ALBUKHARY INTERNATIONAL UNIVERSITY

SEMESTER 2 YEAR 1, COHORT D

| | | |
|---|---|---|
| **Image Color Detection Using OpenCV and Pandas** | **CCC1234** | **Course Name: Artificial Intelligence** |

| | | |
|---|---|---|
| **Group Members** | Dema Yuden (AIU22102234)<br>Kelden Dorji (AIU22102187)<br>Thinley Yeshey Choden (AIU22102188) | Nima Yoezer (AIU22102221)<br>Tandin Namgay (AIU22102185)<br>Ugyen Tshering (AIU22102222) |

**Assessment Marks:**

| No. | Criteria | Weight | Marks |
|---|---|---|---|
| 1 | | | |
| 2 | | | |
| 3 | | | |
| | Total Marks | | |

**Comments:**

**Lecturer’s Signature:**

**Lecturer’s Name: Dr. Akibu Mahmoud Abdullahi** **Date: 24-05-2023**

# INTRODUCTION

## What is Color Detection?

“Image Color Detection” is the technique of identifying the various hues in a given image. Humans perceive color from light using their eyes and brains. When light strikes the eyes, the receptors in the eyes give signals to the brain, which interprets the light and converts it into colors. This happens easily for humans but for machines it is not straightforward.

## About The Python Project

In this Python project, we'll create an application which will enable the users to obtain the names of various colors by clicking anywhere on a picture. A CSV dataset is extracted that contains the name of colors and its values.

## The Dataset

Red, green, and blue are the three fundamental colors that make up all other colors. A computer's specified color values fall between the range 0 and 255. A color may be expressed in approximately 16.5 million various ways (256*256*256 = 16,581,375). We must associate each color's value in the dataset with its corresponding names. However, we won't map every value. Instead, a dataset containing the RGB values and its corresponding names will be used. The CSV dataset is extracted from: Colors Dataset

The colors.csv file contains 865 distinct color names along with their RGB and hex values.

## Prerequisites

Python packages like OpenCV, Pandas, and numpy are necessary for this project.

## OpenCV

With the aid of OpenCV, a potent open-source computer vision toolkit, we can quickly analyze photos and extract color data that is accurate and timely. In several disciplines, including computer vision, image processing, and robotics, color detection is crucial. Thanks to OpenCV's extensive documentation and wide range of tools and algorithms, we were able to develop sophisticated color detection algorithms and modify them to satisfy certain requirements.

## MOTIVATION

Our fascination with color is what motivated this topic. Color is an important aspect of life that should be shared and enjoyed. It holds immense importance in various aspects of our lives, impacting our perception, emotions, communication, and aesthetics.

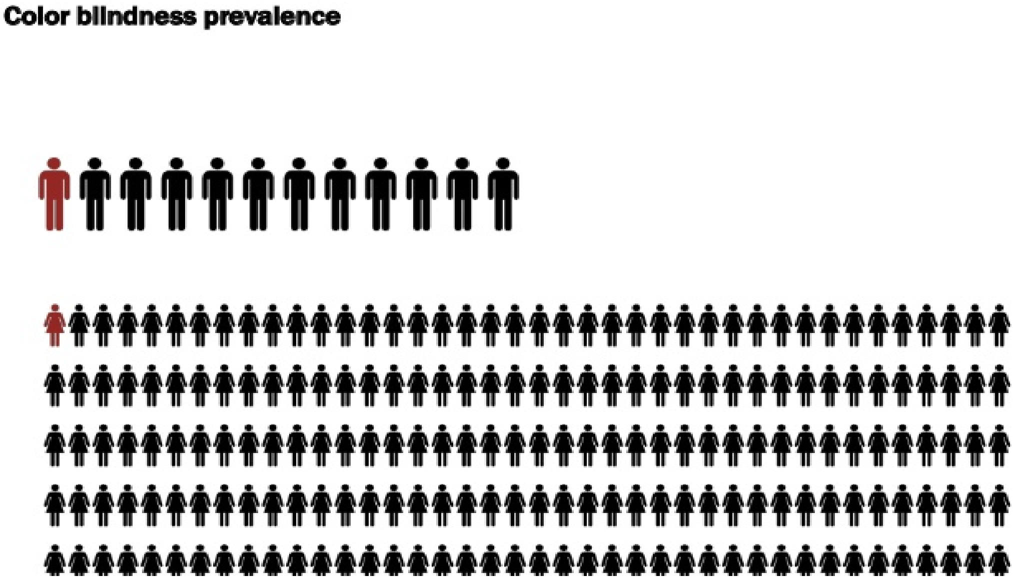
Source: A guide to Bhutan’s colorful festivals

This image depicts a religious celebration in Bhutan. The screen literally seems to peek through with the vivid colors. In various cultures and traditions, colors frequently have symbolic meanings to it. Red, for instance, may signify good fortune and wealth in certain cultures yet danger or caution in others. For respectful inclusion in different societies and for successful communication, it is important to understand these cultural linkages.

In addition to  this, colors have many additional uses. For example, traffic lights use different colors (red, yellow, green) to convey specific meanings, ensuring effective communication and secure navigation.Colors have the ability to stir up strong emotions and affect our mood. Cool colors like blue and green can promote calmness, relaxation, or tranquility whereas warm colors like red and orange can evoke feelings of enthusiasm, passion, or warmth.

However, there are more than 350 million colorblind persons in the globe as a result of the common belief that 1 in 12 males (8%) and 1 in 200 women (about 4.5% of the world's population) are colorblind. In accordance with the 1.05 percent annual growth in global population, this figure rises each year.

Many color-blind people are astonished to find out that peanut butter isn't really green!

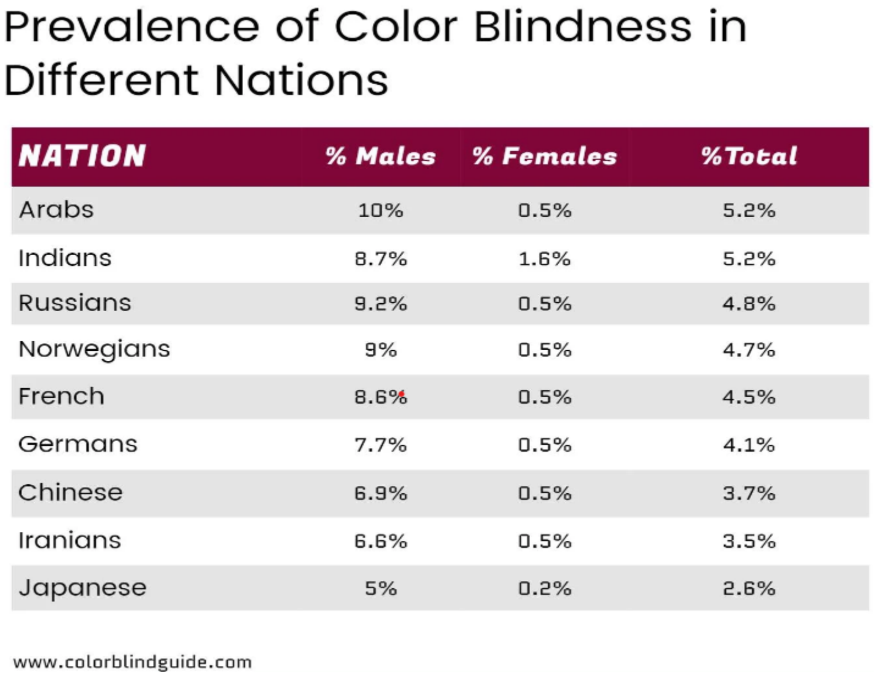


Source: *Colorblind People Population! Statistics*. (2021, June 13). Colorblind People Population! Statistics. Colorblind Guide

The phrase "colorblind" does have ambiguity in it. Most colorblind individuals can perceive a huge range of hues, but it is not as wide as others. Seeing in black and white is really not that common.

Red-green color blindness affects 99% of all colorblind persons, according to Colblindor. Monochromacy color blindness, often known as total color blindness, affects an extremely tiny percentage of people—possibly 1 in 33,000.

Color blindness, even the less common kinds, can even be prevalent in nations with smaller gene pools.



Prevalence of Color Blindness in Different Nations

| NATION | % Males | % Females | %Total |
|---|---|---|---|
| Arabs | 10% | 0.5% | 5.2% |
| Indians | 8.7% | 1.6% | 5.2% |
| Russians | 9.2% | 0.5% | 4.8% |
| Norwegians | 9% | 0.5% | 4.7% |
| French | 8.6% | 0.5% | 4.5% |
| Germans | 7.7% | 0.5% | 4.1% |
| Chinese | 6.9% | 0.5% | 3.7% |
| Iranians | 6.6% | 0.5% | 3.5% |
| Japanese | 5% | 0.2% | 2.6% |

www.colorblindguide.com

The quality of life for so many individuals around the world is hugely impacted by color blindness.

Taking the lead in developing the first validated questionnaire assessment of the impact of color blindness on quality of life (the CBQoL), Dr. John Barry significantly contributed to this condition. The study, which was conducted in cooperation with ophthalmology specialists at the University of Birmingham, was released in the peer-reviewed journal Biomed Central Ophthalmology in 2017. The main UK organization for color blindness, Colour Blind Awareness.org, assisted in recruiting participants for the study.

The primary result of this study was that color blindness significantly impacts a person's life in terms of restricting their profession (certain jobs require proper color vision), their health (for example, some illness signs are ignored by color blind persons), and their happiness.

Even though our project may not directly address the needs of color-blind individuals, they can indirectly contribute to inclusivity.

- ❖ Object recognition based on color can assist in identifying objects using shape and texture cues. Image processing techniques can enhance image differentiation by using alternative color spaces or adjusting contrasts.
- ❖ Automation systems can add cues like texture or barcodes for sorting and quality control.
- ❖ Unique applications can provide tools for color-blind individuals to express color differently.

**Though not entirely targeting color-blindness, these projects can contribute to more inclusive experiences.**

**Steps for Building Image Color Detection Project:-**

1. **Download the zip file and extract the file**

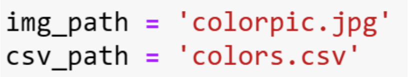
Color Detection Zip File

The zip folder contains the source code, sample images and csv file:

- color_detection.py – source code.
- colorpic.jpg – sample .jpg image for testing.
- colors.csv – csv file containing dataset.

2. **Declaring image path and csv path**

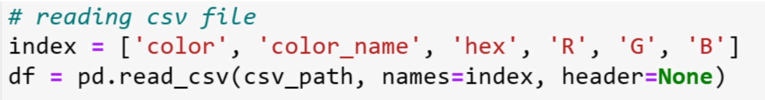
**img_path** will store the path of our image which is in .jpg format and **csv_file** will store the path of the csv file.



```
img_path = 'colorpic.jpg'
csv_path = 'colors.csv'
```

3. **Reading the CSV file with Pandas**

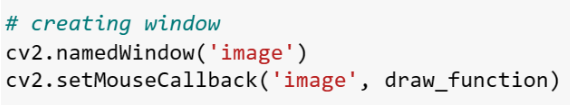
Pandas is a Python library in Python that allows us to easily work with tabular data that is often stored in .csv files. The **pd.read_csv()** method reads and loads the CSV file into the Pandas DataFrame. For simple access, each column is given a name.



```
# reading csv file
index = ['color', 'color_name', 'hex', 'R', 'G', 'B']
df = pd.read_csv(csv_path, names=index, header=None)
```

4. **Set a mouse callback event on a window**

To show the supplied image, we first created a window as ‘image’.  A callback function called **draw_function()** is defined that will be activated in response to a mouse event.

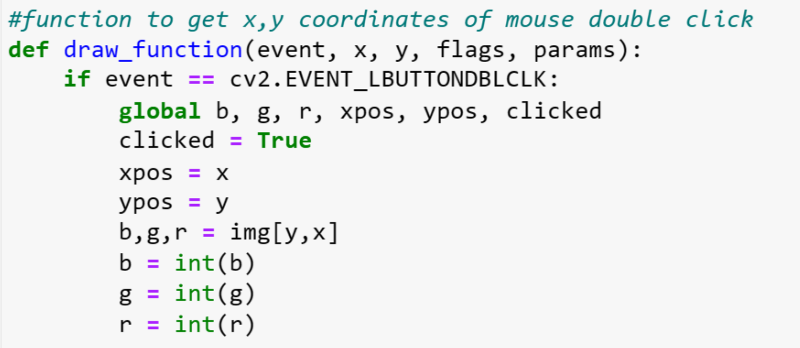


```
# creating window
cv2.namedWindow('image')
cv2.setMouseCallback('image', draw_function)
```

### 5. Creating draw_function

When we double-click a pixel, it will calculate the rgb values for that pixel. The function's arguments include the event name, the mouse position's (x, y) coordinates, etc.

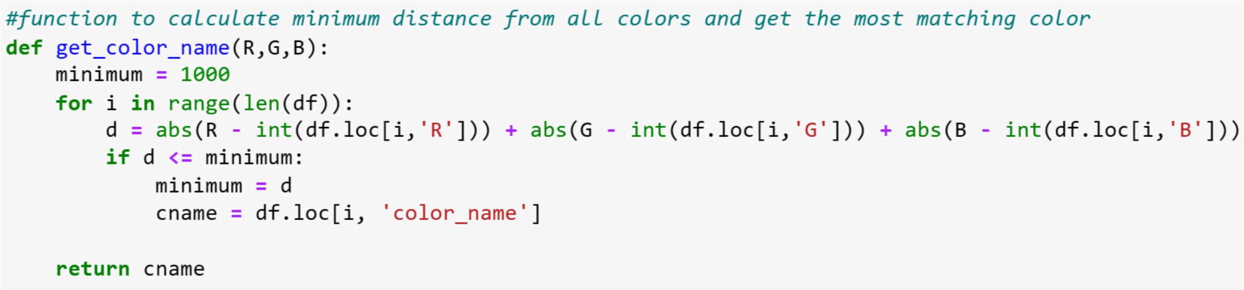
The function first determines if a double-click occurred before calculating and setting the r, g, and b values along with the mouse's x and y positions.



```
#function to get x,y coordinates of mouse double click
def draw_function(event, x, y, flags, params):
    if event == cv2.EVENT_LBUTTONDBLCLK:
        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)
```

### 6. Calculating the minimum distance to get the most matching color name

We know the values for r, g, and b. We now want another function that will return the color name from the RGB data. We determine a distance (d) that indicates how close we are to a color in order to obtain its name, then we select the color with the smallest distance.



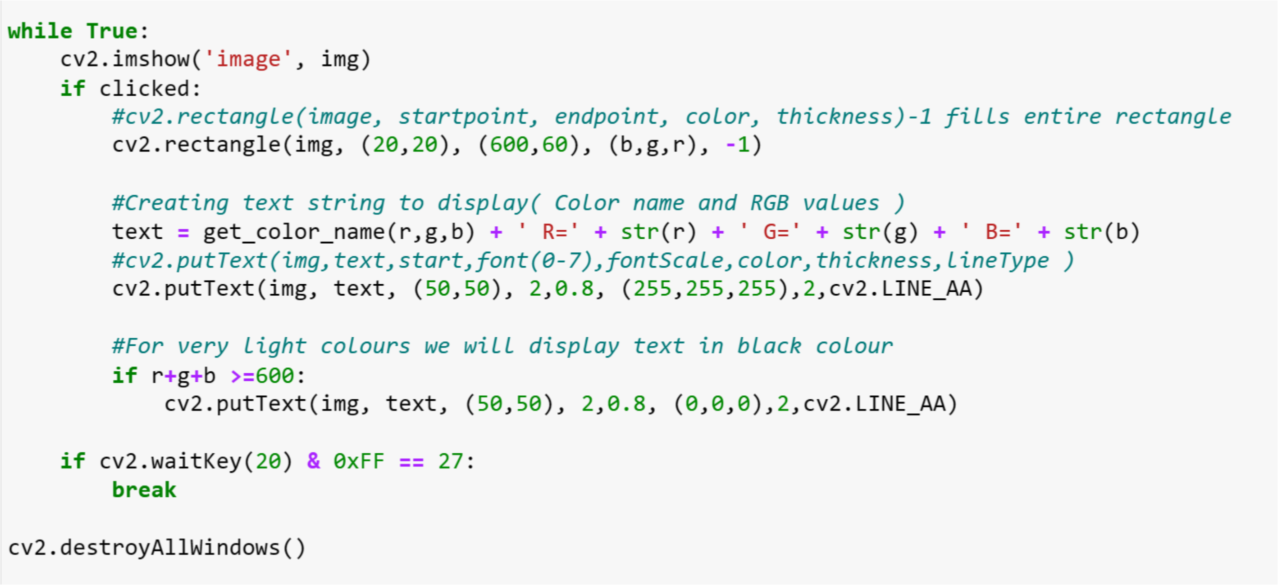
```
#function to calculate minimum distance from all colors and get the most matching color
def get_color_name(R,G,B):
    minimum = 1000
    for i in range(len(df)):
        d = abs(R - int(df.loc[i,'R'])) + abs(G - int(df.loc[i,'G'])) + abs(B - int(df.loc[i,'B']))
        if d <= minimum:
            minimum = d
            cname = df.loc[i, 'color_name']

    return cname
```

Distance is calculated using the formula:

*d = abs(Red – ithRedColor) + (Green – ithGreenColor) + (Blue – ithBlueColor)*

### 7. Display image on the window

Each time a mouse double click event takes place, the color name and RGB values on the window are updated. The cv2.imshow() method is used to display the image on the window. Using the cv2.rectangle and cv2.putText() methods, we construct a rectangle when the user double-clicks the window and the right color name is obtained.

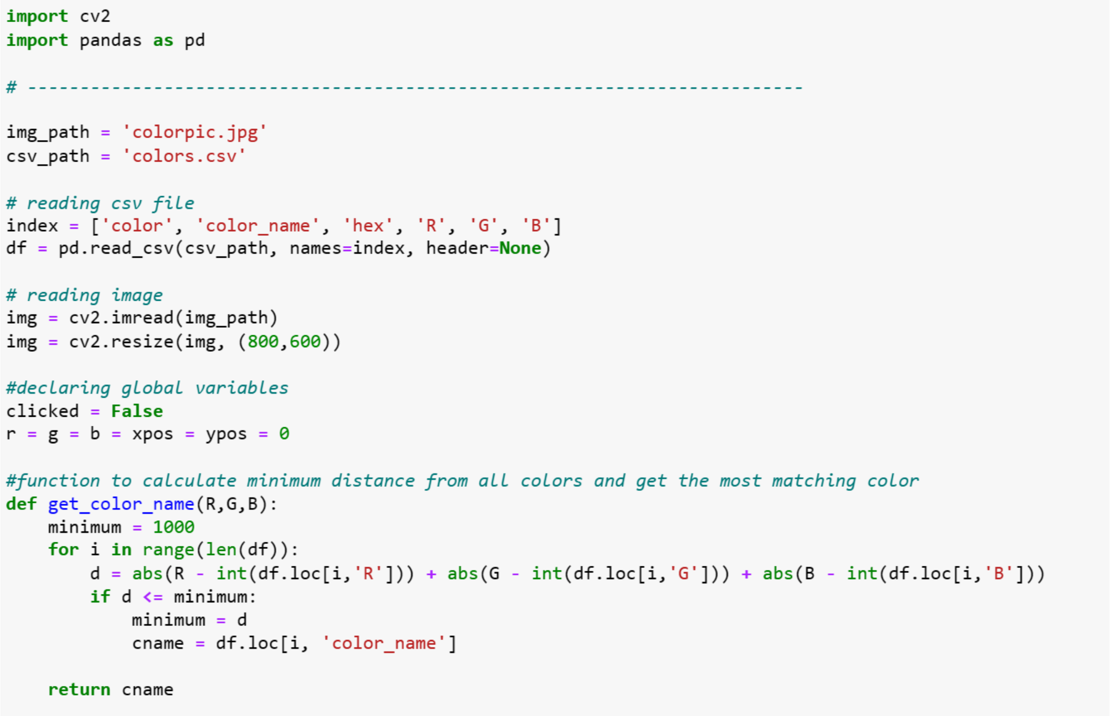


```
while True:
    cv2.imshow('image', img)
    if clicked:
        #cv2.rectangle(image, startpoint, endpoint, color, thickness)-1 fills entire rectangle
        cv2.rectangle(img, (20,20), (600,60), (b,g,r), -1)

        #Creating text string to display( Color name and RGB values )
        text = get_color_name(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)

    if cv2.waitKey(20) & 0xFF == 27:
        break

cv2.destroyAllWindows()
```

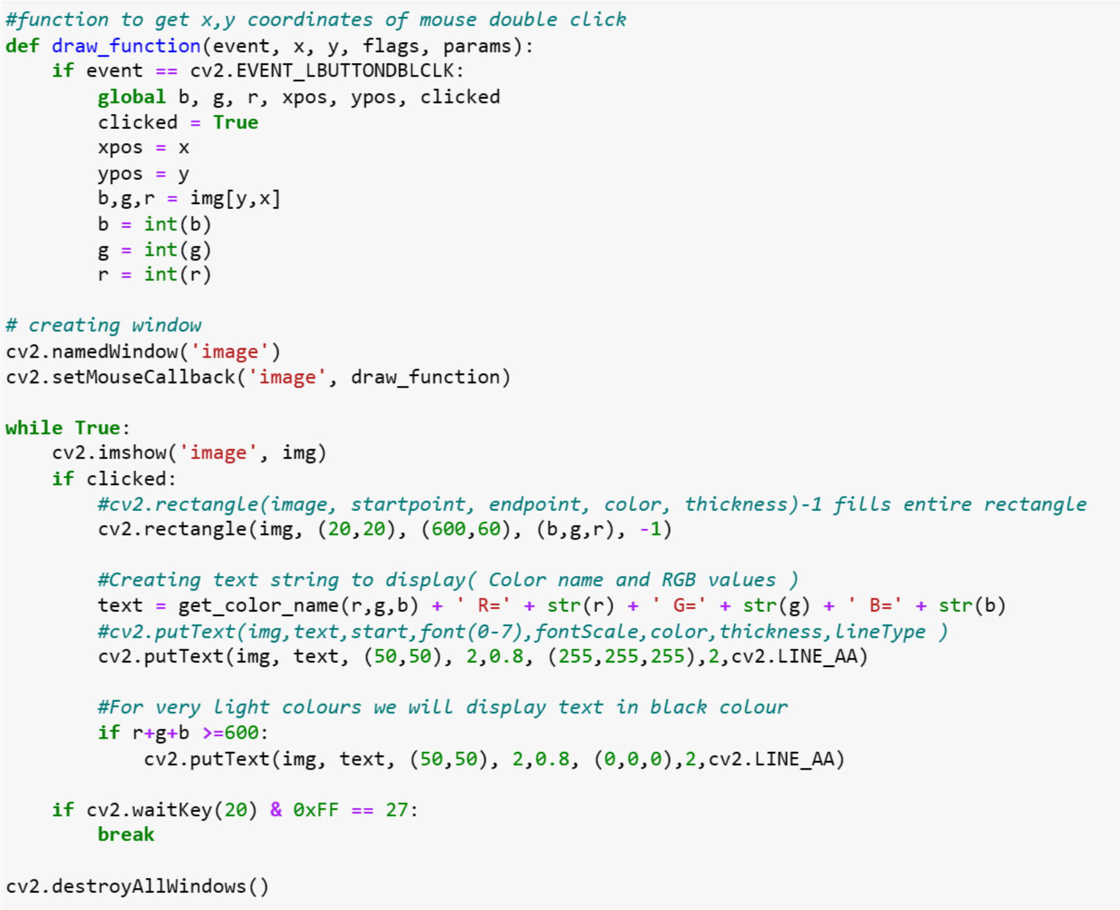
## 8. Running the Python Code

After every code is in place, we run it and the required output is displayed.

# APPENDIX



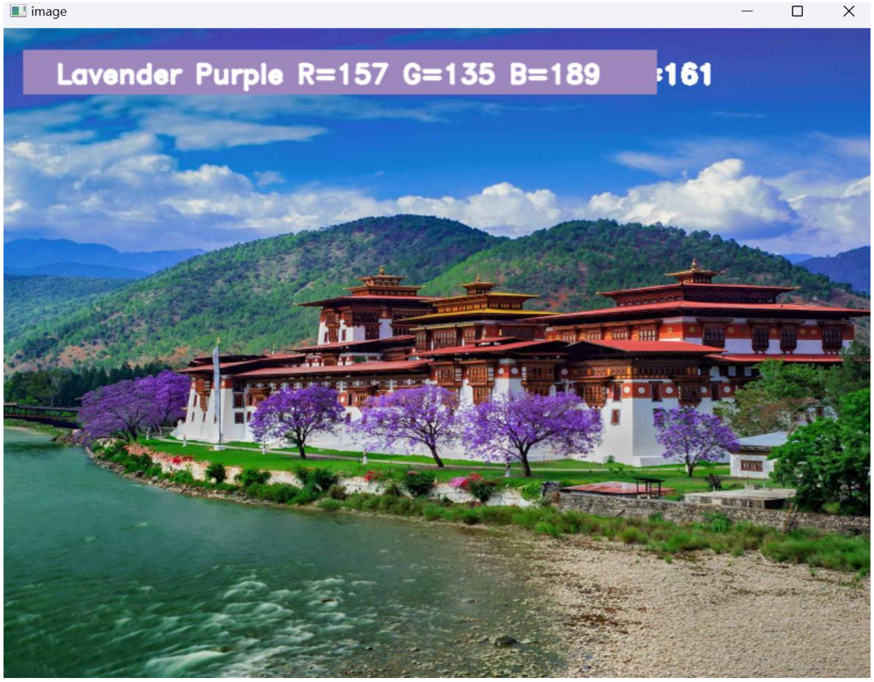
```
import cv2
import pandas as pd

# --------------------------------------------------------------------------

img_path = 'colorpic.jpg'
csv_path = 'colors.csv'

# reading csv file
index = ['color', 'color_name', 'hex', 'R', 'G', 'B']
df = pd.read_csv(csv_path, names=index, header=None)

# reading image
img = cv2.imread(img_path)
img = cv2.resize(img, (800,600))

#declaring global variables
clicked = False
r = g = b = xpos = ypos = 0

#function to calculate minimum distance from all colors and get the most matching color
def get_color_name(R,G,B):
    minimum = 1000
    for i in range(len(df)):
        d = abs(R - int(df.loc[i,'R'])) + abs(G - int(df.loc[i,'G'])) + abs(B - int(df.loc[i,'B']))
        if d <= minimum:
            minimum = d
            cname = df.loc[i, 'color_name']

    return cname
```



```
#function to get x,y coordinates of mouse double click
def draw_function(event, x, y, flags, params):
    if event == cv2.EVENT_LBUTTONDBLCLK:
        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)

# creating window
cv2.namedWindow('image')
cv2.setMouseCallback('image', draw_function)

while True:
    cv2.imshow('image', img)
    if clicked:
        #cv2.rectangle(image, startpoint, endpoint, color, thickness)-1 fills entire rectangle
        cv2.rectangle(img, (20,20), (600,60), (b,g,r), -1)

        #Creating text string to display( Color name and RGB values )
        text = get_color_name(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)

    if cv2.waitKey(20) & 0xFF == 27:
        break

cv2.destroyAllWindows()
```

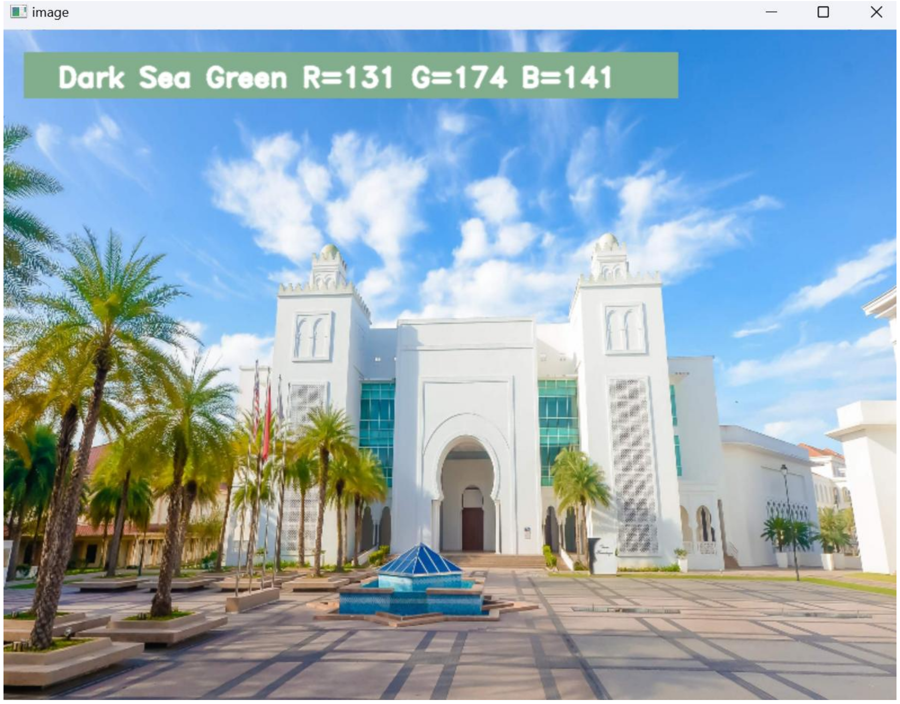
**Output:**

Using our mouse cursor, we can double click anywhere on the image below and the corresponding color will be displayed.



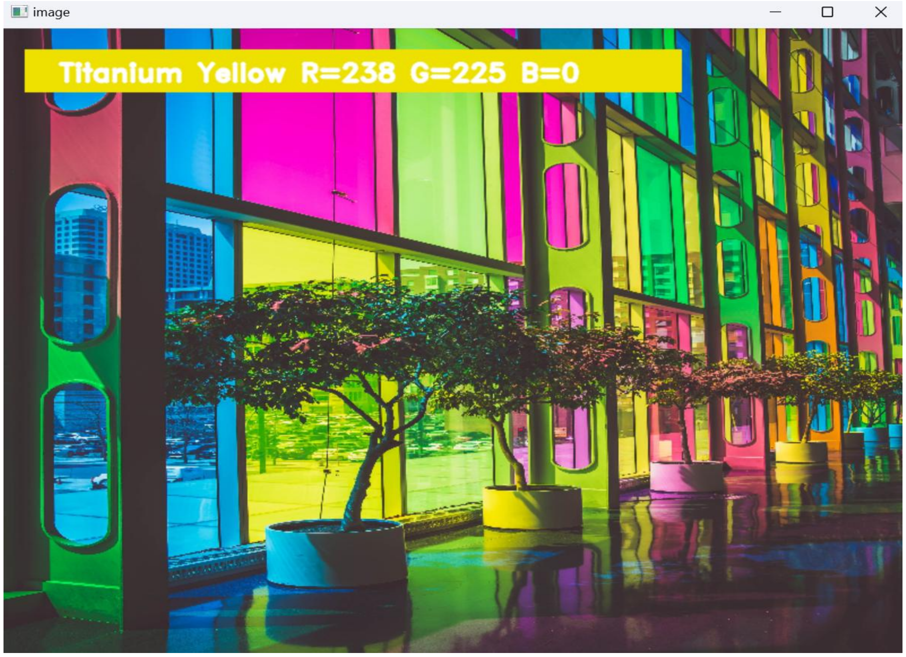
```
Lavender Purple R=157 G=135 B=189  :161
```

Source: S. (2022, September 19). *Bhutan National Day*. BHUTAN NATIONAL DAY - December 17, 2023 - National Today. https://nationaltoday.com/bhutan-national-day/



```
Dark Sea Green R=131 G=174 B=141
```

Source: *StackPath*. (n.d.). StackPath. https://aiu.edu.my/portfolio/campus-life/



Titanium Yellow R=238 G=225 B=0

Source: *Color detection.zip*. (n.d.). Google Docs. https://drive.google.com/file/d/1FEKZ7SsbLVrS__QeVb_Ehrp5dDBfuplp/view?usp=embed_facebook

## SUMMARY

The goal of this project was to develop a color detection algorithm that can identify and extract colors from input images. Using OpenCV, we develop a color detection algorithm that can identify and extract dominant colors from input images. Then Pandas was used to analyze and visualize the extracted color data. From this project, we learnt about colors and how to determine a pixel's color name and RGB values. Using Pandas, we demonstrated how to read CSV files, process the data, and respond to user input like double-clicking a window. Our results show that the algorithm is effective in identifying colors.

Apart from this, a number of capabilities and practical applications might be included in our project on picture color detection. Below are some potential paths to explore:

- Color Classification: Categorize pictures based on their color palettes, such as warm or cold tones or strong contrast.
- Color Harmonization: Adjusting colors in an image to achieve a visually pleasing result.
- Real-time Color Detection; A system that can instantly recognize colors using live camera feed. This can be used in augmented reality, object detection, and interactive art installations, among many fields.
- Integration with other systems to develop more robust and flexible image analysis systems.

## Reference

Anjali, K. (2001, January 3). *Detecting Specific Color From Image using Python OpenCV*. Detecting Specific Color From Image Using Python OpenCV. https://www.etutorialspoint.com/index.php/329-detect-specific-color-from-image-using-python-opencv

Rosebrock, A. (2014, August 4). *OpenCV and Python Color Detection - PyImageSearch*. PyImageSearch. https://pyimagesearch.com/2014/08/04/opencv-python-color-detection/

*Colorblind People Population! Statistics*. (2021, June 13). Colorblind People Population: https://www.colorblindguide.com/post/colorblind-people-population-live-counter

*Colour Blindness Research &mdash; Dr John Barry*. (2020, November 10). Dr John Barry. https://www.johnbarrypsychologist.com/colourblindness

(*A Guide to Bhutan's Colourful Festivals | Luxury African Safaris,South America & South Asia Tours|andBeyond*, 2023): https://www.andbeyond.com/advice/asia/a-guide-to-bhutans-colourful-festivals/

*pyGuru123 - Overview*. (n.d.). GitHub. https://github.com/pyGuru123

*Color detection.zip*. (n.d.). Google Docs. https://drive.google.com/file/d/1FEKZ7SsbLVrS__QeVb_Ehrp5dDBfuplp/view?usp=embed_facebook