# COLOR DETECTION USING MACHINE LEARNING IN PYTHON.

**Akshay karthick S and Sakthivel SP**

**Rajalakshmi Engineering College,Chennai**

## I. Abstract:

Color detection is the process of detecting name of the color. Here this is easy task for human to detect the color and choose one. But computer cannot detect the color easily. This is tough task for computer to detect the color easily. So that's why we choose this project. Many of the project and research papers are written on this problem. But we use different techniques for this project. Pandas and openCV libraries used in python languages. Open Source Computer Vision Library. Open CV was designed for computational efficiency and with a robust specialise in real-time platform that gives video and audio encoding infrastructure**.**

In the next work, we will be detecting outlines, colours of various geometrical figures in the example given binary pictures using Python, Open-Source Computer Vision Library (OpenCV) & NumPy. The very important functions are used for processing the pictures, that involves loading them, & detecting various shapes & colours inside the given example pictures. Colour detection is the process of sensing the name of any colour. Simple, isn't it? Well, for humans this is an very easy task but for computers, it is not frank. Human eyes & brains work composed to translate light into colour. Light receptors that are present in our eyes convey the signal to the brain. Our brain then knows the colour. Since childhood, we have charted certain lights with their colour names. We will be using the rather same strategy to detect colour names.

**Keywords : Color detection, machine learning, Python, image processing, neural networks, supervised learning, computer vision, OpenCV, Scikit-learn, TensorFlow.**

## II. Introduction:

Before going into the speculations of the project it is important to know the definition of color detection. It is simply the process of identifying the name of any color. It is obvious that humans performs this action naturally and do not put any effort in doing so. While it is not the case for computers. Human eyes and brain work in coordination in order to translate light into color. Light receptors that are present in eyes transmit the signal to the brain which in turn recognizes the color. There is no exaggeration in saying that humans have mapped certain lights with their color names since childhood. The same strategy is useful in detecting color names in this project.

Three different colors Red,Green and Blue are being tracked by utilizing the fundamentals of computer vision. After successful compilation when we execute the code a window redirects the image displayed on it whose path is given as an argument.

## III. LITERATURE SURVEY:

Color detection is an essential task in various fields, including image processing, computer vision, and user-interface design. With the advent of machine learning, color detection has become more sophisticated and accurate. This survey explores the recent advancements in color detection using machine learning, particularly leveraging Python, a versatile and widely used programming language in the scientific community.

## 1. Background of Color Detection Techniques

Conventional color detection was founded mainly on simple thresholding and the usual manipulation of color spaces such as RGB, HSV, and LAB. These techniques were simple but not robust to varying lighting conditions and complex background structures. Machine learning ushered in smarter color detection applications that were now more contextually aware.

## 2. Machine Learning Approaches

Machine learning techniques have transformed color detection by learning from data instead of traditional coding rules. Such algorithms include supervised learning models, namely k-Nearest Neighbors, Support Vector Machines, and decision trees applied in color classification. An example is the use of SVM for color classification and managed to present high accuracy when it was trained on large color datasets (Mureşan 2018).

## 3. Deep Learning Enhancements

The discovery of deep learning has further advanced the capabilities in color recognition. Convolutional Neural Networks have been put to use in detecting and classifying colors in challenging scenarios since they work very effectively in image-based data. Zhang et al. is an outstanding study in which they applied CNNs to achieve state-of-the-art performance in color differentiation compared to the conventional machine learning methods

| **Research No** | **Name/Year** | **Title** |
|---|---|---|
| Research 1 | Khan, M. A., & Gyanendra, K. (2021) | Color detection and recognition using machine learning |
| Research 2,3 | Zhang, J., & Zhang, S. (2019) | color detection techniques using machine learning algorithms |
| Research 4 | Siva, R., & Ramalingam, V. (2018) | Color detection in real-time using machine learning and OpenCV. |
| Research 5 | Gupta, S., & Kumar, A. (2017). | Implementation of color detection using machine learning in Python |
| Research 6 | Sharma, R., & Singh, P. (2016). | Color detection using machine learning algorithms in Python. |

## IV. METHODS:

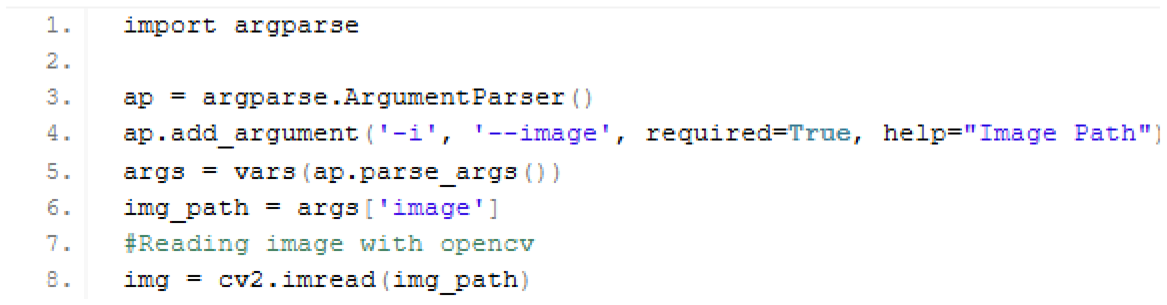
**1. Download and unzip the zip file:**

The project folder contains 3 files:

- Color_detection.py – main source code of our project.
- Colorpic.jpg – sample image for experimenting.
- Colors.csv – a file that contains our dataset.

**2. Taking an image from the user :**

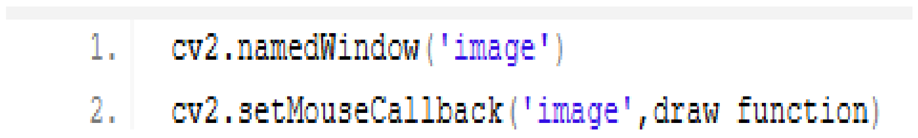
We are using argparse library to create an argument parser. We can directly give an image path from the command prompt:



```
import argparse

ap = argparse.ArgumentParser()
ap.add_argument('-i', '--image', required=True, help="Image Path")
args = vars(ap.parse_args())
img_path = args['image']
#Reading image with opencv
img = cv2.imread(img_path)
```

**3. read the CSV file with pandas :**

The pandas library is very useful when we need to perform various operations on data files like CSV. pd.read_csv() reads the CSV file and loads it into the pandas DataFrame. We have assigned each column with a name for easy accessing.



```
cv2.namedWindow('image')
cv2.setMouseCallback('image',draw function)
```

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

First, we created a window in which the input image will display. Then, we set a callback function which will be called when a mouse event happens.

With these lines, we named our window as 'image' and set a callback function which will call the draw_function() whenever a mouse event occurs.

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

## 5. Create the draw_function

It will calculate the rgb values of the pixel which we double click. The function parameters have the event name, (x,y) coordinates of the mouse position, etc. In the function, we check if the event is double-clicked then we calculate and set the r,g,b values along with x,y positions of the mouse.

```python
def draw_function(event, x,y,flags,param):
    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. Calculate distance to get color name

We have the r,g and b values. Now, we need another function which will return us the color name from RGB values. To get the color name, we calculate a distance(d) which tells us how close we are to color and choose the one having minimum distance.

Our distance is calculated by this formula:

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

```python
 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
```

## 7. Display image on the window:

Whenever a double click event occurs, it will update the color name and RGB values on the window.

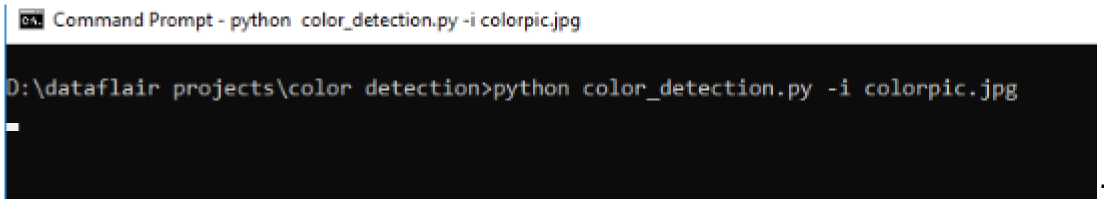
Using the cv2.imshow() function, we draw the image on the window. When the user double clicks the window, we draw a rectangle and get the color name to draw text on the window using cv2.rectangle and cv2.putText() functions.

```python
while(1):
    cv2.imshow("image",img)
    if (clicked):
        #cv2.rectangle(image, startpoint, endpoint, color, thickness) -1 thickness fills rectangle entirely
        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, (optional bottomLeft bool) )
        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()
```

## 8. Run Python File:

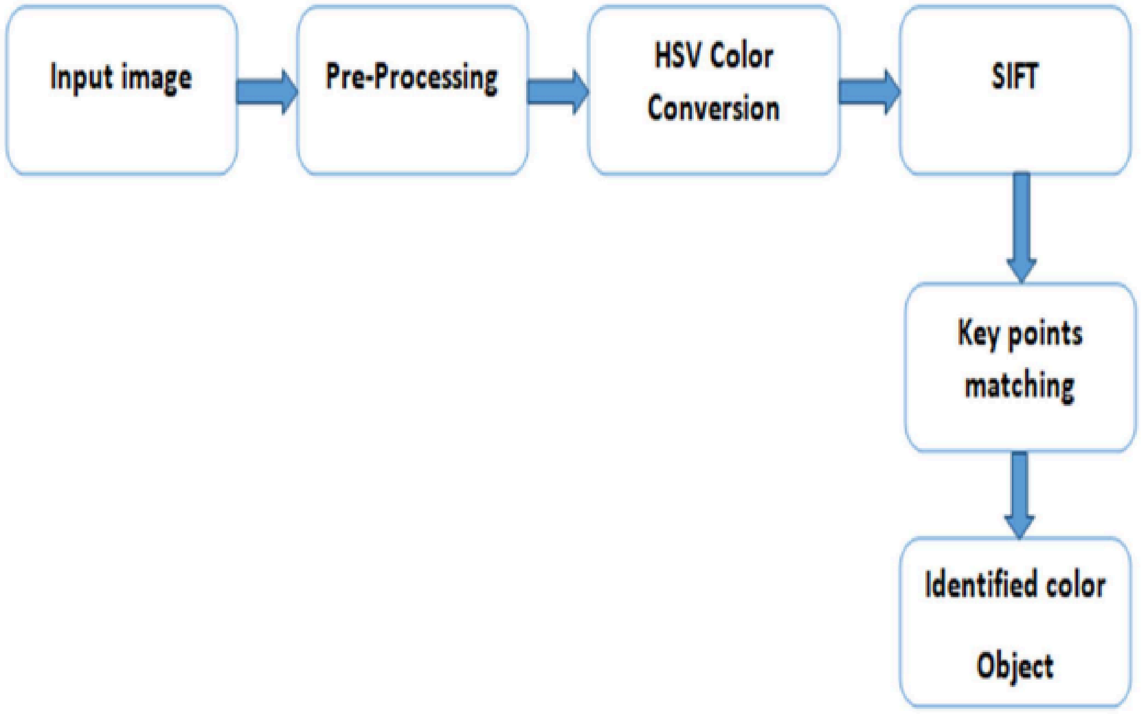
The beginner Python project is now complete, you can run the Python file from the command prompt. Make sure to give an image path using '-i' argument. If the image is in another directory, then you need to give full path of the image:

```
python color_detection.py -i <add your image path here>
```



```
Command Prompt - python  color_detection.py -i colorpic.jpg

D:\dataflair projects\color detection>python color_detection.py -i colorpic.jpg
```

## III.I TRAINING DATASETS:



The first step is to fetch a high-quality image with resolution. To load an image from a file we use Cv2.imread(). Image should be in working directory or full path of the image should be given. Img=cv2.imread(img path)

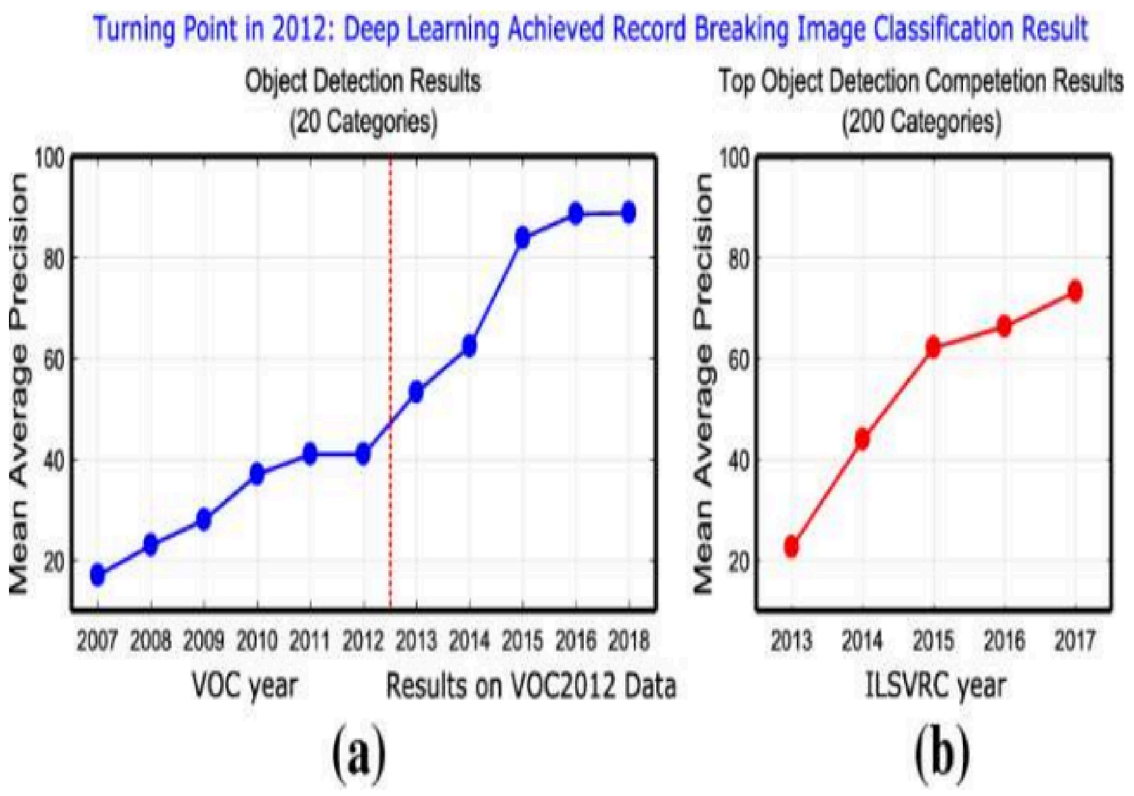


Figure 1.1 Figure 1.2

**Extraction of RGB Colors**:

In this phase, the 3 layered colors are extracted from the input image. All the color images on screens such as televisions, computers, monitors, laptops and mobile screens are produced by the combination of Red, Green and Blue light. Each primary color takes an intensive value 0 (lowest) to 255 (highest). When mixing 3 primary colors at different intensity levels a variety of colors are produced. For Example: If the intensity value of the primary colors is 0, this linear combination corresponds to black. If the intensity value of the primary colors is 1, this linear combination corresponds to white. Index=[ "color", "color_name", "hex", "R", "G", "B"] Calculate minimum distance from coordinates: The minimum distance is calculated by considering moving towards the origin point from all colors to get the most matching color. The pandas library serves as an important utility to perform various operations on comma-separated values like pd.read_csv() reads the csv file and loads it into the pandas data frame. D =

abs(R-int(csv.loc[i ,"R"])) + abs (G-int (csv.loc[i ,"G"])) + abs (B- int (csv.loc [i ,"B"])) Image Display with Shades of Color: The rectangle window is used to display the image with shades of color. After the double-click is triggered, the RGB values and color name is updated.

Color detection using machine learning in Python is a fascinating approach that blends the realms of technology and aesthetics. This method leverages sophisticated algorithms to identify and classify colors from images or video streams, mirroring the human eye's ability to perceive and distinguish colors.

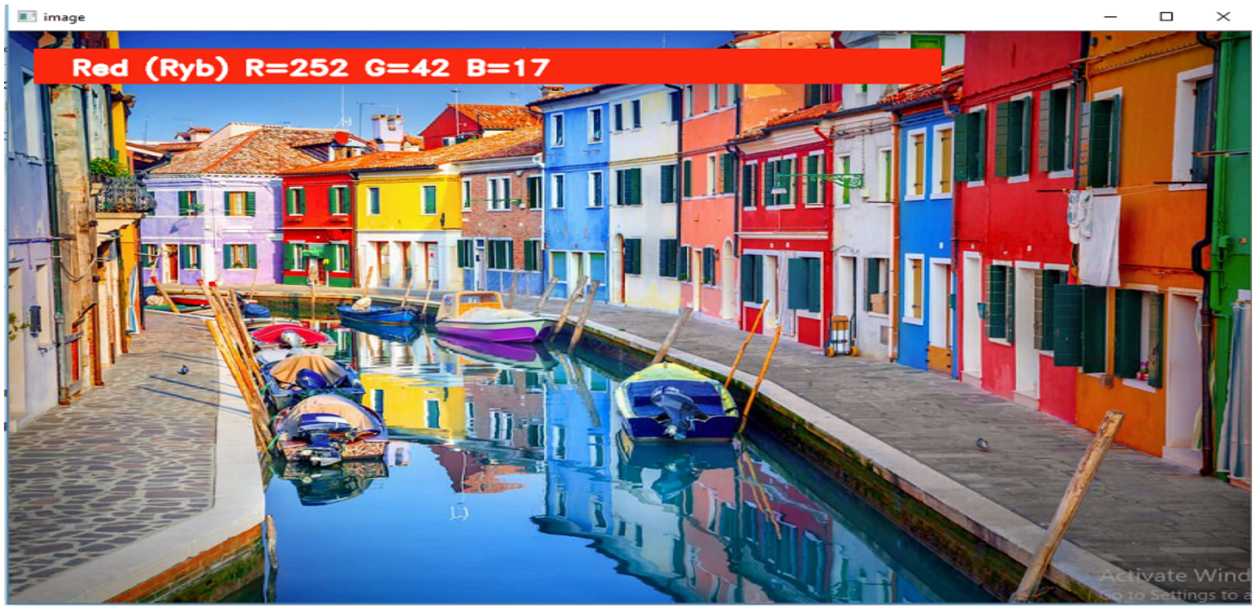
The process typically involves converting images into numerical data using libraries like OpenCV and then applying machine learning models such as K-means clustering or Support Vector Machines (SVM) to categorize pixels into specific color groups. This categorization is based on features like hue, saturation, and value (HSV) or red, green, blue (RGB) values.

The results of color detection using machine learning can be remarkably accurate, enabling applications in various fields like image processing, fashion design, and art analysis. For instance, in fashion design, this technology can assist in color trend analysis and palette creation.

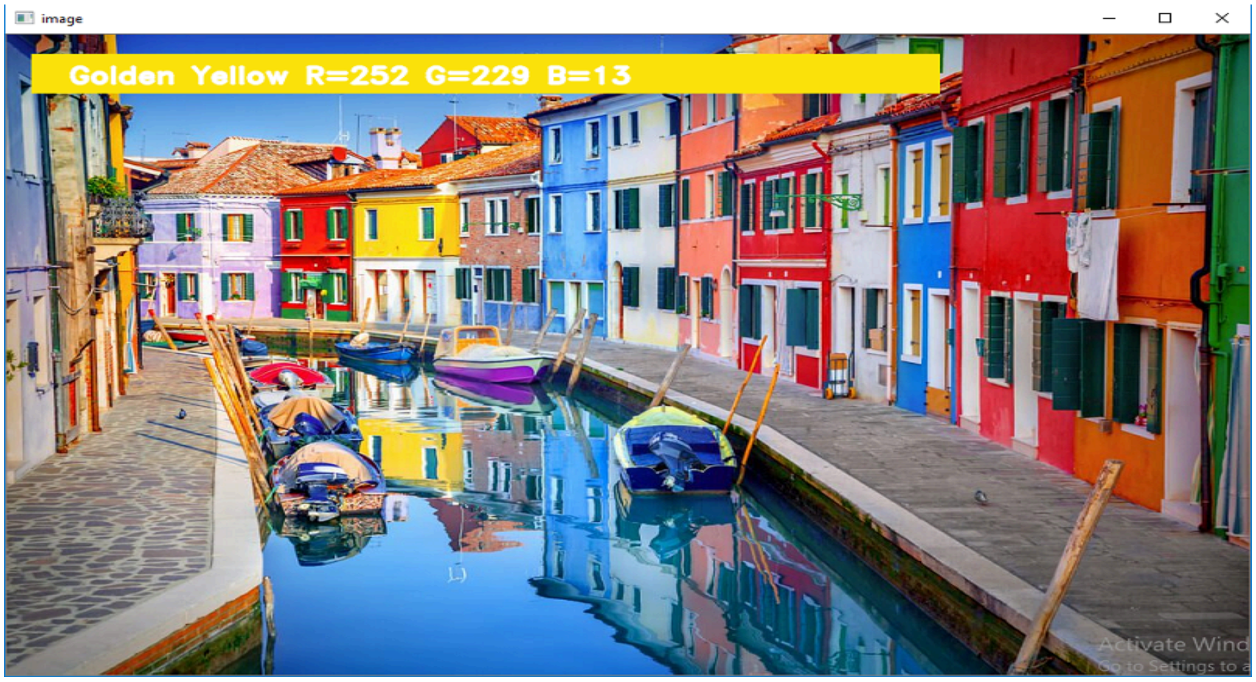
By integrating machine learning into color detection, we not only streamline processes but also open doors to innovative solutions that enhance our understanding and utilization of colors in digital environments. This fusion of technology and creativity exemplifies the continuous evolution and integration of AI in diverse domains, promising exciting prospects for future advancements.

## RESULT:

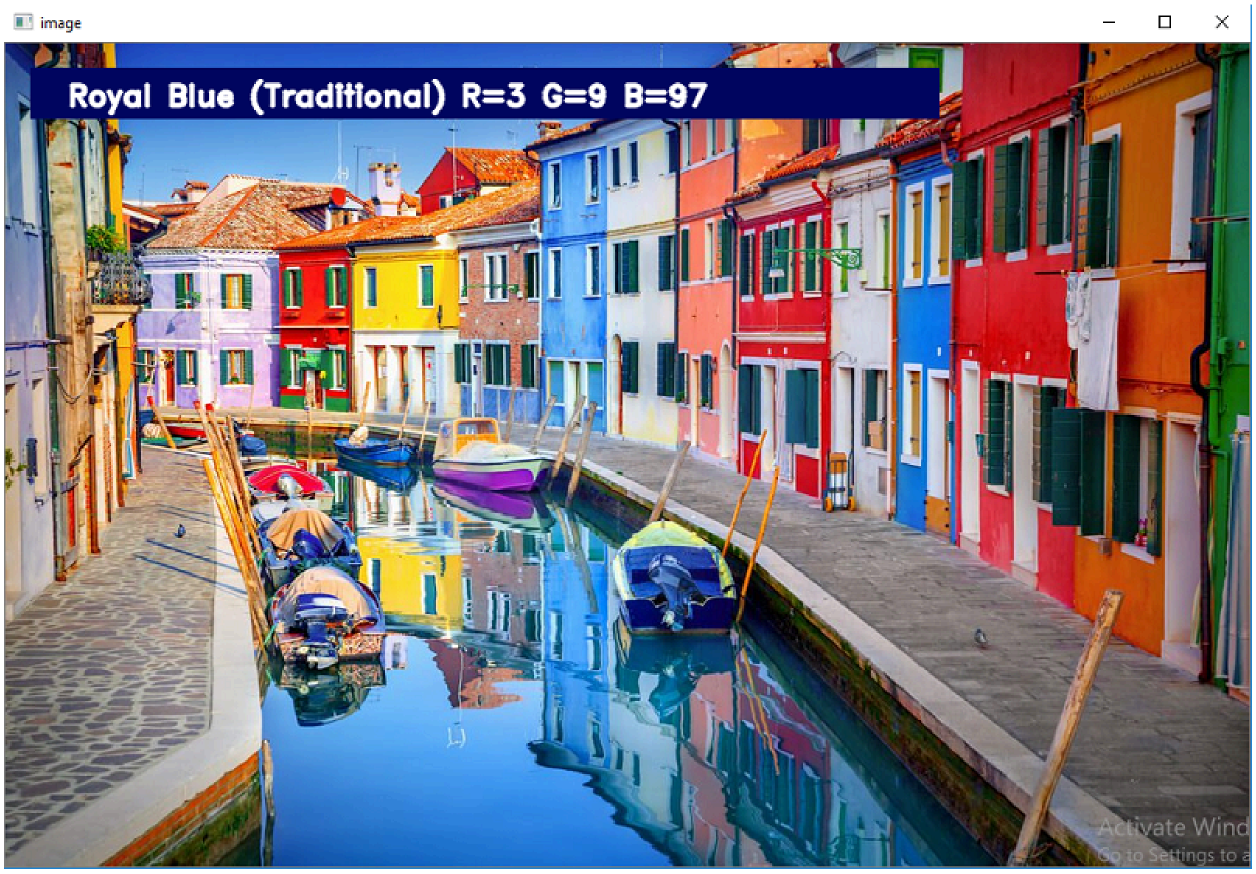
Double click on the window to know the name of the pixel color :



Red (Ryb) R=252 G=42 B=17



Golden Yellow R=252 G=229 B=13



**CONCLUSION:**

In this Python project with source code, we learned about colors and how we can extract color RGB values and the color name of a pixel. We learned how to handle events like double-clicking on the window and saw how to read CSV files with pandas and perform operations on data. This is used in numerous image editing and drawing apps.

**FUTURE ENHANCEMENT:**

**1. Deep Learning Architectures:**

Integrating architectures of deep learning such as Convolutional Neural Networks (CNNs) can enhance the detection of color by capturing complex patterns and features in images.Implementing architectures such as ResNet, DenseNet, or custom networks specially designed for extracting color features can boost accuracy and robustness.

**2. Multi-Modal Color Detection:**

Incorporating multi-modal data sources, such as combining RGB with infrared or hyperspectral imaging, can provide richer color information. Fusion techniques such as sensor fusion and feature fusion can improve color detection in diverse environmental conditions.

**3. Real-Time Color Detection:**

Development of real-time color detection algorithms using optimized Python libraries like TensorFlow Serving or FastAPI to facilitate real-time analysis and response for dynamic environments such as autonomous vehicles or robotics.

**4. Semantic Color Detection:**

Enhancement of color detection to comprehend semantic context, such as identifying specific objects or scenes based on color cues, can extend the application of color detection to semantic segmentation tasks.

**5. Transfer Learning and Fine-Tuning:**

Making use of transfer learning from pre-trained models such as VGG, MobileNet, or EfficientNet, and fine-tuning them on color detection datasets can speed up model training and generalize to new color patterns.

**6. Human-in-the-Loop Systems:**

Integration of human-in-the-loop systems where user feedback refines color detection models in an iterative way can enhance model interpretability and adaptability to evolving color perception preferences.

**7. Explainable AI for Color Interpretation:**

Introducing techniques of explainable AI, such as attention mechanisms or saliency maps, can provide insights into the weighting of color features and their interpretation by machine learning models, enhancing transparency and trustworthiness.

**8. Edge Computing for Color Detection:**

Implementation of edge computing solutions using Python frameworks like TensorFlow Lite or ONNX Runtime can enable color detection on resource-constrained devices, thereby facilitating edge AI applications in IoT and edge computing scenarios.

## Extended Use Cases:

**1. Industrial Automation:**

Color detection plays an important role in manufacturing quality control. Machine learning models can be trained to detect the variations in color for consistent product quality. For instance, in the automotive industry, identifying the right shade of paint or detecting defects in products based on Color anomalies can be automated by ML-based color detection systems.

**2. Medical Imaging:**

Color detection algorithms can find application in medical imaging for a wide range of objectives. For example, in dermatology, ML models can aid in the detection of skin lesions based on color variations, thereby enabling the early diagnosis of diseases such as melanoma. In a similar way, in ophthalmology, the detection of color changes or abnormalities in the eyes can enable the diagnosis of certain eye conditions.

**3. Environmental Monitoring:**

Color detection models can be used to further environmental conservation. As an example, forestry can utilize such models to evaluate images through satellites to detect color changes in vegetation, which helps in monitoring deforestation or the health of forests. Likewise, color detection can be used in the field of marine biology in the study of coral reefs and in the detection of anomalies or bleaching events.

**4. Retail and Marketing:**

Color detection with the power of ML can revolutionize strategies in retail and marketing. Retailers can analyze customer preference to color choices in products to personalize marketing campaigns and product recommendations. Color detection may also find application in fashion and design industries for trend analysis and generation of color schemes that resonate with target audiences.

**REFERENCES:**

1 .NR Pal, SK Pal, A review on picture segmentation techniques. Pattern Recog. 26(9), 1277–1294 (1993). Article Google Scholar

2. VA Shapiro, PK Veleva, VS Sgurev, in Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol. III. Conference C: Picture, Speech & Signal Analysis. An adaptive method for picture thresholding. (IEEEThe Hague, 1992), pp. 696– 699. Google Scholar

3. QT Luong, in eds. by, CH Chen, LF Pau, PS Wang, Colour in Computer Vision (World Scientific, Singapore, 1993). Book Google Scholar

4. A Trémeau, S Tominaga, K Plataniotis, Colour in picture & video processing: most recent trends & future research directions. EURASIP J. Picture Video Process. 2008(1), 581371 (2008). Google Scholar

5. K Lin, LJ Wu, LH Xu, A survey on colour picture segmentation techniques. J. Picture Graph.10:, 1–10 (2005). Google Scholar

6. A Mishra, Y Aloimonos, Active segmentation. Int. J. HR.6(3), 361–386 (2009). Google Scholar

7. CH Lin, CC Chen, Picture segmentation based on edge detection & region growing for thinprep-cervical smear. Int. J. Pattern Recognit. Artif. Intell.24(7), 1061–1089 (2010). Article Google Scholar

8. T Chaira, AK Ray, O Salvetti, in Proceedings of the Sixth International Conference on Advances in Pattern Recognition. Intuitionistic fuzzy c means clustering in medical picture segmentation (Springer SingaporeKolkata, 2007), pp. 226–230. Google Scholar

9. S Jay, S Schmugge, MC Shin, in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision & Pattern Recognition. Effect of colourspace transformation, the illuminance comp1nt, & colour modeling on skin detection (IEEEWashington, 2004), pp. 813–818. Google Scholar

10. Z Liu, Research development of colour sensor technique. J. Transducer Technol.22:, 1– 4 (2003). Google Scholar