

COLOUR DETECTION PROJECT

INTERNSHIP REPORT

QUARTER IV(REPORT-1)

*submitted by*

Monika Devi -E0320047

Gayathri B - E0120007

Ambati Renuka -E0120053

Kaif Ibrahim - E0120014

Mohammed Umar - E0220009

***In partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**Sri Ramachandra Engineering and Technology**

**Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai -600116**

**JULY, 2020**



*BONAFIDE CERTIFICATE*

Certified that this project report” COLOUR DETECTION” is the Bonafide work of Monika Devi Reg: E0320047 ,Gayathri B Reg: E0120007 ,Ambati Renuka Reg: E0120053 , Kaif Ibrahim Reg: E0120014 , Mohammed Umar Reg: E0220009 who carried out the supervision under

**Signature of Faculty Mentor Signature of Vice-Principal**

|  |  |
| --- | --- |
| **Rajive Gandhi**  Sri Ramachandra Engineering and Technology  Porur  Chennai-600116  **Evaluation Date :**  **Table of Content**  **Title Pages**  1.Introduction 4-5  1.1 Detection 5-5  1.2 Applications 5-5  2. Objective 6-7  3. Workflow 8-8  4.Implementation 9-9  4.1 Software Implementation 9-9  5.Sample 9-11  Conclusion 14-14  References 15-15 | **Prof. M. Prema**  Vice-Principal  Sri Ramachandra Engineering and Technology  Porur  Chennai-600116 |

**ACKNOWLEDGEMENT**

I express my sincere gratitude to our Chancellor, Vice-Chancellor and our sincere gratitude to our Provost **Dr. Raju** and our Vice-Principal **Prof. Prema** for their support and for providing the required facilities for carrying out this study.

I wish to thank my faculty mentor, **Prof. Nandhini JJ,** Department of Computer Science and Engineering, Sri Ramachandra Engineering and Technology for extending help and encouragement throughout the project. Without his/her continuous guidance and persistent help, this project would not have been a success for me.

I am grateful to Department of Computer Science and Engineering, Sri Ramachandra Engineering and Technology, our beloved parents and friends for extending the support, who helped us to overcome obstacles in the study.

1. INRODUCTION – COLOUR DETECTION
   1. Detection of the colour

On a computer, color can be represented in many formats. However, in this tutorial, we will be strictly concerned with only BGR (Blue, Green, Red) and HSV (Hue Saturation Value).With BGR, a pixel is represented by 3 parameters, blue, green, and red. Each parameter usually has a value from 0 – 255. For example, a pure blue pixel on your computer screen would have a B value of 255, a G value of 0, and a R value of 0. Your computer would read this and say, “Ah. This pixel is 255 parts blue, 0 parts green, and 0 parts red.”With HSV, a pixel is also represented by 3 parameters, but it is instead Hue, Saturation and Value.

* 1. Applications
* Color information plays an important role in the color image segmentation and real-time color sensor, which affects the result of video image segmentation and correct real-time temperature value. In this paper, a novel real-time color image segmentation method is proposed, which is based on color similarity in RGB color space. According to the color and luminance information in RGB color space, the dominant color is determined at first, and then color similarity can be calculated with the proposed calculation method of color component, which creates a color-class map. Next, the information of the corresponding color-class map is utilized to classify the pixels.
* In self driving cars , detect the traffic signals.
* Multiple colour detection is used in some industrial robots , to perform pick-and-place task to separate different coloured objects
* This is an implementation of detecting multiple colours in real time using Python program

1. Objective:

The objective of the project is in the color image segmentation, the first step is to choose a color space. The color model we know contains RGB, HSI, HSV, CMYK, CIE, YUV, and so on. RGB model is the most commonly used for hardware color model while the HSI model is the most commonly used color model for color processing. They are often used in image processing technology .RGB space is represented by the three primary colors of red, green, blue; other colors are made up with the three primary colors. The RGB model is represented by the Cartesian coordinate system, as shown in Fig. 1. The three axes stand for R, G, B, respectively, and every point in the three-dimensional space means the three components of brightness value. The brightness value is between one and zero.

Chart

Description automatically generated with medium confidence

In Fig. 1, the origin is black, which value is (0,0,0); while the farthest vertex with a value of (1,1,1) from the origin is white. The straight line between black and white called gray line means that the gray value changes from black to white. The remaining three corners represent the complementary color of the three primary colors - yellow, cyan, magenta.

The three components in the RGB color space, which is highly relevant. And it will be changed accordingly as long as the brightness is changed. RGB is a non-uniform color space, so the perception of differences (color) between the two colors cannot stand fort the distance that between two points in the color space. Thus, the RGB color space is often converted to the other color spaces, such as HSI, HSV, the CIE, and Lab, by using linear or nonlinear transform in image processing. However, the original image we have collected usually is the RGB space, color space conversion will increase the amount of computation. And there are many segmentation methods using RGB color space, for example, license location [16] gets the license plate area accurately by calculating the contrast in the RGB components, reducing the calculated amount.

1. Workflow

Diagram

Description automatically generated

1. Implementation

We started with our PC to store the different versions of our code and update each component of the code as the work is given to us .The implementation of the project has been done by using referential sources of books and small video courses and coding part was developed by using Python

* 1. Software Implementation

We are using argparse library to create an argument parser. We can directly give an image path from the command prompt.The argparse module in Python helps create a program in a command line- environment in a way that appears not only easy to code but also improves interaction.The cv2 module . Open-CV module will help in Image processing like operation on images, videos using objects.Here , we are taking the help of the Open-CV module for Reading an Image. **pd.read\_csv()** reads the CSV file and loads it into the pandas Data Frame.

We have assigned each column with a name for easy accessing.

1. Sample

A picture containing text, orange, colorful

Description automatically generated

A group of colorful buildings next to a body of water

Description automatically generated with low confidence



Code:

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)

#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)

cv2.namedWindow('image')

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

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)

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

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()

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.

References

* <https://data-flair.training/blogs/project-in-python-colour-detection/>
* <https://www.kaggle.com/residentmario/creating-reading-and-writing>
* <https://www.tutorialspoint.com/python_pandas/index.htm>
* <https://www.w3schools.com/python/numpy_intro.asp>
* <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_setup/py_intro/py_intro.html>

\*\*\*