# **Color Detection from Images**

# Color Detection from Image - Google Colab Version

# Install required libraries

!pip install opencv-python pandas

# Import libraries

import cv2

import pandas as pd

import numpy as np

from google.colab.patches import cv2\_imshow

from google.colab import files

# Upload an image

uploaded = files.upload()

image\_path = list(uploaded.keys())[0]

# Read image using OpenCV

img = cv2.imread(image\_path)

img = cv2.resize(img, (800, 600)) # Resize for display

cv2\_imshow(img)

# Load color dataset

url = 'https://raw.githubusercontent.com/codebrainz/color-names/master/output/colors.csv'

colors = pd.read\_csv(url)

# Function to calculate closest color name

def get\_color\_name(R, G, B):

minimum = float('inf')

cname = "Unknown"

for i in range(len(colors)):

d = abs(R - int(colors.loc[i, "R"])) + abs(G - int(colors.loc[i, "G"])) + abs(B - int(colors.loc[i, "B"]))

if d < minimum:

minimum = d

cname = colors.loc[i, "Name"]

return cname

# Define mouse click event

clicked = False

r = g = b = xpos = ypos = 0

def draw\_function(event, x, y, flags, param):

global b, g, r, xpos, ypos, clicked

if event == cv2.EVENT\_LBUTTONDOWN:

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 True:

cv2.imshow("Image", img)

if clicked:

# Display color info

color\_name = get\_color\_name(r, g, b)

cv2.rectangle(img, (20, 20), (750, 60), (b, g, r), -1)

text = f'{color\_name} R={r} G={g} B={b}'

cv2.putText(img, text, (50, 50), 2, 0.8, (255 - b, 255 - g, 255 - r), 2, cv2.LINE\_AA)

clicked = False

if cv2.waitKey(20) & 0xFF == 27: # ESC key to exit

break

cv2.destroyAllWindows()