**NAME :Dua zahra**

**Roll number:036**

**SUBJECT:PAI-Lab**

**Web cam drwing**

**Introduction**

This project demonstrates a real-time drawing application using a webcam and OpenCV. The program tracks specific colors in the video feed and allows the user to draw on the screen by moving a colored object. This can be used for interactive applications, digital whiteboards, or creative drawing without physical contact.

**Explanation of task6.py Code in Simple English**

This Python script is used to track specific colors in a video feed from a webcam and draw on the screen based on detected colors. Below is a step-by-step explanation of how the code works:

### ****1. Setting Up the Webcam****

* The script starts by capturing video from the webcam using cv2.VideoCapture(0), where 0 represents the default camera.
* The resolution of the webcam is set using cap.set(3, frameWidth) and cap.set(4, frameHeight).
* The brightness of the camera is also adjusted using cap.set(10, 150).

### ****2. Defining Color Ranges****

* The myColors list contains HSV (Hue, Saturation, Value) ranges for different colors. These colors will be detected in the webcam feed.
* The myColorValues list stores the RGB values that will be used to draw corresponding colors on the screen.

### ****3. Detecting Specific Colors****

* The function findColor(img, myColors, myColorValues) processes the webcam image to find specific colors.
* It converts the image to the HSV color space using cv2.cvtColor(img, cv2.COLOR\_BGR2HSV).
* A loop runs through the predefined color ranges and creates a mask using cv2.inRange(imgHSV, lower, upper), which filters out everything except the desired color.
* The function then finds the position of the detected color and stores it in newPoints.

### ****4. Finding Contours (Shapes of the Detected Objects)****

* The function getContours(img) is used to identify the shapes of the detected colors.
* It extracts the outlines of detected objects using cv2.findContours().
* If the area of a detected object is greater than 500, it is considered a valid object.
* The position (x, y) of the detected color is determined and returned.

### ****5. Drawing on the Screen****

* The function drawOnCanvas(myPoints, myColorValues) draws circles on the screen at the detected points.
* It takes the stored points from myPoints and uses cv2.circle() to draw on the screen.

### ****6. Running the Program in a Loop****

* The while True loop continuously reads frames from the webcam.
* It processes the frames, finds colors, detects their positions, and draws on the screen.
* If the user presses the **'q' key**, the loop breaks, and the program stops.

### ****7. Closing the Program****

* After exiting the loop, the webcam is released using cap.release().
* The window is closed using cv2.destroyAllWindows().

### ****Summary****

This code detects specific colors from a webcam feed and allows the user to "draw" by moving a colored object in front of the camera. The detected points are stored and continuously displayed on the screen.