Artificial Intelligence

Task Report : Image Color Detector

JULY 1

Company Name: Uneeq Interns
Authored by: Injy Nashaat Makram

Explanation of code in details:

1. Color Ranges (color_ranges Dictionary):

- Each color has its HSV range defined based on some values.
- Adjustments were made to ensure each color range accurately reflects the specified hues, saturations, and values for better detection.

2. get_color Function:

- It iterates through color_ranges and returns the color name if the HSV values fall within any defined range.

3. Video Capture and Setup:

- Uses the default camera (0) and sets the frame width and height to 1280x720 pixels.

4. Main Loop (while True):

- Continuously reads frames from the video capture.
- Converts each frame from BGR to HSV color space.

5. Pixel Processing:

- Retrieves HSV values of the center pixel.
- Converts them into a NumPy array (hsv_value).

6. Color Detection:

- Calls get_color to determine the color name based on the HSV values.

7. Display:

- Draws a rectangle filled with white to display the detected color name.
- Puts text (color name) on the frame.
- Draws a small circle at the center of the frame.

8. User Input Handling:

- Waits for a key press (cv2.waitKey(1)). If the Esc key (ASCII value 27) is pressed, it breaks out of the loop (if key == 27: break).

9. Release Resources:

- Releases the video capture (cap.release()) to free up the camera.
- Closes all OpenCV windows (cv2.destroyAllWindows()) upon exiting the loop.

```python
import cv2
import numpy as np

# Define HSV ranges for color detection
color_ranges = {
    "black": ([0, 0, 0], [180, 255, 50]),        # Black
    "white": ([0, 0, 150], [180, 30, 255]),      # White
    "red": ([170, 100, 20], [180, 255, 255]),   # Red range 2 (wrap-around in HSV)
    "yellow": ([20, 100, 100], [40, 255, 255]),  # Yellow
    "green": ([40, 50, 50], [90, 255, 255]),     # Green
    "blue": ([90, 50, 50], [130, 255, 255]),     # Blue
    "purple": ([130, 50, 50], [160, 255, 255])   # Purple
}

def get_color(hsv_value):
    for color_name, (lower, upper) in color_ranges.items():
        lower_bound = np.array(lower)
        upper_bound = np.array(upper)
        if np.all(hsv_value >= lower_bound) and np.all(hsv_value <= upper_bound):
            return color_name
    return "Undefined"

cap = cv2.VideoCapture(0)  # Use 0 for the default camera
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)

while True:
    _, frame = cap.read()
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    height, width, _ = frame.shape

    cx = int(width / 2)
    cy = int(height / 2)

    # Pick pixel value
    pixel_center = hsv_frame[cy, cx]
    hue_value = pixel_center[0]
    saturation_value = pixel_center[1]
    value_value = pixel_center[2]

    # Get color name
    hsv_value = np.array([hue_value, saturation_value, value_value])
    color_name = get_color(hsv_value)

    # Display the detected color
    pixel_center_bgr = frame[cy, cx]
    b, g, r = int(pixel_center_bgr[0]), int(pixel_center_bgr[1]), int(pixel_center_bgr[2])

    cv2.rectangle(frame, (cx - 220, 10), (cx + 200, 120), (255, 255, 255), -1)
    cv2.putText(frame, color_name, (cx - 200, 100), cv2.FONT_HERSHEY_SIMPLEX, 3, (b, g, r), 5)
    cv2.circle(frame, (cx, cy), 5, (25, 25, 25), 3)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1)
    if key == 27:  # Press 'Esc' to exit
        break

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