🔍 **Color Detection from Image & Video using OpenCV**

Over the past few days, I have developed two interactive computer vision tools that allow users to click anywhere on an image or video frame to detect and display the **closest color name** and **exact RGB values** with a color-highlighted rectangle and text label. This project helped me deepen my understanding of image processing and user interaction using Python.

🧠 **Project Overview**  
I developed two interactive tools using **OpenCV**, **NumPy**, and **Pandas**:

✅ **Image Color Detection Tool**  
✅ **Video Color Detection Tool**

🧩 **How it works**

**1. Media Input**

* For images: cv2.imread()
* For video: cv2.VideoCapture()

**2. Mouse Interaction**

* Used cv2.setMouseCallback() to capture click coordinates
* Retrieved the RGB values of the clicked pixel

**3. Color Matching**

* A CSV dataset (colors.csv) contains color names and RGB values
* The app calculates the closest match using **Manhattan distance**

**4. Visualization**

* Displayed the color name and RGB in a labeled rectangle using cv2.rectangle() and cv2.putText()
* Rectangle position is auto centered at the top of the frame

**5. Specific Features**

* Press **“Q”** to quit
* Each click clears the previous detection to avoid overlapping

**Technologies used**

* **Python**
* **OpenCV** – for image handling, display, and interaction
* **Pandas** – for reading and searching the color dataset (colors.csv)
* **NumPy** – *(Imported but not directly used in this version)*  
  OpenCV images are represented as NumPy arrays. Although I didn’t use NumPy functions explicitly in this version, it's often essential for more advanced operations.

📌 **Challenges I Solved**

* Ensured accurate color matching with large datasets
* Handling file errors or missing datasets gracefully
* Made color display seamless and non-overlapping
* Improving user experience during real-time video playback