

**Bhattbhoomi111**

9 Followers

Lists

About



# [12]Mini Project : Colour Detection using Pandas & OpenCV



Bhattbhoomi111 2 hours ago · 3 min read

Colour detection is the process of detecting the name of any color. for humans this is an extremely easy task but for computers, it is not straightforward. Human eyes and brains work together to translate light into color. Light receptors that are present in our eyes transmit the signal to the brain. Our brain then recognizes the color. Since childhood, we have mapped certain lights with their color names. We will be using the somewhat same strategy to detect color names.

Colors are made up of 3 primary colors; red, green, and blue. In computers, we define each color value within a range of 0 to 255. So in how many ways we can define a color? The answer is  $256 \times 256 \times 256 = 16,581,375$ . There are approximately 16.5 million different ways to represent a color. In our dataset, we need to map each color's values with their corresponding names. But don't worry, we don't need to map all the values. We will be using a dataset that contains RGB values with their corresponding names. The CSV file for our dataset has been taken from this link: [Colors Dataset](#)

```
import cv2
import numpy as np
import pandas as pd
import argparse

#Creating argument parser to take image path from command line
ap = argparse.ArgumentParser()
ap.add_argument('-i', '--image', required=True, help="Image Path")
args = vars(ap.parse_args())
img_path = args['image']

#Reading the image with opencv
img = cv2.imread(img_path)

#declaring global variables (are used later on)
clicked = False
r = g = b = xpos = ypos = 0
```

```
#Reading csv file with pandas and giving names to each column
index=["color","color_name","hex","R","G","B"]
csv = pd.read_csv('colors.csv', names=index, header=None)

#function to calculate minimum distance from all colors and get the
most matching color
def getColorName(R,G,B):
    minimum = 10000
    for i in range(len(csv)):
        d = abs(R- int(csv.loc[i,"R"])) + abs(G-
int(csv.loc[i,"G"]))+ abs(B- int(csv.loc[i,"B"]))
        if(d<=minimum):
            minimum = d
            cname = csv.loc[i,"color_name"]
    return cname

#function to get x,y coordinates of mouse double click
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 = int(b)
        g = int(g)
        r = int(r)

        cv2.namedWindow('image')
        cv2.setMouseCallback('image',draw_function)

while(1):

    cv2.imshow("image",img)
    if (clicked):
        #cv2.rectangle(image, startpoint, endpoint, color,
```

```
thickness)-1 fills entire rectangle
cv2.rectangle(img, (20,20), (750,60), (b,g,r), -1)

#Creating text string to display( Color name and RGB values )
text = getColorName(r,g,b) + ' R=' + str(r) + ' G=' + str(g) +
' B=' + str(b)

#cv2.putText(img, text, start, font(0-
7), fontScale, color, thickness, lineType )
cv2.putText(img, text, (50,50), 2, 0.8,
(255,255,255), 2, cv2.LINE_AA)

#For very light colours we will display text in black colour
if(r+g+b>=600):
    cv2.putText(img, text, (50,50), 2, 0.8,
(0,0,0), 2, cv2.LINE_AA)

clicked=False

#Break the loop when user hits 'esc' key
if cv2.waitKey(20) & 0xFF ==27:
    break

cv2.destroyAllWindows()
```



Data source : [Color Detection Zip File](#)

Thank you!