D.Y. PATIL COLLEGE OF ENGINEERING, AKURDI, PUNE

Color Detection Using Python and OpenCV

A Report Submitted for Partial Fulfilment of

Course-

Project Based Learning

First Year Engineering Programme of SPPU, Semester-II

(2023-24)

Submitted By:

Roll No.	Name of Student
FE22AI141	Aditya Pote
FE22AI152	Aditya Nanaware
FE22AI159	Parth Shinde
FE22AI162	Yash Mane
FE22AI166	Abhishek Nangare

Under the Guidance of

Prof. Pramila Karale

June-July 2023

CERTIFICATE

This is to certify that, report **“Color Detection Using Python and OpenCV”** submitted by Aditya Pote, Aditya Nanaware, Parth Shinde, Yash Mane and Abhishek Nangare of First Year Engineering Programme, is bonafide work completed in partial fulfilment of Course **Project Based Learning.**

Prof. Pramila Karale
Project Guide

Prof. Hnnie Williams
Academic Coocrdinator

Dr. Sanjay K. Babar
Dean- FE Section

Dr. Vijay M. Wadhai
Principal

SYNOPSIS

OpenCV is a Computer Vision library. It is a collection of C functions with a few C++ classes that implement popular Image Processing and Computer Vision algorithms. Computer vision is the science that means to give a comparative, if not better, capacity to a machine or PC. Computer vision is worried about the programmed extraction, investigation and comprehension of valuable data from a single picture or a grouping of pictures. Some of the basic image processing capabilities include filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms and many more. Color detection using OpenCV has many advantages like, it allows the detection of a specific color in a livestream video content. In this OpenCV color detection system there are four major modules, activated webcam, scan object, match frame parts and system results. Users can open webcam by clicking the webcam button. Then the algorithm analysis the pattern of the framed part of webcam. Pattern is matched with defined color pattern by RGB color model. If the pattern matched with the potential pattern of RGB color model then the system results with the correct output.

INDEX

Literature Review

[1] Color can be identified from the sensory optic nerves of the eyes. Color can only be seen or identified when a source of light is applied to an object. Color blindness can be termed as inability of the differentiation between colors. It is incurable disease that can be termed as lifelong disease. Edges can be very helpful in color differentiation boundary.

[2] Color detection model can be used in mixing of colors especially in paints, dyes and color pigments. It can be also very helpful in to differentiating colors that are used in robotics and in other medical fields.

It can also be used in Graphic Arts Industry. Other implementations can also be used in agricultural industry like especially detection of quality of soil.

[3] Color Detection can be used in agriculture industry to find the weeds the along with the crops. Via color detection weeds can be identified and destroyed and the crops can be saved. It can be also used in medical

industries to detect the disease and other disorders especially in face and other internal diseases like cancers.

[4] The main aim of computer vision is to analyze the behavior of human eye and the reduction of human effort. Through computer vision various task can be done that is done by human eye, whether to detect the object or identify its color. By this method it is very helpful to detect the symptoms of the disease and the other applications in other industries like agriculture.

Design and Development

a)Problem Formulation

- Import the modules necessary in solving the problem.

- Take the image path of which you want to find the colors in.

- Read the csv file in the hex code of colors is stored.

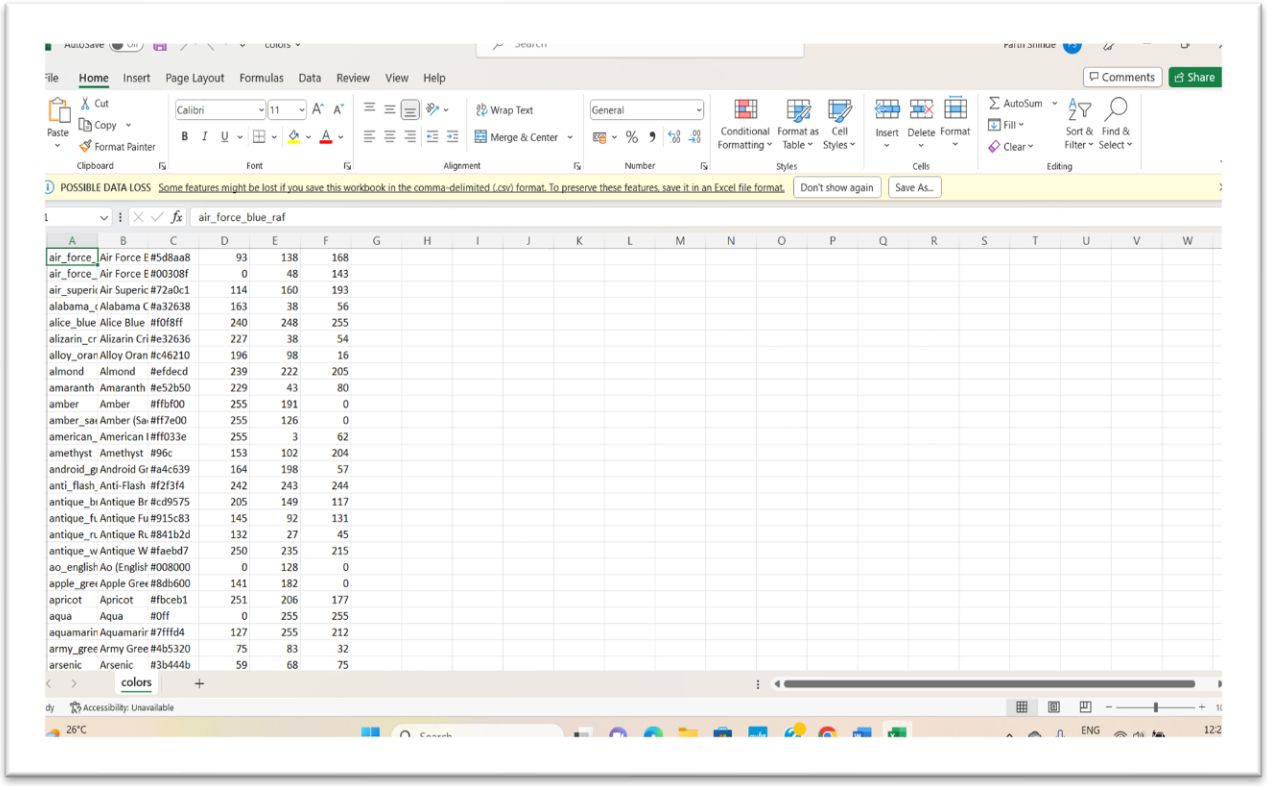
- Now calculate the minimum distance from all the color and get the most matching color using the get_color_name function.

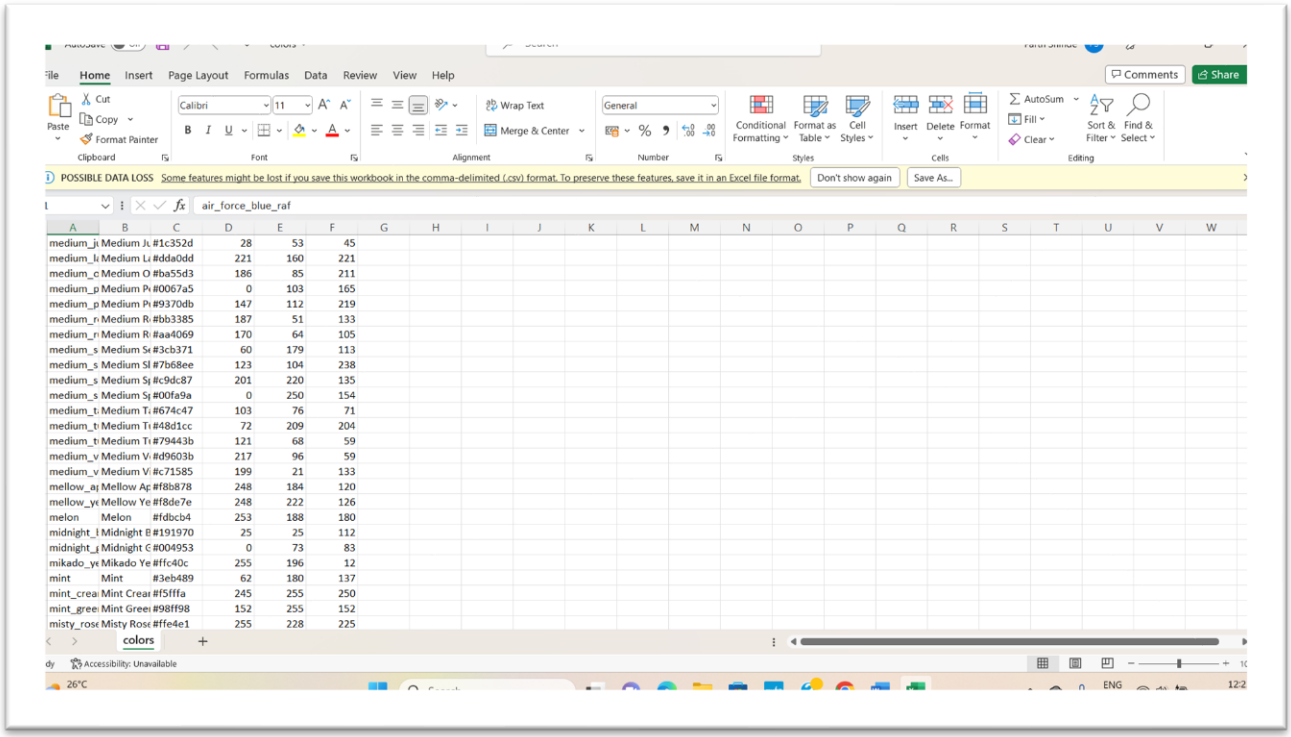
- Make a function to get the x, y coordinates of the click point of mouse.

- Display image on the window with a pointer to select any point on image.

- Display the color name of the point along with the R, G, B values

B)Dataset of colors





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*256*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.

Explanation of Code:

- **Camera Settings:** In order to perform runtime operations, the device's web-camera is used. To capture a video, we need to create a VideoCapture object. Its argument can be either the device index or the name of a video file. The device index is just the number to specify which camera. Normally one camera will be connected, so we simply pass 0. You can select the second camera by passing 1 and so on. After that, you can capture frame-by-frame. But in the end, don't forget to release the capture. Moreover, if anyone wants to apply this colour detection technique on any image it can be done with little modifications in the code which I'll discuss later.

- **Capturing frames:** The infinite loop is used so that the web camera captures the frames in every instance and is open during the entire course of the program.
  After capturing the live stream frame by frame we are converting each frame in BGR color space(the default one) to HSV color space. There are more than 150 color-space conversion methods available in OpenCV. But we will look into only two which are most widely used ones, BGR to Gray and BGR to HSV. For color conversion, we use the function cv2.cvtColor(input_image, flag) where flag determines the

type of conversion. For BGR to HSV, we use the flag cv2.COLOR_BGR2HSV. Now we know how to convert BGR
images to HSV, we can use this to extract a colored object. In HSV, it is more easier to represent a color than RGB colorspace.
In specifying the range , we have specified the range of blue color. Whereas you can enter the range of any colour you wish.

- **Masking technique:** The mask is basically creating some specific region of the image following certain rules. Here we are creating a mask that comprises of an object in blue color. After that, I have used a bitwise_and on the input image and the threshold image so that only the blue coloured objects are highlighted and stored in res.
  We then display the frame, res, and mask on 3 separate windows using imshow function.

- **Display the frame:** As imshow() is a function of HighGui it is required to call waitKey regularly, in order to process its event loop.
  The function waitKey() waits for key event for a “delay” (here, 5 milliseconds). If you don’t call waitKey, HighGui cannot process windows events

like redraw, resizing, input event etc. So just call it, even with a 1ms delay .

Summarizing the process:

1. Take each frame of the video.
2. Convert each frame from BGR to HSV color-space.
3. Threshold the HSV image for a range of blue color.

We need a function in the program that is going to return the color of the point where the mouse is clicked. And as we know for getting the color we need to get the distance and then compare it with the dataset. Calculating the distance is done by the formulae given below :-

D= abs (Red-ithRedColor) + abs (Green-ithGreenColor) + abs (BlueithBlueColor)

(ithRedColor, ithGreenColor, ithBlueColor are the colors R,G,B values from the dataset)

Required tools

OpenCV:

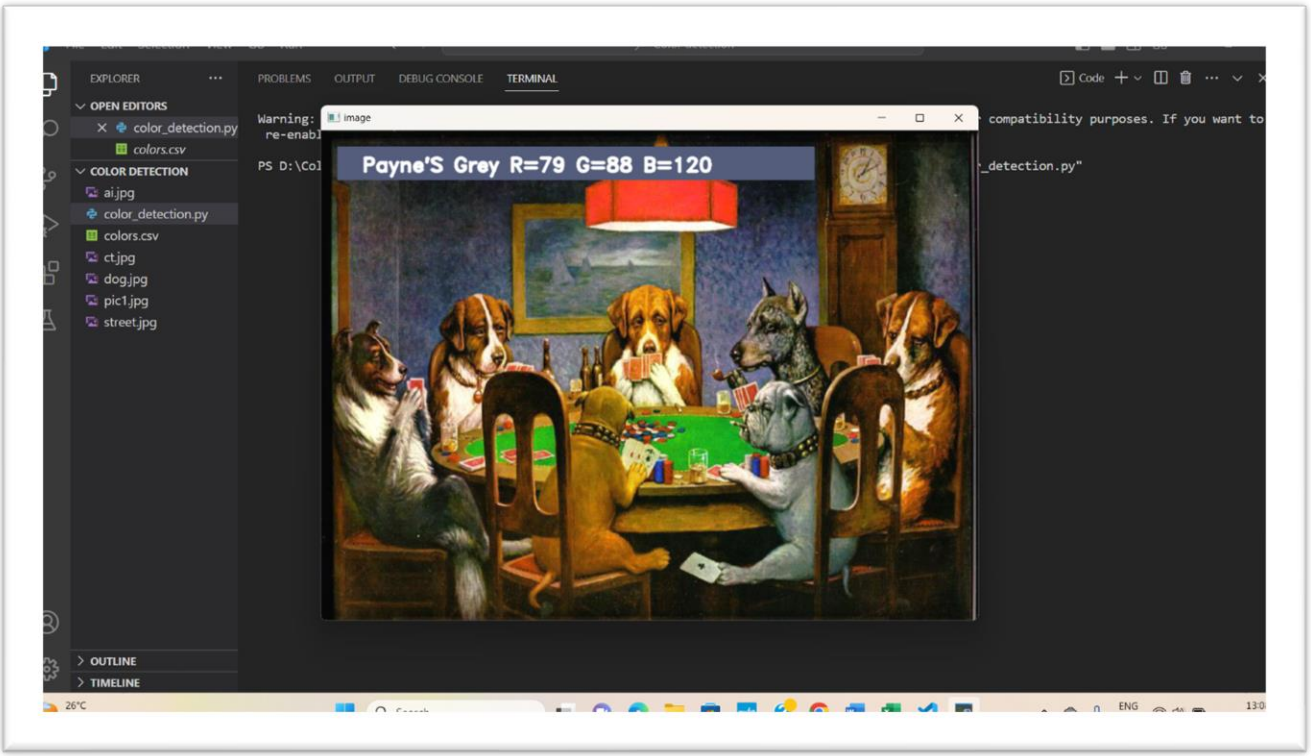
OpenCV (Open Source Computer Vision) library aims at real time Computer Vision. It is mainly used to do all the operations related to images.

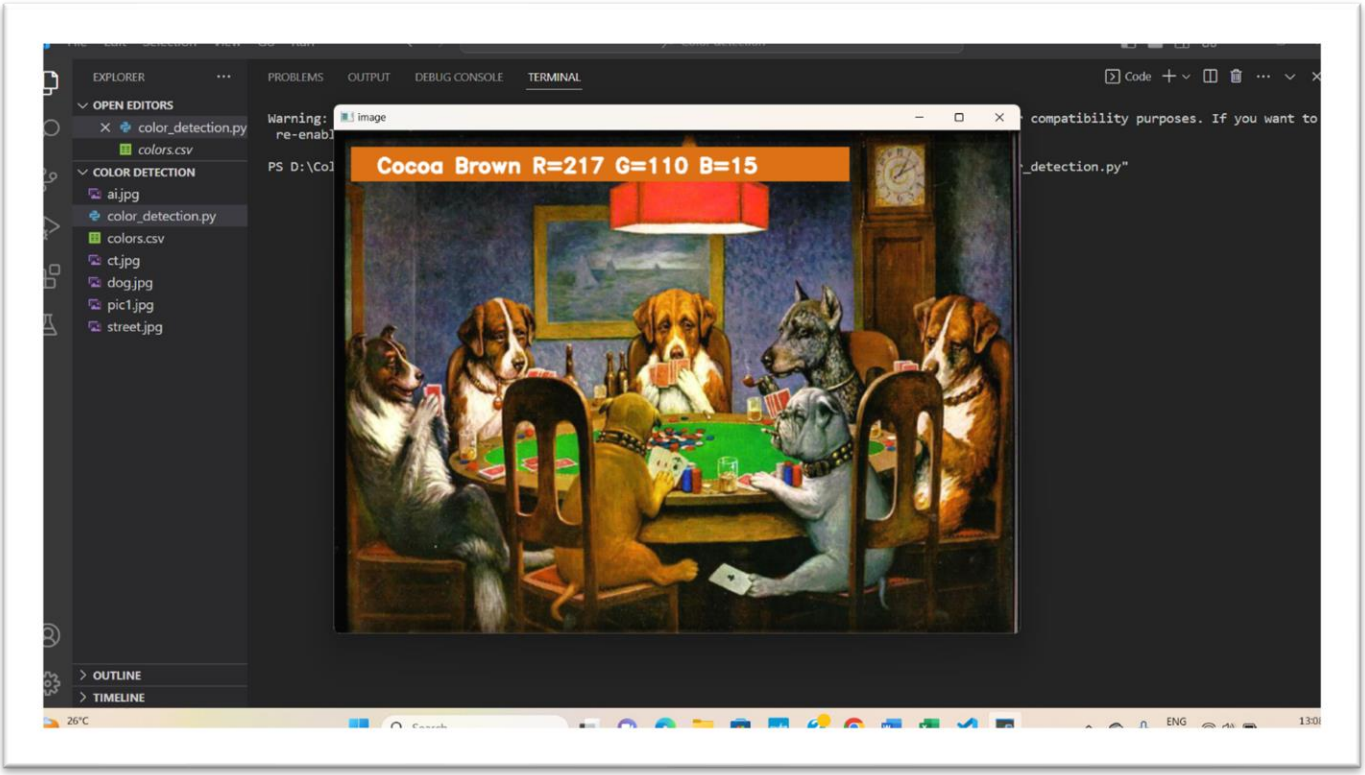
Image Processing OpenCV:

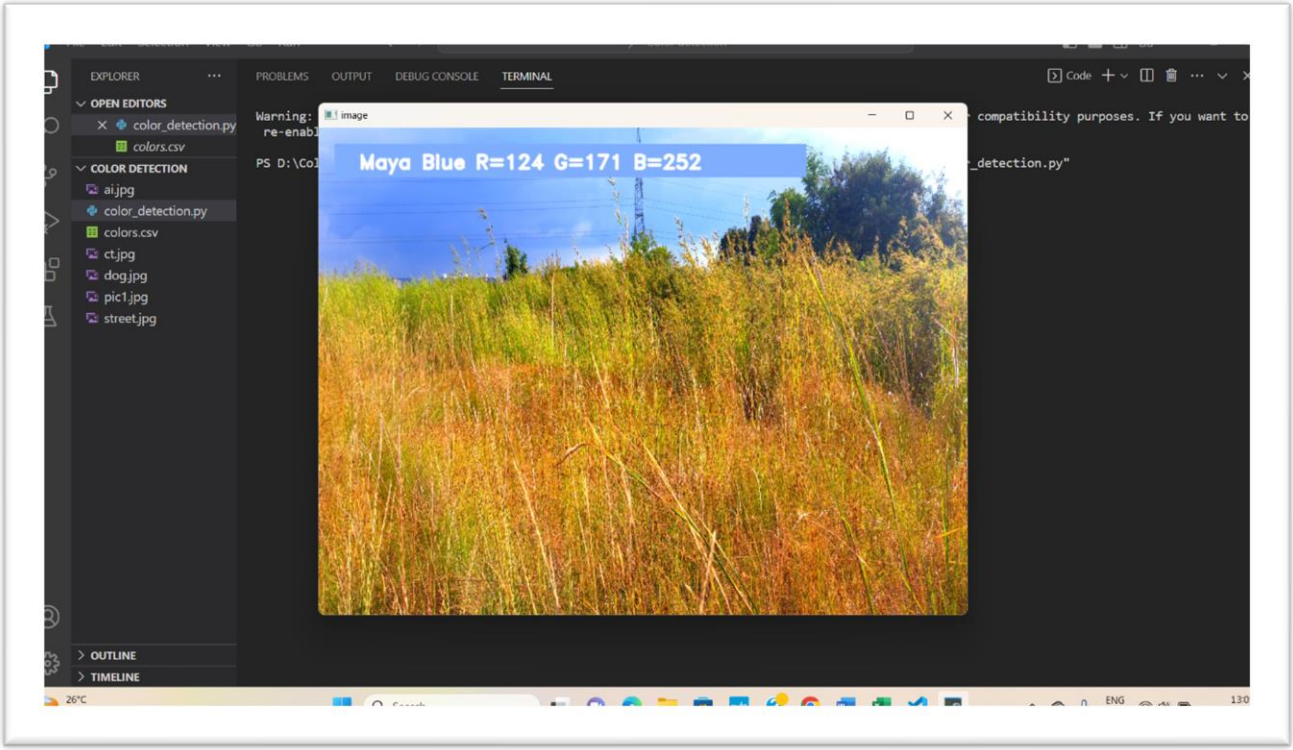
Image Processing technique is used to perform some certain operations on an image, in order to get an enhanced image as an output or to extract some useful information from the image.

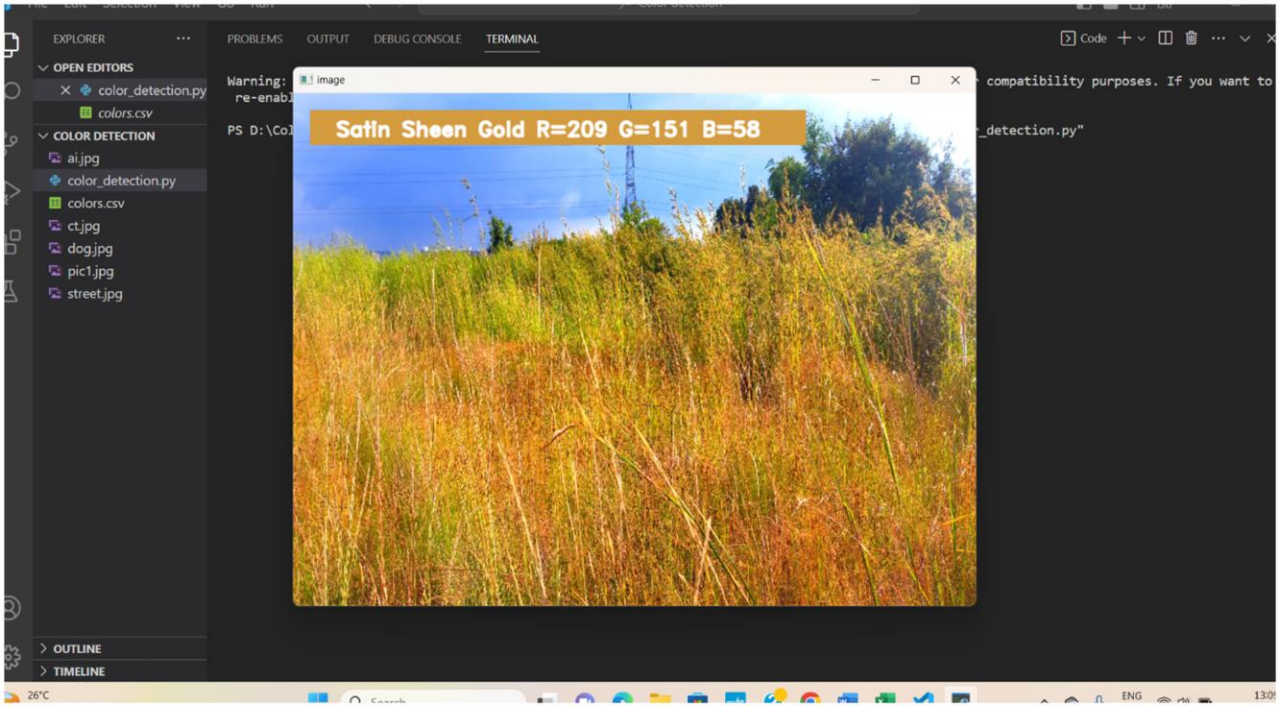
It acts as a type signal processing in which input is an image and output may be an image or characteristics/features associated with that image (we have used it for resizing the image) .

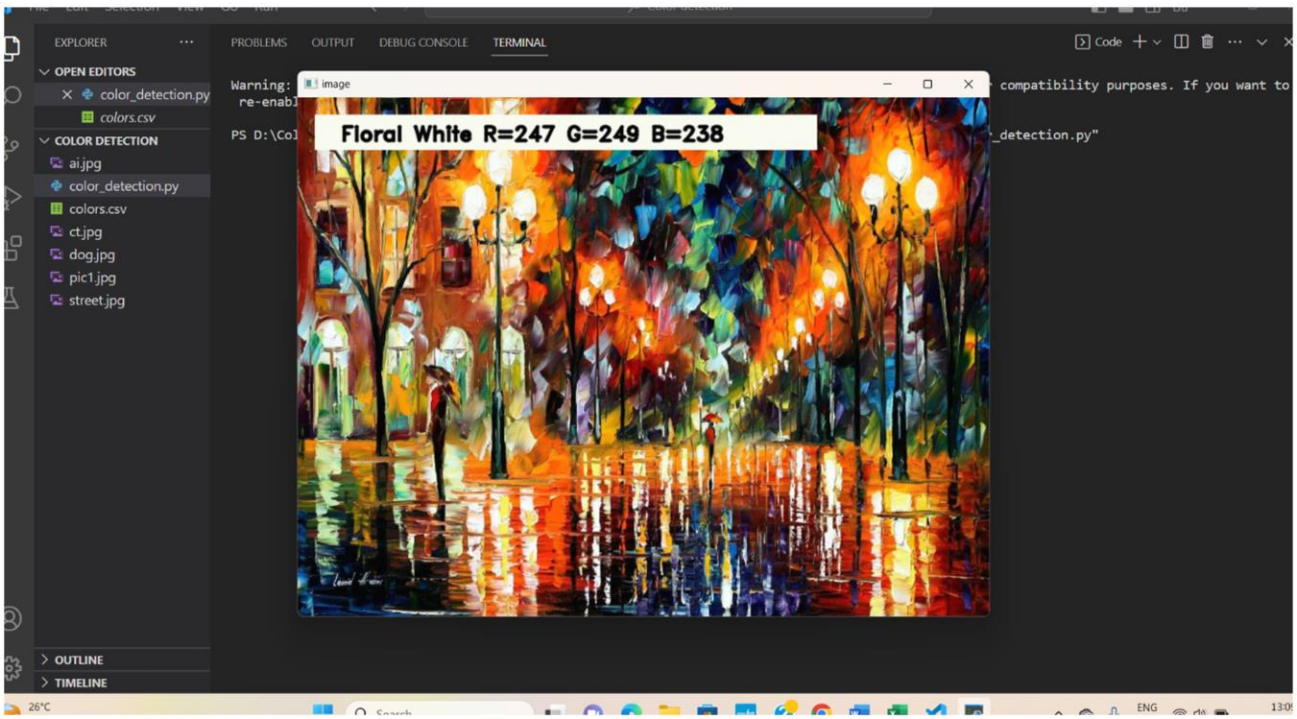
Output

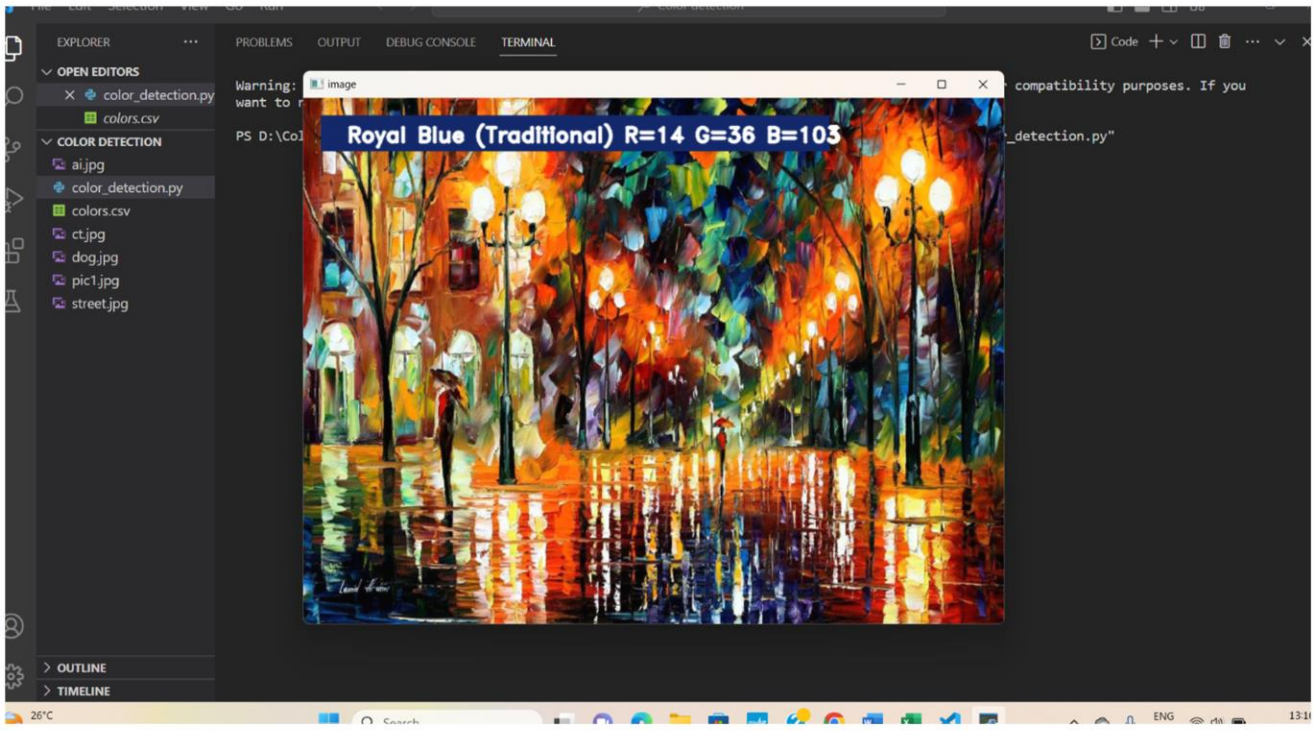

EXPLORER
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
OPEN EDITORS
color_detection.py
colors.csv
COLOR DETECTION
ai.jpg
color_detection.py
colors.csv
ct.jpg
dog.jpg
pic1.jpg
street.jpg
OUTLINE
TIMELINE
image
Payne'S Grey R=79 G=88 B=120


EXPLORER
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
OPEN EDITORS
color_detection.py
colors.csv
COLOR DETECTION
ai.jpg
color_detection.py
colors.csv
ct.jpg
dog.jpg
pic1.jpg
street.jpg
OUTLINE
TIMELINE
image
Cocoa Brown R=217 G=110 B=15


EXPLORER
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
OPEN EDITORS
color_detection.py
colors.csv
COLOR DETECTION
ai.jpg
color_detection.py
colors.csv
ct.jpg
dog.jpg
pic1.jpg
street.jpg
OUTLINE
TIMELINE
image
Maya Blue R=124 G=171 B=252


EXPLORER
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
OPEN EDITORS
color_detection.py
colors.csv
COLOR DETECTION
ai.jpg
color_detection.py
colors.csv
ct.jpg
dog.jpg
pic1.jpg
street.jpg
OUTLINE
TIMELINE
image
Satin Sheen Gold R=209 G=151 B=58


EXPLORER
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
OPEN EDITORS
color_detection.py
colors.csv
COLOR DETECTION
ai.jpg
color_detection.py
colors.csv
ct.jpg
dog.jpg
pic1.jpg
street.jpg
OUTLINE
TIMELINE
image
Floral White R=247 G=249 B=238


EXPLORER
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
OPEN EDITORS
color_detection.py
colors.csv
COLOR DETECTION
ai.jpg
color_detection.py
colors.csv
ct.jpg
dog.jpg
pic1.jpg
street.jpg
OUTLINE
TIMELINE
image
Royal Blue (Traditional) R=14 G=36 B=103

REFERENCE:

Color.csv: https://github.com/codebrainz/color-names/blob/master/output/colors.csv

Code Reference: http://www.youtube.com/@itspyguru