2.imshow('HSV Sketchbook', img)

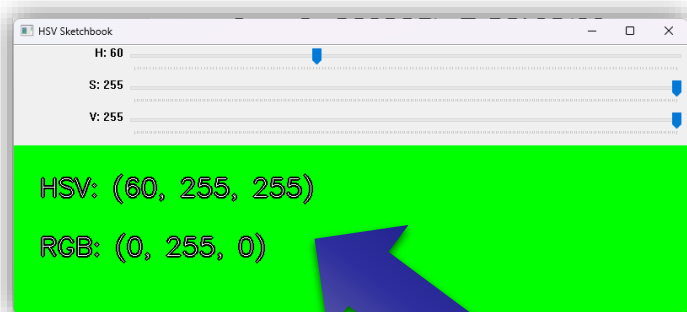
    if cv2.waitKey(1) & 0xFF == 27:
        break

cv2.destroyAllWindows()
```

I . Color Space

HSV Color Space (cont.)

- cv2.putText 함수를 사용하여 RGB 값과 HSV 값을 아래처럼 text 로 출력해 주도록 구현하기



```
import cv2
import numpy as np
def nothing(x):
    pass
img = np.zeros((200, 800, 3), np.uint8)
cv2.namedWindow('HSV Sketchbook')
cv2.createTrackbar('H', 'HSV Sketchbook', 0, 179, nothing) # Hue: 0~179 (OpenCV range)
cv2.createTrackbar('S', 'HSV Sketchbook', 0, 255, nothing) # Saturation: 0~255
cv2.createTrackbar('V', 'HSV Sketchbook', 0, 255, nothing) # Value: 0~255

while True:
    # Get trackbar values
    h = cv2.getTrackbarPos('H', 'HSV Sketchbook')
    s = cv2.getTrackbarPos('S', 'HSV Sketchbook')
    v = cv2.getTrackbarPos('V', 'HSV Sketchbook')

    # Convert HSV to BGR (OpenCV uses BGR)
    hsv_color = np.uint8([[h, s, v]])
    b, g, r = cv2.cvtColor(hsv_color, cv2.COLOR_HSV2BGR)[0][0]

    # Apply background color
    img[:] = [b, g, r]

    # Prepare text
    text_hsv = 
    text_rgb = 

    # Draw text (black border + white text)
    cv2.putText(img, text_hsv, 
    cv2.putText(img, text_hsv, 

    cv2.putText(img, text_rgb, 
    cv2.putText(img, text_rgb, 

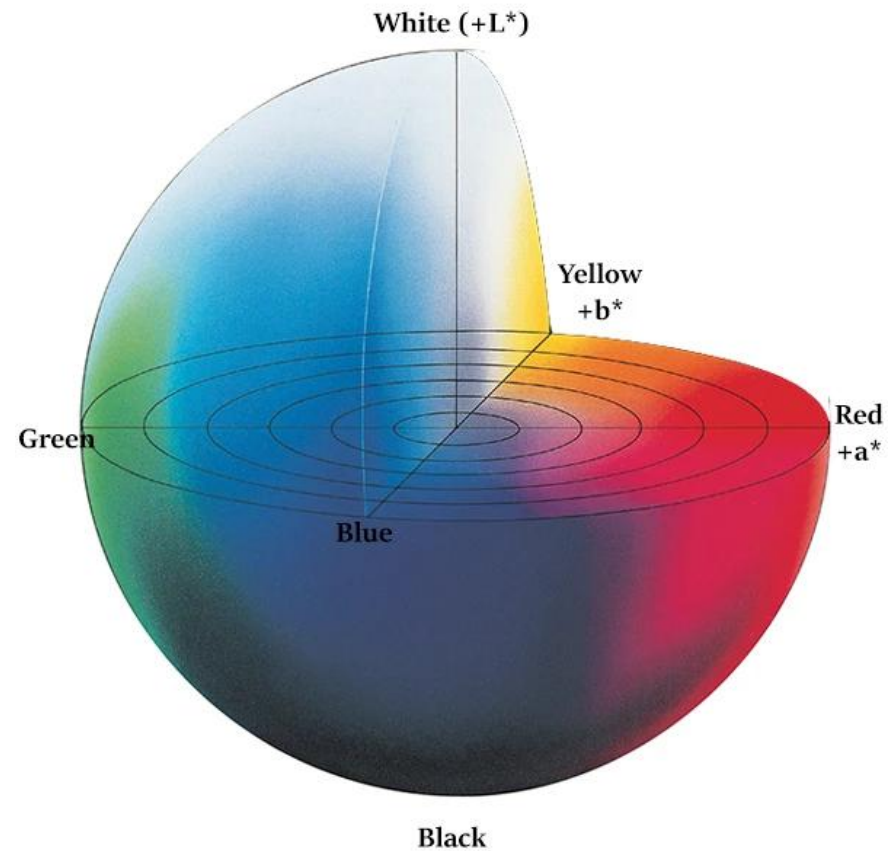
    cv2.imshow('HSV Sketchbook', img)
    if cv2.waitKey(1) & 0xFF == 27:
        break

cv2.destroyAllWindows()
```


I . Color Space

LAB Color Space

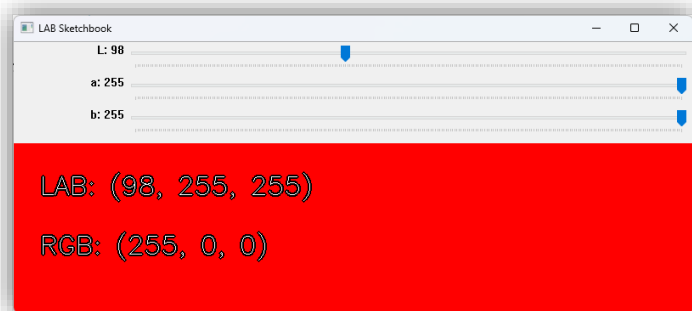
- 사람이 실제로 느끼는 색감에 더 가깝게 만든 색상 모델로 조명 변화에 강해서, 색 필터링, 색 추출, 색 비교 시 강점
- L (Lightness): 밝기 (0 = 어두움, 255 = 밝음)
- a: 초록색 ↔ 빨간색 (값이 작을수록 초록, 클수록 빨강)
- b: 파란색 ↔ 노란색 (값이 작을수록 파랑, 클수록 노랑)



I. Color Space

LAB Color Space (cont.)

- Tracker 를 사용해 L,A,B 값을 control 하여 변하는 LAB color 값을 확인하는 예제 코드



OpenCV 에서 기본적으로 사용하는 색공간은 BGR 이기 때문에 Display 출력을 위해 LAB 을 BGR 로 변경이 필요함

```
import cv2
import numpy as np
def nothing(x):
    pass

img = np.zeros((200, 800, 3), np.uint8) # Create a blank image (sketchbook)
# Create LAB Sketchbook window and add trackbars
cv2.namedWindow('LAB Sketchbook')
cv2.createTrackbar('L', 'LAB Sketchbook', 0, 255, nothing) # Lightness: 0~255
cv2.createTrackbar('a', 'LAB Sketchbook', 0, 255, nothing) # a: green-red, center 128
cv2.createTrackbar('b', 'LAB Sketchbook', 0, 255, nothing) # b: blue-yellow, center 128

while True:
    l = cv2.getTrackbarPos('L', 'LAB Sketchbook') # Get current trackbar position for L
    a = cv2.getTrackbarPos('a', 'LAB Sketchbook') # Get current trackbar position for a
    b = cv2.getTrackbarPos('b', 'LAB Sketchbook') # Get current trackbar position for b

    lab_color = np.uint8([[[l, a, b]]]) # Convert LAB to BGR (OpenCV uses BGR)
    rgb_b, rgb_g, rgb_r = cv2.cvtColor(lab_color, cv2.COLOR_LAB2BGR)[0][0]

    text_lab = f"LAB: ({l}, {a}, {b})" # Prepare text for LAB
    text_rgb = f"RGB: ({rgb_r}, {rgb_g}, {rgb_b})" # Prepare text for RGB

    # Draw text (black border + white text)
    cv2.putText(img, text_lab, (30, 60), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 3)
    cv2.putText(img, text_lab, (30, 60), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 1)
    cv2.putText(img, text_rgb, (30, 130), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 3)
    cv2.putText(img, text_rgb, (30, 130), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 1)

    img[:] = [rgb_b, rgb_g, rgb_r] # Apply background color

    cv2.imshow('LAB Sketchbook', img) # Show image

    # Exit on ESC key
    if cv2.waitKey(1) & 0xFF == 27:
        break

cv2.destroyAllWindows()
```

I . Color Space

RGB vs HSV vs LAB

항목	RGB	HSV	LAB
구성	Red, Green, Blue	Hue, Saturation, Value	Lightness, a (green-red), b (blue-yellow)
범위 (OpenCV)	0~255 (각 채널)	H: 0179, S/V: 0255	L: 0255, a/b: 0255 (중심 128)
시각적 직관성	낮음	높음 (색 분리 쉬움)	높음 (사람 눈 기준)
조명 영향	높음	중간	낮음 (조명 보정에 강함)
용도	이미지 저장/표현	색 추적, 객체 인식	색 보정, 색 차이 분석 (ΔE), 밝기 분리 작업

