** **

**COLOR DETECTION USING**

**PYTHON AND OPENCV**

**A PROJECT REPORT**

***Submitted by***

**ARUNA.K [REGISTER NO:211419104020]**

**PREETHI.B [REGISTER NO:211419104200]**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

COMPUTER SCIENCE AND ENGINEERING

**PANIMALAR ENGINEERING COLLEGE,**

**CHENNAI-600123.**

**ANNA UNIVERSITY: CHENNAI 600 025**

**MAY 2022**

**BONAFIDE CERTIFICATE**

Certified that this project report **“COLOR DETECTION USING PYTHON AND OPENCV”** is the bonafide work of **“ARUNA.K(211419104020), PREETHI.B(211419104200) ”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr.S.MURUGAVALLI,M.E.,Ph.D., Dr.K.SANGEETHA M.E, PhD.,**

**HEAD OF THE DEPARTMENT SUPERVISOR**

**ASSOCIATE PROFESSOR**

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE,

NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on...........................

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like express our heartfelt and sincere thanks to our Director **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHIKUMAR, M.E., Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Dr.K.SANGEETHA M.E,PhD.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

**NAME OF THE STUDENTS**

**ARUNA.K**

**PREETHI.B**

**ABSTRACT**

The main objective of this python project is the methodology for identifying the shades of colours with an exact prediction of their names with their RGB values. For this, we will have a data file that contains the colour name and its values. we will calculate the distance from each colour and find the shortest one. We will calculate the distance from each colour and find the shortest one. In the existing system they have gone through with the opencv but while extraction of the colors they got the wrong outputs. There is no exact colour representation of colors with accuracy. In the proposed system, we are introducing the CV database and according to it the number of shades that can be identified using 865 colour names along with their RGB and hex values. Whenever the cursor clicks the image, it automatically shows the RGB shades colour values. The system is user-friendly and has simple interface. Can be used in manufacturing company.

***TABLE OF CONTENTS***

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | | **PAGENO** |
|  | **ABSTRACT** |  | |
|  | **LIST OF TABLES** |  | |
| **1** | **INTRODUCTION** |  | |
|  | * 1. OVERVIEW   2. PROBLEM STATEMENT |  | |
|  |  |  | |
| **2** | **LITERATURE SURVEY** |  | |
|  |  |  | |
|  |  |  | |
| **3** | **SYSTEM ANALYSIS** |  | |
|  | 3.1 EXISTING SYSTEM  3.2 PROPOSED SYSTEM  3.3 HARDWARE REQUIREMENTS |  | |
|  | 3.4 SOFTWARE REQUIREMENTS |  | |
| **4** | **SYSTEM DESIGN**  4.1 ER DIAGRAM  4.2 DATA FLOW DIAGRAM  4.3 USECASE DIAGRAM  4.4 CLASS DIAGRAM |  | |
| **5** | **SYSTEM ARCHITECTURE** |  | |
| **6** | **SYSTEM IMPLEMENTATION** |  | |
|  |  |  | |
| **7** | **RESULT AND DISCUSSION** |  | |
| **8** | **CONCLUSION AND FUTURE ENHANCEMENT**  **APPENDICES**  **REFERENCES** |  | |
|  |  |  | |

***CHAPTER-1***

***INTRODUCTION***

**1.1 OVERVIEW**

Colour detection is the process of detecting the name of any colour. Humans perform this action naturally and do not put any effort in detecting colours .Human eyes and brains work together to translate light into colour . Light receptors that are present in our eyes transmit the signal to the brain. Our brain then recognizes the colour. Since childhood, we have mapped certain lights with their colour names. The same strategy is useful in detecting colour names in this project. Three different colours 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 colour name of the pixel along with the composition of three different colours red, blue and green values. It is helpful in recognizing colours and in robotics. One of the applications of colour 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 coloured object by the robotic arm. Colour detection is also used as a tool in various image editing and drawing apps.

**1.2 PROBLEM STATEMENT**

The goal of the project is to display the name of the colour with RGB values.We will use a data file that contains the colour name and its values.We will calculate the distance from each colour and find the shortest one.

Objectives Of The Project

• Improving the Accuracy of the colour prediction

• Improving the Scalability.

• Enhancing the user experience.

***CHAPTER-2***

***LITERATURE SURVEY***

**LITERATURE SURVEY**

**2.1 INTRODUCTION**

The following shows survey did for Colour Detection. The most popular of the existing techniques is been discussed as follows.

**2.2 LITERATURE SURVEY**

**2.2.1 Biologically Inspired QuadTree Color Detection in Dermoscopy Images of Melanoma**

**Author: Sahar Sabbaghi Mahmouei , Mohammad Aldeen,William V.Stoecker & Rahil Garnavi**

**Year: 2019**

This paper presents a QuadTree-based melanoma detection system inspired by dermatologists' color perception. Clinical color assessment in dermoscopy images is challenging because of subtle differences in shades, location-dependent color information, poor color contrast, and wide variation among images of the same class. To overcome these challenges, color enhancement and automatic color identification techniques, based on QuadTree segmentation and modeled after expert color assessments, are developed. The approach presented in this paper is shown to provide an accurate model of expert color assessment. Specifically, the proposed model is shown to: 1) identify significantly more colors in melanomas than in benign skin lesions; 2) identify a higher frequency in melanomas of three colors: blue-gray, black, and pink; and 3) delineate locations of melanoma colors by quintiles, specifically predilection for blue-gray and pink in the periphery and a trend for white and black in the lesion center. The results indicate that the biologically inspired method of automatic color detection proposed in this paper has the potential to play an important role in melanoma diagnosis in the clinic.

**Advantages**

The proposed colour palette mimics human colour interpretation and detected colors and colour locations can be immediately applied in the clinic.

**Disadvantages**

Application of the colour palette and QuadTree colour clustering to larger, publicly available datasets is not available.

**2.2.2** **A Detection Method for Apple Fruits Based on Color and Shape Features**

**Author: Xiaoyang Liu, Dean Zhao, Weikuan Jia,Wei Ji & Yueping Sun**

**Year: 2019**

The skins of most mature apple fruits are incompletely red and also include green and pale yellow color, which increases the difficulty of fruit detection by machine vision. A detection method based on color and shape features is proposed for this kind of apple fruits. Simple linear iterative clustering (SLIC) is adapted to segment images taken in orchards into super-pixel blocks. The color feature extracted from blocks is used to determine candidate regions, which can filter a large proportion of non-fruit blocks and improve detection precision. Next, the histogram of oriented gradient (HOG) is adopted to describe the shape of fruits, which is applied to detect fruits in candidate regions and locate the position of fruits further. The proposed method was tested by images taken under different illuminations. The average values of recall, precision, and F1 reach 89.80%, 95.12%, and 92.38% respectively. The performance of detecting fruits covered at different levels is also tested. The values of the recall are all more than 85%, which indicates that proposed method can detect a great part of covered fruits.

**Advantages**

The proposed method combines the colour feature and shape feature to detect fruits and can improve the accuracy of fruit detection.

**Disadvantages**

Pixel-wise segmentation is more precision than detection boxes.But , the system just detects fruits by rectangle boxes.

**2.2.3.** **Color Edge Detection Using the Normalization Anisotropic Gaussian Kernel and Multichannel Fusion**

**Author:Dongyun Wang, Jiawei Yin, Chu Tang, Xiaojun Cheng,**

**Binzhao Ge**

**Year: 2020**

Color edge detection is a key technique in image processing for vision engineering. In this paper, a new edge detector based on normalized Anisotropic Gaussian Directional Derivative and Multi-channel Gradient Matrix Fusion is proposed. Firstly, the color image is decomposed into six components in the RGB model and the HSV model, respectively. The gradient amplitude of the image edge is emphasized by Contrast Limited Adaptive Histogram Equalization (CLAHE). A normalized Anisotropic Gaussian Derivative is constructed by Multi-direction ANGK to extract the edge strength map of original color image. Finally, Singular Value Decomposition (SVD) was adopted to fuse each channel component in combination with a Multi-channel Morphological Gradient Derivative Matrix to improve the accuracy of edge detection. The proposed detector is compared with three state-of-art edge detectors with the Berkeley dataset (BSDS500) as the database. The results show that the proposed algorithm is more prominent in the performance of noise robustness and edge detection resolution.

**Advantages**

A new edge detector is proposed based on normalizing the Anisotropic Gaussian Kernel and Multi-channel Gradient Derivative Matrix Fusion for detecting the edge of color natural scene images is more efficient in the performance of noise robustness.

**Disadvantages**

It is not focused on extracting the appearance defects in color natural scene images based on the edge detection algorithm proposed in this paper.

**2.2.4 Advanced Color Edge Detection Using Clifford Algebra in Satellite Images**

**Author:** **Uzair Aslam Bhatti, Zhou Ming-Quan, Huo Qing-Song, Sajid Ali, Aamir Hussain,Yan Yuhuan, Zhaoyuan Yu, Linwang Yuan, Saqib Ali Nawaz**

**Year: 2021**

Edge detection is widely used for image processing to improve the detection and classification of objects, segmentation, and extraction of other features. Satellite images are rich in information about objects with different color intensity and have a large amount of noise, so it is difficult to achieve recognition, classification, and feature extraction of small objects through traditional edge detection algorithms.The main contributions of the proposed research are : 1) This paper proposes a new quaternion-based Fourier transform approach for edge detection through the implementation of a mask. 2) A novel approach was applied to color edge detection using Clifford algebra on all types of image. Object detection can be made better through recognition of objects’ geometrical features using advanced Clifford algebra methods. 3) The advanced use of quaternions with detailed implementation in colour.

**Advantages**

This study gives an advanced solution to the problems of edge detection in color images. A new area of research is explored using Clifford algebra and quaternions.

**Disadvantages**

Because high-resolution, remote sensing images have the characteristics of complex scenes and multiple forms of the same target, information accuracy, timeliness etc. are still areas of challenge where progress is needed to obtain more accurate results.

**2.2.5** **A Novel Chicken Meat Quality Evaluation Method Based on Color Card Localization and Color Correction**

**Author:** **Mengbo You, Jiahao Liu, Jian Zhang, Mingdong Xv,**

**Dongjian He**

**Year: 2021**

Among all the chicken meat quality evaluation metrics, color is one of the most significant factors directly related to the freshness of meat inducing the purchase desire. Biochemical tests for evaluating meat quality may contaminate or damage the test samples. Visual rating method is subjectively inefficient and difficult to realize online detection. Colorimeter has the disadvantages of complicated time-consuming operations, high technical requirements and expensive instruments. This paper proposes a low-cost, contactless chicken meat quality evaluation method by examining the color image of chicken meat. Specifically, the meat image is acquired by the camera of a smartphone. To eliminate the chromatic aberration, a pre-defined color card is put beside meat and automatically localized to extract the captured color information for color correction. Finally, the corrected colors of all the experimental meat samples are analyzed by hierarchical clustering to achieve 3 different quality levels.

**Advantages**

This system proposes a low-cost, contactless chicken meat quality evaluation method by examining the color image of chicken meat. To eliminate the chromatic aberration, a pre-defined color card is put beside meat and automatically localized to extract the captured color information for color correction.

**Disadvantages**

Colorimeter has the disadvantages of complicated time-consuming operations, high technical requirements and expensive instruments.

***CHAPTER-3***

***SYSTEM ANALYSIS***

**3.1 EXISTING SYSTEM**

* Color Detection System for Industrial Applications Using Arduino
* In the existing system they have gone through with the opencv but while extraction of the colors they got the wrong outputs.
* There is no exact colour representation of colors with accuracy.

**3.2 PROPOSED SYSTEM**

Our project of Colour detection System can work in any computer with minimum specification. The detection process takes less than a moment and this is very beneficial for companies. The first thing to start with is observation. With the help of Colour Detection, it will be easy to identify the colour and give its name to the persons using it. This project is made to reach each and everyone in the society suffering from colour blindness so that everyone can get benefit from this. Colour Detection technology is very accurate and no one can doubt it. The project designed by us can be used through basic camera also. No special cameras are required for its basic functioning. But for using it as an assistant in self drive cars the camera should be good and more than one should be used for getting an overall view around the car. And all the cameras should be connected to each other.

**3.3 Hardware Requirements**

* A PC with Windows/Linux OS
* Processor with 1.7-2.4gHz speed
* Minimum of 8gb RAM
* 2gb Graphic card 3.2 Software Specification
* Text Editor (VS-code/WebStorm)

**3.4 Software Requirements**

* Text Editor (VS-code/WebStorm)
* Python libraries
* Windows 7 or higher
* Python packages OpenCV, Pandas, and numpy

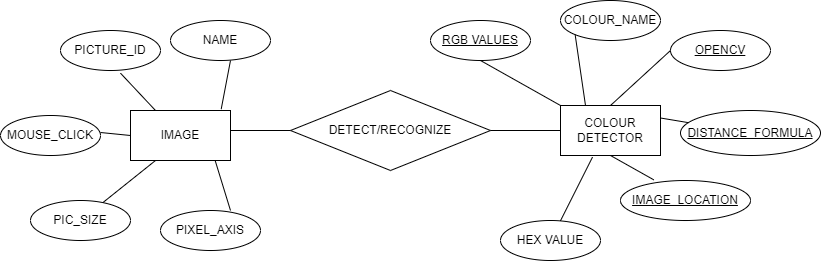
***CHAPTER-4***

***SYSTEM DESIGN***

**SYSTEM DESIGN**

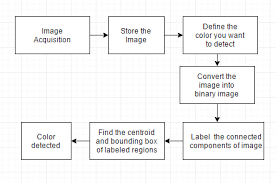
**4.1 ER DIAGRAM**

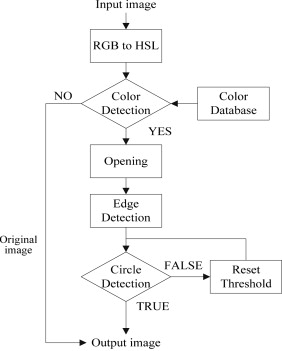
The below figure depicts the entities of the project. The attributes are Image name, picture id, mouse click, pic size, pixel axis, RGB values ,colour name, opencv, distance formula, image location and hex values.

****

**4.2 DATA FLOW DIAGRAM**

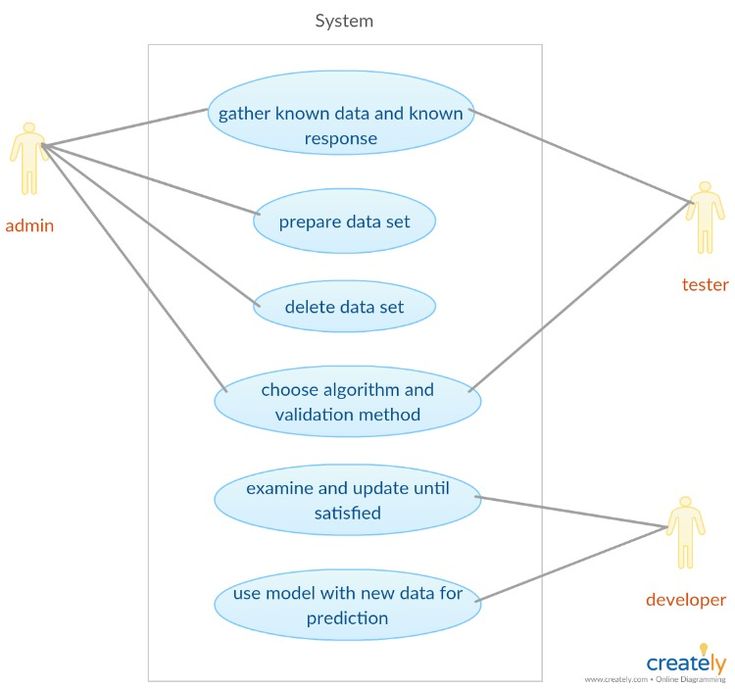
This figure depicts the data flow diagram of the project. The flow starts from image , store the image, then define the colour we want to detect, mouse click event , colour is returned with the name of the colour and RGB values.

****

****

**4.3 USECASE DIAGRAM**

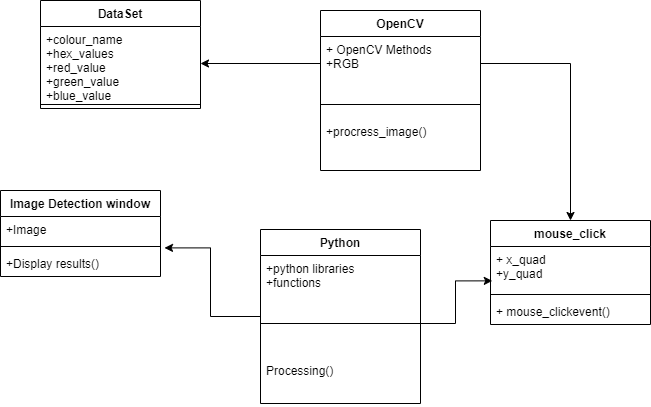
The below figure includes 3 actors (admin, tester and developer) .

****

### **4.4 CLASS DIAGRAM**

The class diagram represents the five main classes. Dataset attributes are colour name, hex values, red values ,green value , blue value and their relationship with openCV containing attributes such as OpenCV methods RGB which is connected to mouse click . The attributes of mouse click are

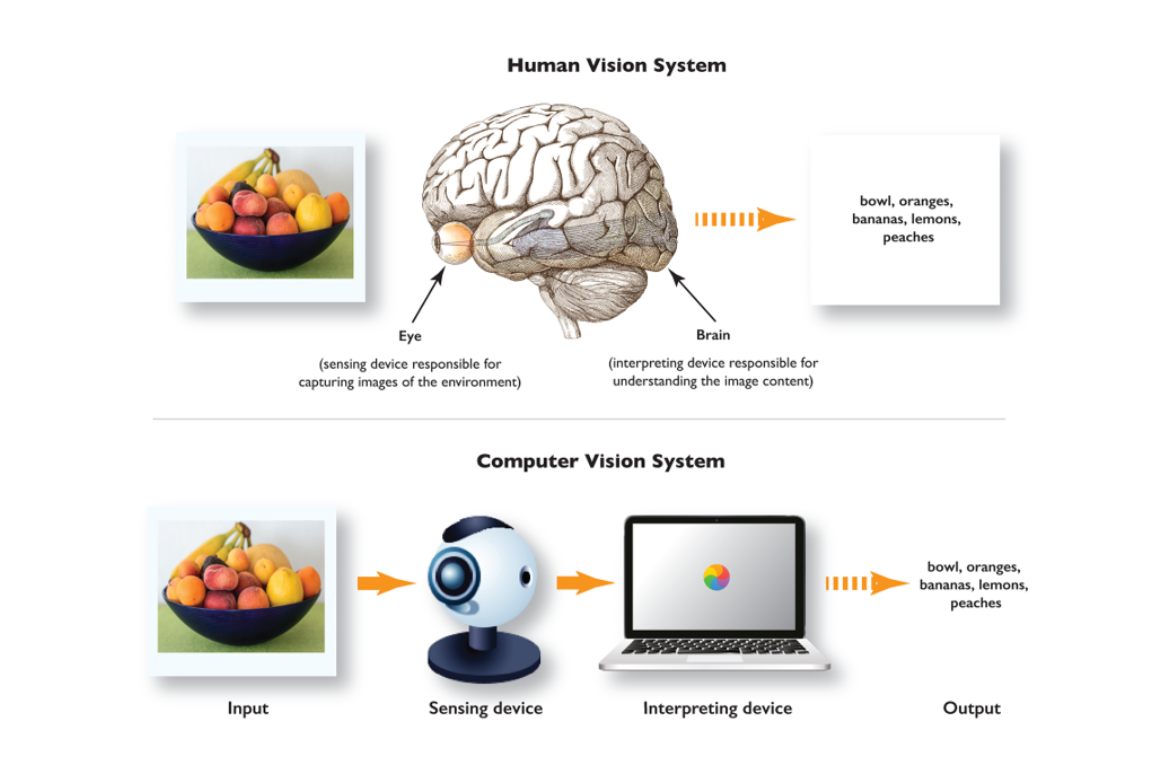
X-coordinate ,Y-coordinate with mouse click event .It is connected to Python class which has python libraries functions and processing. It has a relation with Image detection window that has attributes Image and Display results.



***CHAPTER-5***

***SYSTEM ARCHITECTURE***

**5.1 Architecture Diagram :**



***CHAPTER-6***

***SYSTEM IMPLEMENTATION***

**SYSTEM IMPLEMENTATION**

**Colour\_detection**

import cv2

import numpy as np

import pandas as pd

import argparse

**#Creating argument parser to take image path from command line**

ap = argparse.ArgumentParser()

ap.add\_argument('-i', '--image', required=True, help="Image Path")

args = vars(ap.parse\_args())

img\_path = args['image']

**#Reading the image with opencv**

img = cv2.imread(img\_path)

**#declaring global variables (are used later on)**

clicked = False

r = g = b = xpos = ypos = 0

**#Reading csv file with pandas and giving names to each column**

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

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

**#function to calculate minimum distance from all colors and get the most matching color**

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

**#function to get x,y coordinates of mouse double click**

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)

cv2.namedWindow('image')

cv2.setMouseCallback('image',draw\_function)

while(1):

cv2.imshow("image",img)

if (clicked):

**#cv2.rectangle(image, startpoint, 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 = getColorName(r,g,b) + ' R='+ str(r) + ' G='+ str(g) + ' B='+ str(b)

**#cv2.putText(img,text,start,font(07),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()

***CHAPTER-7***

***RESULT AND DISSCUSION***

***CHAPTER-8***

***CONCLUSION AND FUTURE ENHANCEMENT***

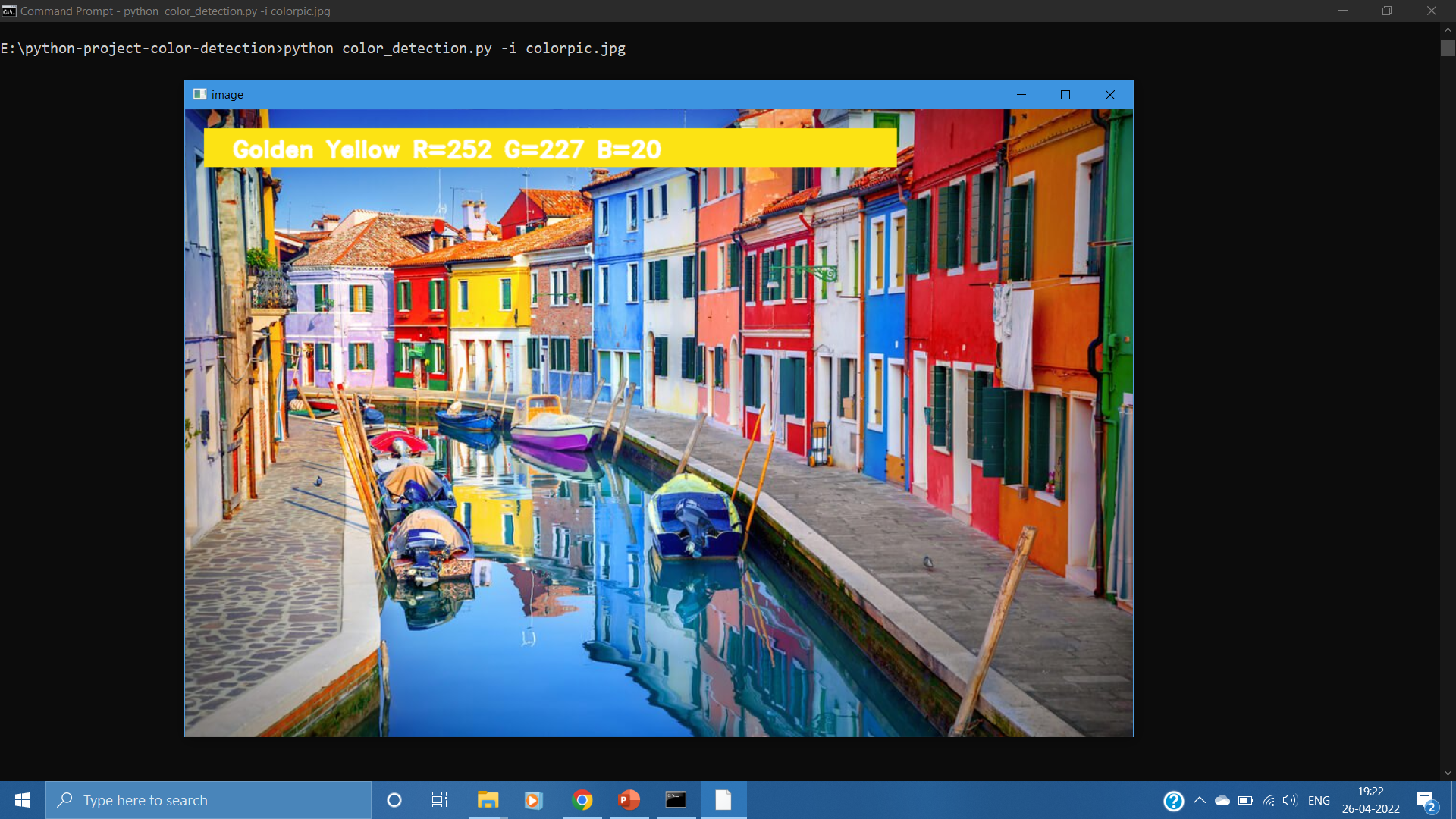
**8.1 CONCLUSION**

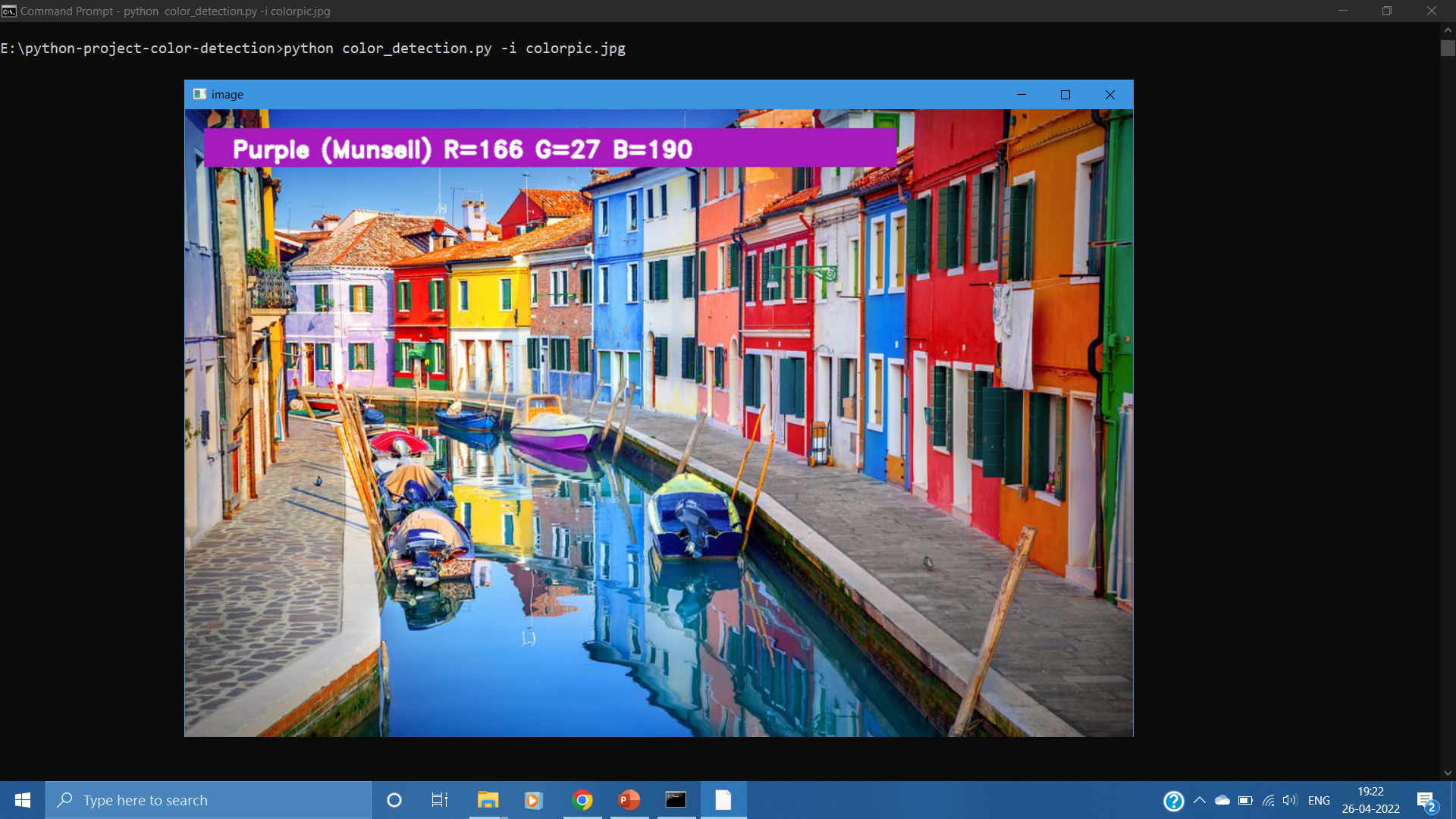
Colour detection technology has come a long way and has a long way to go. When we see selfdrive cars running on roads by themselves following the traffic rules. Today, the machines are ready to for it. Tesla is a frontrunner in this technology. However, next-generation color detection programs will have more upgradations. The apps in smart environments - where computers and equipment are similar to assistant assistants. To achieve this goal computers must be able to reliably identify nearby things and their basic properties like size shape and color(we can’t forget that) in a manner that is naturally consistent within the normal human pattern. They do not require special interactions and should be in line with people's understanding of when recognition goes. This suggests that future intelligent environments should use the same methods as humans, and have the same limitations. These goals are now achievable.

**8.2 FUTURE WORK**

In this paper we defined to get the required color field from an RGB image. In this various steps are implemented using openCv platform. 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.. 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.

***APPENDICES***

******

******

***REFERENCE***

[1]<https://shsu-ir.tdl.org/shsu-ir/bitstream/handle/20.500.11875/1164/0781.pdf?sequence=1>

[2]<https://ieeexplore.ieee.org/document/6208293/>

[3]https://ieeexplore.ieee.org/document/4679917/

[4] <https://pubmed.ncbi.nlm.nih.gov/>

[5]OpenCV– medium.com

[6]Color detection \_–Medium.com

[7]<https://data-flair.training/blogs/project-in-python-colour-detection/>

[8]<https://www.researchgate.net/publication/349355136_Color_Detection_of_RGB_Images_Using_Python_and_OpenCv>