# PROFESSIONAL TRAINING REPORT

AT

Sathyabama Institute of Science and Technology

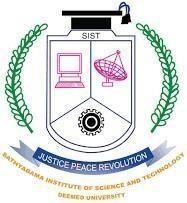
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

## Sravya Namburu

## (Reg. No:39110962)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU**

**April 2022**

 **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

### (DEEMED TO BE UNIVERSITY)

**Accredited with Grade “A” by NAAC**

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI– 600119

[**www.sathyabama.ac.in**](http://www.sathyabama.ac.in)

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the Bonafide work of **Sravya Namburu(Reg.No: 39110962)** who carried out the project entitled **“COLOR RECOGNITION USING OPEN CV”** under our supervision from January 2022 to may 2022.

## Internal Guide

**Dr. A. Viji Amutha Mary, M. Tech., Ph.D.,**

**Head of the Department**

**Dr.L.Lakshmanan, M.E., Ph.D.,**

**Dr.S.Vigneshwari, M.E., Ph.D.,**

**Submitted for Viva voice Examination held on**

**Internal Examiner External Examiner**

**DECLARATION**

I **SRAVYA NAMBURU (Reg. No: 39110962)** hereby declare that the Project Report entitled **“COLOR RECOGNITION USING OPEN CV”**done by me under the guidance of **Dr. A. Viji Amutha Mary, M.Tech., Ph.D**is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.

|  |  |
| --- | --- |
| **DATE:** | Sravya.N |
| **PLACE:** CHENNAI | **SIGNATURE OF THE CANDIDATE** |

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D.**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. Viji Amutha Mary M. Tech., Ph.D, for** his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

**COURSE COMPLETION CERTIFICATE**

# ABSTRACT

Color recognition is the process of detectingname of the color. Here this is easy task for humans to detect thecolor and choose one. But computer cannot detect the color easily.

The main objective of the application is the methodology for identifying the shades of colors with an exact prediction with their names.

A study says a normal human can able to clearly identify nearly 1 million shades of colors. But in the case of human having “enchroma”, could be able to see only 1% (i.e.10, 000 colors) from the normal humans. While painting pictures, a painter needs to identify the color patterns exactly or else the reality of image is not clear

Recognizing the colors done by using some machine learning libraries Numpy, Pandas and Opencv .

OpenCV (OPEN SOURCE COMPUTER VISION LIBRARY) , it Was designed for computational efficiency and with robust specialize in real time applications.

Keywords – OpenCV , color recognition, Numpy, pandas, python libraries.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| *CHAPTER NO* | TITLE | PAGE NO |
|  | ABSTRACT | vi |
|  | LIST OF FIGURES | ix |
|  | LIST OF ABBREVATIONS | x |
| 1 | INTRODUCTION | 01 |
|  | * 1. GENERAL   2. OBJECTIVES   3. SCOPE | 01 |
|  |  | 01 |
|  |  | 01 |
|  |  | 01 |
|  |  | 02 |
| 2 | METHODS AND MATERIAL | 03 |
|  | 2.1 IMAGE CAPTURE | 03 |
|  | 2.2 EXTRACTION OF RGB COLORS | 04 |
|  | 2.3 HARDWARE&SOFTWARE REQURIMENTS | 04 |
|  | 2.4 DESIGN CRITERIA  2.5 METHODOLOGY | 05 |
|  |  | 06 |
|  |  | 07 |
| **3** | SYSTEM ARCHITECTURE | 07 |
|  | 3.1 ARCHITECTURE | 08 |
|  | 3.2 MODULES | 09 |
|  | 3.3 OPENCV  3.4 IMPLEMENTATION | 10 |
| 4 | RESULTS AND DISCUSSION | 11 |
| 5 | CONCLUSION AND FUTURE WORK | 12 |
|  | SOURCE CODE | 13 |
|  | REFERENCES | 13 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Fig No | Fig Name | Page No |
| 1 | IMAGE | 14 |
| 2 | DATASET IMAGE | 15 |

3 ARCHITECTURE 16

4 CODE IMAGES

5 OUTPUT IMAGES

**LIST OF ABBREVIATIONS**

OPENCV-OPEN SOURCE COMPUTER VISION

NP- NUMPY

PD- PANDAS

**CHAPTER 1**

**INTRODUCTION**

**1.1 GENERAL**

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 perform 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 co-ordination 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 to the image displayed on it whose path is given as an argument.

Additionally, we obtain the color name of the pixel along with the composition of three different colors red, blue and green values. It is helpful in recognizing colors and in robotics. One of the applications of color detection by computer vision is in driver less cars. This system is useful in detecting traffic and vehicle backlights and takes decision to stop, start and continue driving. This also have much application in industry to pick and place different colored object by the robotic arm. Color detection is also used as a tool in various image editing and drawing apps.

**1.2OBJECTIVE**

Aim of the this project is to build color recognizer using Open CV .

Color recognition using Open CV also allows the identifying of a

Specific coloring live stream video. To create a color recognition

App which will help users to detect color in the image .We will use

three main modules for this project Numpy, Pandas, Open Cv.

Open CV is a highly optimized library with a focus on real time

applications.Our aim is to retrieve RGB values from an image and

determine the color.

**1.3 SCOPE OF THE PROJECT**

Computer Vision – Color recognition is the basic and important

step for proceeding in computer vision.

Object Segregation – An object can be segregated (separated) on

the basis of color.

Object Tracking – A moving object can be tracked on the basis of its

color.

**CHAPTER 2**

**METHODS AND MATERIAL**

# 2.1 IMAGE CAPTURE:

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)

# 2.2 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, computer, 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.

**2.3 HARDWARE AND SOFTWARE REQURIMENTS**

\* Programming language: Python 3.7.0 and above versions

\* Hardware requirements: CPU

\* Software requirements: Microsoft Windows10

**2.4 DESIGN CRITERIA**

\* Simplicity: easily understood

\* Efficiency: uses minimal resources

\* Completeness: solves the problem

\* Not independent

\* Simplicity by default

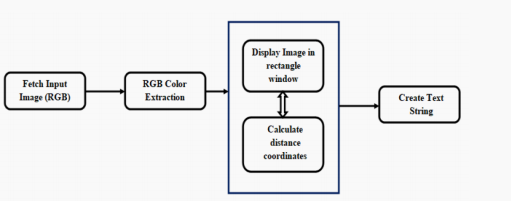
**2.5 METHODOLOGY**

I recently started working on image color detection using python with OpenCV use for detection color. When I found OpenCV which allows import and manipulation of images in Python shown that it might be extracted out of these images using Machine Learning and utilized in how. We’ve all seen that we will search online on the idea of certain filters one among which is color. I got inspired to truly write the code which will extract colors out of images and filter the pictures supported those colors. In this research paper, I explain how I understood the fundamentals of OpenCV, extracted colors from images using machine learning libraries and filtered images from a set of images supported RGB values of colors.

# CHAPTER3

**SYSTEM ARCHITECTURE**

**3.1 ARCHITECTURE**



*Architecture Diagram*

The above architecture shows the capability for the project. It consists of a well defined sequence diagram that is abstracted from the source code. It leverages the rich capabilities of the technology such as OpenCv library in python. The above architecture makes the process more efficient based on principles and properties related to each other. As we know that Red, Green and Blue are the primary colors that can be mixed to produce different colors. The present color detection project

takes the path of an image as an input and looks for the composition of three different colors red, green and blue in the given image*.*

**3.2 MODULES**

Numpy - Supports large Multi-dimesional arrays and matrices, along with a large collection of high-level mathematical function to operate on the arrays.

Numpy is a general purpose array preprocessing package.

Pandas – Pandas is a software library written for the python programming language for data manipulation and analysis.

Matplotlib – Matplotlib is a 2D plotting library for the python programming language and its numerical mathematics extension Numpy.

CV2 – Open CV is a library of programming function at real time computer vision and it is a python wrapper. It makes use of Numpy

**3.3 OPENCV**

OpenCV-Python is **a Python wrapper for the original OpenCV C++ implementation**. OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays.

**OpenCV** is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e. whatever operations one can do in Numpy can be combined with OpenCV.

The purpose of computer vision is to understand the content of the images. It extracts the description from the pictures, which may be an object, a text description, and three-dimension model, and so on. For example, cars can be facilitated with computer vision, which will be able to identify and different objects around the road, such as traffic lights, pedestrians, traffic signs, and so on, and acts accordingly.

# 3.4 IMPLEMENTATION

In this project first step is library installation; it is very easy step using pip. After the installation is completed, we have to import them to our program. Open a new file in favorite code editor. OpenCv is imported as cv2 and for other libraries we imported as so that it is easier to call them in the program.

Now we choose any image we want and save the image in the same folder as program, which makes it easier to find and import.

**

First, we have to teach them colors. To do that we need data that includes color names and some values to match with those colors. Since most of the colors can be defines using Red, Green, and Blue. That’s why we will use the RGB format as our data points. Dataset with 1000 color names and the RGB values.



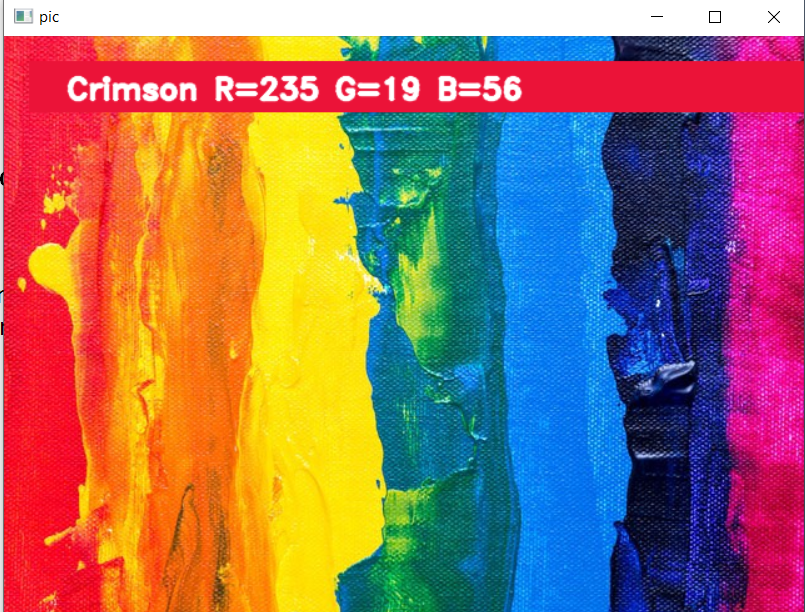
To make the application work smoothly, we need some global variables and we define color recognition function and it will be called when we double click on the area of the image. It will return the color name and RGB values of that color. And other function called Mouse click function to define our double click process.

If we worked with OpenCV projects. We have to define how to end and close the window.

**CHAPTER 4**

**RESULTS AND DISCUSSION**

We will get the result of color names and RGB values of the color by clicking on the image using this color recognizer application.



**Chapter 5**

**CONCLUSION AND FUTURE WORK**

**Conclusion:**

In this paper we defined to get the required color field from an RGB image. In this various steps are implemented using openCv platform. The main positive point of this method is its color

differentiation of a mono color. In the future scope, the detection of the edge detection techniques has different other applications like facial detection, color conversion for grey scale

image etc. that can also be implemented.

# FUTURE WORK:

In existing system there is no exact color representation of colors with accuracy. In proposed system, we are introducing the CV datasets and according to it the number of shades that can be

identified using 865 color names along with their RGB and hex values. Whenever the cursor clicks the image, it automatically shows the RGB shades color values. Proposed system uses OpenCv for sorting of primary colors.

**CHAPTER 6**

**REFERENCES**

G.M. Snoek, “Evaluating Color Descriptors for Object and Scene Recognition”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 32,

NO. 9, SEPTEMBER 2010

Claudia Nieuwenhuis, “Spatially Varying Color Distributions for Interactive Multi Label Segmentation”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL 35, NO. 5, MAY 2013

Kok -Meng Lee, “Effects of Classification Methods on Color-Based Feature Detection with Food Processing Applications”, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, VOL. 4, NO. 1, JANUARY 2007

# SOURCE CODE

pip install numpy pandas opencv-python

import numpy as np

import pandas as pd

import cv2

import matplotlib.pyplot as plt

img\_path = r'C:\Users\dell\Downloads\image.jpeg'

img = cv2.imread(img\_path)

plt.imshow(img)

index=["color", "color\_name", "hex", "R", "G", "B"]

csv = pd.read\_csv("colors.csv", names=index, header=None)

csv.head()

clicked = False

r = g = b = xpos = ypos = 0

def colorname(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, "colorname"]

return cname

def mousefunction(event, x, y, flags, param):

if event == cv2.EVENT\_LBUTTONDBLCLK:

global b, g, r, x\_pos, y\_pos, clicked

clicked = True

x\_pos = x

y\_pos = y

b, g, r = img[y, x]

b = int(b)

g = int(g)

r = int(r)

cv2.namedWindow('pic')

cv2.setMouseCallback('pic', mousefunction)

while True:

cv2.imshow("pic", img)

if clicked:

# cv2.rectangle(image, start point, 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 = colorname(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)

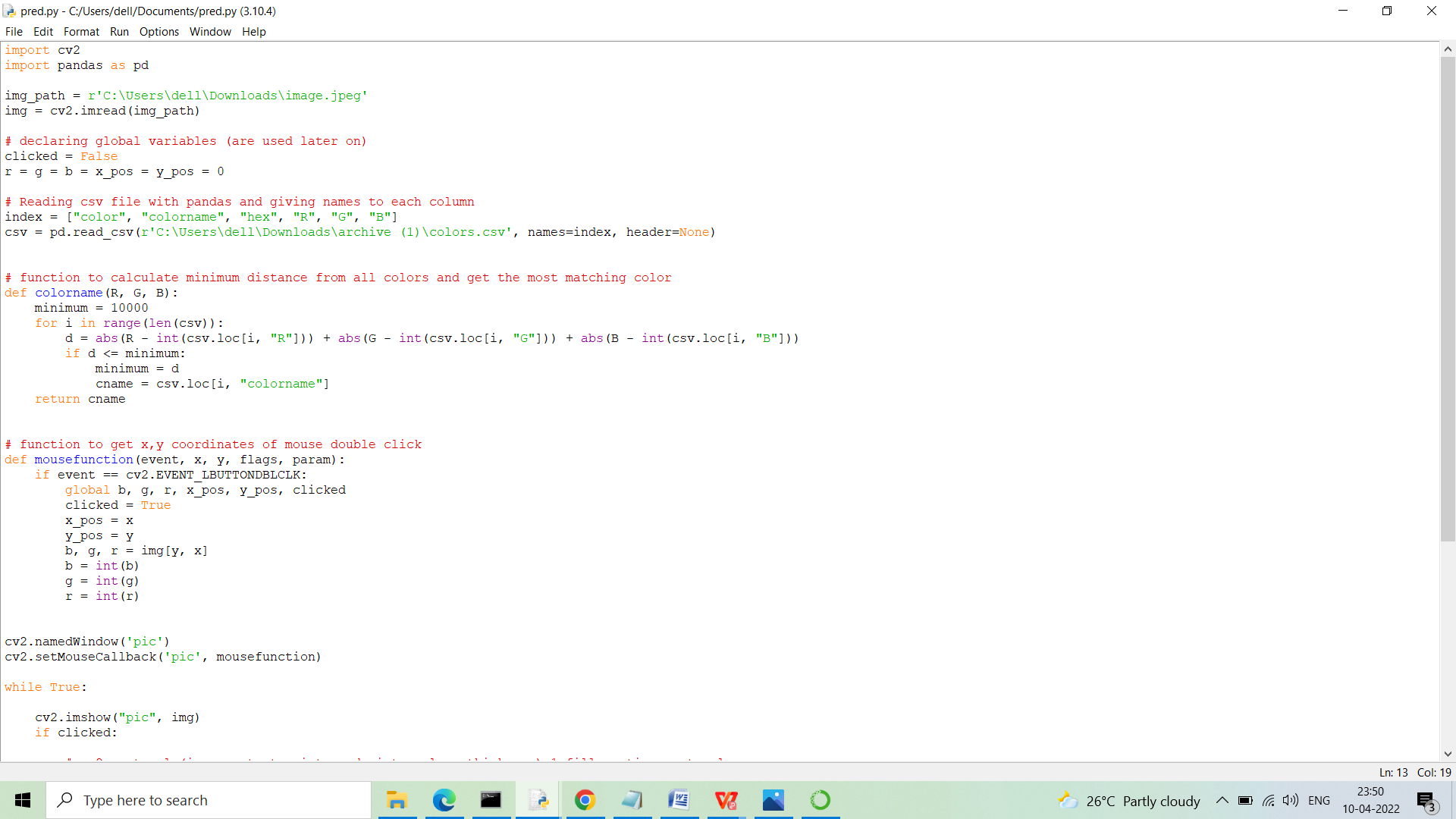
clicked = False

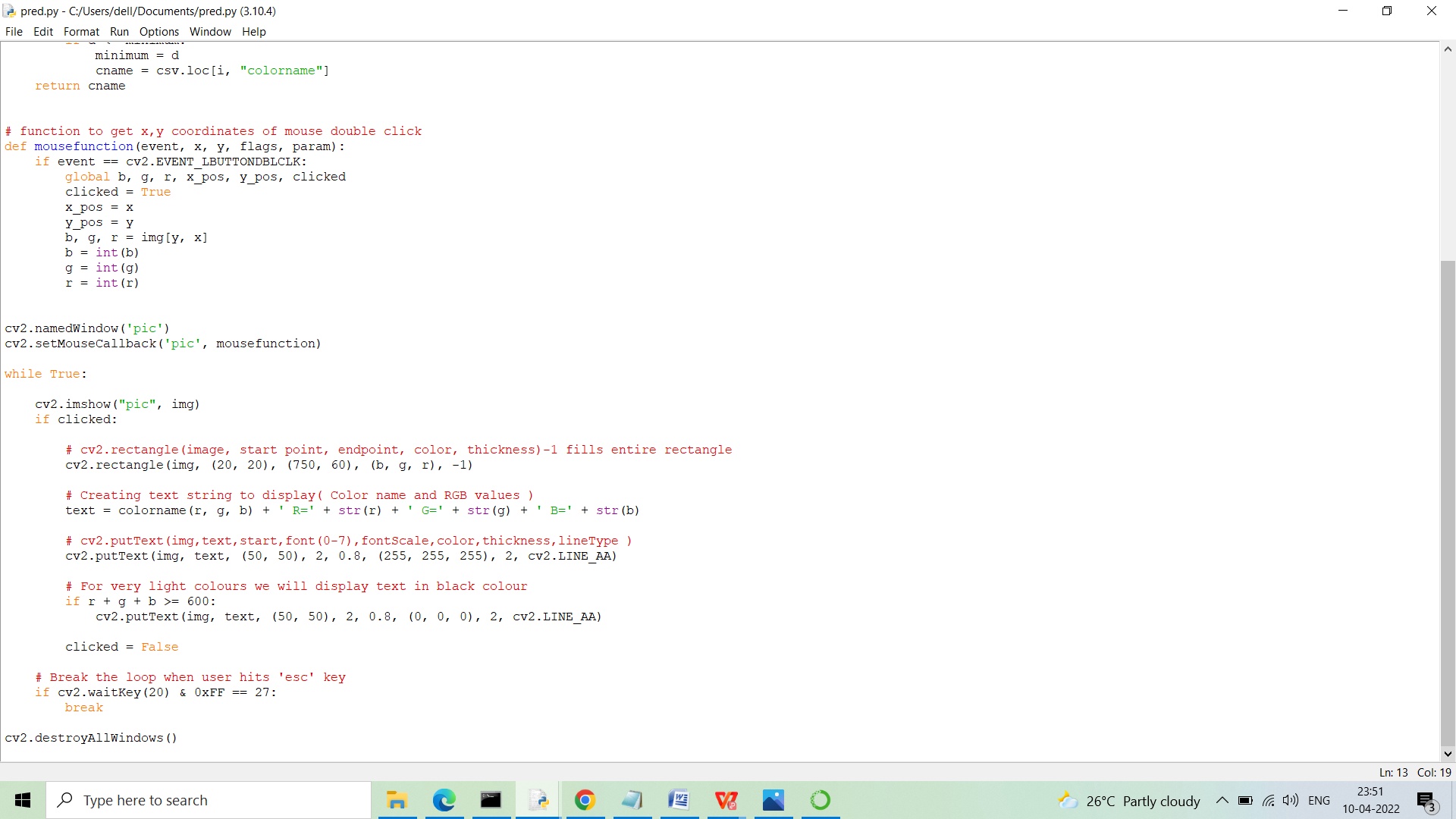
# Break the loop when user hits 'esc' key

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

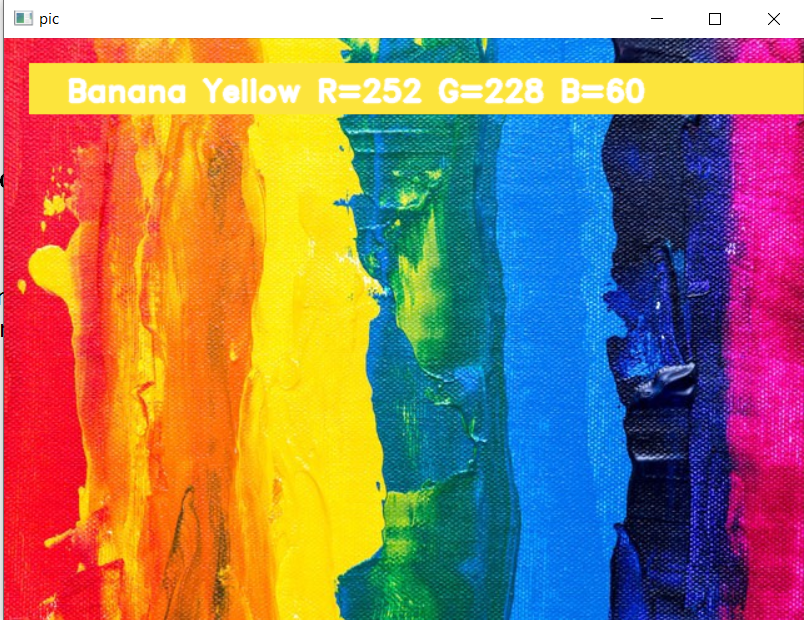
break

cv2.destroyAllWindows()

******

******

******

******