Day 10: Color Detection in Live Video

Outcomes:

- Capture live video from a webcam
- Apply HSV-based color detection on video frames
- Combine trackbars + video feed
- Perform real-time color segmentation

Try it yourself

Problem Statement

Write a Python program using OpenCV to perform real-time color detection on a live video using Trackbars with a proper exit mechanism.

Hints

- In this problem, three concepts are combined:
    - Capturing Video,
    - Color Detection and
    - Thresholding using Trackbars.
- The operations like creating a window, creating the trackbars need to be performed only once, should be kept outside the loop.
- Operations like getting frame input and reading trackbar positions need to be done continuously, hence they need to be placed inside the loop.

References

Capturing Video	: `Day 1 / 1-camera.py`
Color Detection	: `Day 8 / 13-colorDetection.py`
Trackbars	: `Day 9 / 14-trackbar.py`

15-videoColorDetection.py

```python
#Practical 15: Color Detection in Live Video

import cv2 as cv
import numpy as np

cap = cv.VideoCapture(0)

if not cap.isOpened():
    print("Camera not accessible")
    exit()

# Dummy function
def nothing(x):
    pass

# Create a Window
cv.namedWindow("Trackbars")

# Create Trackbars
cv.createTrackbar("LH", "Trackbars", 0, 179, nothing)
cv.createTrackbar("LS", "Trackbars", 0, 255, nothing)
cv.createTrackbar("LV", "Trackbars", 0, 255, nothing)
cv.createTrackbar("UH", "Trackbars", 179, 179, nothing)
cv.createTrackbar("US", "Trackbars", 255, 255, nothing)
cv.createTrackbar("UV", "Trackbars", 255, 255, nothing)
```

```python
while True:
    ret, frame = cap.read()
    if not ret:
        print("An Error Occurred")
        break

    # Convert Into HSV Image
    hsv = cv.cvtColor(frame, cv.COLOR_BGR2HSV)

    # Get Trackbar Position
    lh = cv.getTrackbarPos("LH", "Trackbars")
    ls = cv.getTrackbarPos("LS", "Trackbars")
    lv = cv.getTrackbarPos("LV", "Trackbars")

    uh = cv.getTrackbarPos("UH", "Trackbars")
    us = cv.getTrackbarPos("US", "Trackbars")
    uv = cv.getTrackbarPos("UV", "Trackbars")

    # Update Threshold
    lowerbound = np.array([lh, ls, lv])
    upperbound = np.array([uh, us, uv])

    # Mask
    mask = cv.inRange(hsv, lowerbound, upperbound)

    # Apply the Mask
    result = cv.bitwise_and(frame, frame, mask=mask)
```

© 2026 Priyal OpenCV with Python Beginner Notes

```python
    # Display the result
    cv.imshow("Mask", mask)
    cv.imshow("Result", result)

    # Exit on 'q' Press
    if cv.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv.destroyAllWindows()
```