

```

import cv2
import pandas as pd

# Files - Ensure these are in the same folder
img_path = 'nature.jpg'
csv_path = 'colors.csv'

# Reading the CSV file
index = ["color", "color_name", "hex", "r", "g", "b"]
data = pd.read_csv(csv_path, names=index, header=None)

# Reading and Resizing image
img = cv2.imread(img_path)
if img is None:
    print("Error: Could not find nature.jpg")
    exit()
img = cv2.resize(img, (800, 600))
# Create a copy to allow clearing/resetting annotations
original_img = img.copy()

clicked = False
r = g = b = xpos = ypos = 0

def get_color_name(R, G, B):
    minimum = 10000
    for i in range(len(data)):
        d = abs(R - int(data.loc[i, "r"])) + abs(G - int(data.loc[i, "g"])) + abs(B - int(data.loc[i, "b"]))
        if d <= minimum:
            minimum = d
            cname = data.loc[i, "color_name"]
    return cname

def draw_function(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        global b, g, r, xpos, ypos, clicked
        clicked = True
        xpos = x
        ypos = y
        b, g, r = img[y, x]
        b, g, r = int(b), int(g), int(r)

cv2.namedWindow('Nature Color Detector')
cv2.setMouseCallback('Nature Color Detector', draw_function)

print("--- COMMANDS ---")
print("Double-click: Detect Color")
print("Press 'c' : Clear annotations")

```

```
print("Press 'ESC' : Exit")

while True:
    cv2.imshow("Nature Color Detector", img)
    if clicked:
        # Draw display bar
        cv2.rectangle(img, (20, 20), (780, 65), (b, g, r), -1)

        # Determine text color based on brightness
        #
        text_color = (0, 0, 0) if (r + g + b) > 600 else (255, 255, 255)

        text = f"{{get_color_name(r, g, b)}} (R:{r}, G:{g}, B:{b})"
        cv2.putText(img, text, (40, 50), cv2.FONT_HERSHEY_SIMPLEX, 0.8, text_color, 2)
        clicked = False

    key = cv2.waitKey(20) & 0xFF
    if key == 27: # ESC to exit
        break
    elif key == ord('c'): # 'c' to clear
        img = original_img.copy()

cv2.destroyAllWindows()
```