



KLEF
KONERU LAKSHMAIAH EDUCATION FOUNDATION
(Deemed to be university estd, u/s, 3 of the UGC Act, 1956)
(NAAC Accredited "A" Grade University)

PROJECT REPORT

On

**COLOUR DETECTION
SYSTEM**

**Submitted in partial fulfilment of the
Requirements for the award of the Degree of**

Bachelor of Technology

In

Computer science and Engineering

Under the esteemed guidance of

Dr.VIJAYA SRI KOMPALLI

By

180030523 – Maganti Sreelekha

(DST-FIST Sponsored Department)

K L EDUCATION FOUNDATION

Green Fields, Vaddeswaram, Guntur District-522 502

2020-2021

K L EDUCATION FOUNDATION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DST-FIST Sponsored Department)



CERTIFICATE

This is to certify that this project based lab report entitled “**COLOUR DETECTION SYSTEM**“ is a bonafide work done by **180030523-Maganti Sreelekha** in the course **18CS3065S BIG DATA ANALYTICS** in partial fulfillment of the requirements for the award of Degree in Bachelor of Technology in **COMPUTER SCIENCE & ENGINEERING** during the Even Semester of Academic year 2020-2021.

Faculty in Charge
Dr.Vijaya Sri Kompalli

Head of the Department
Hari kiran Vege

K L EDUCATION FOUNDATION
DEPT OF COMPUTER SCIENCE AND ENGINEERING
(DST-FIST Sponsored Department)



DECLARATION

We hereby declare that this project based lab report entitled “**COLOUR DETECTION SYSTEM**” has been prepared by us in the course **18CS3065S BIG DATA ANALYTICS** in partial fulfillment of the requirement for the award of degree bachelor of technology in **COMPUTER SCIENCE & ENGINEERING** during the Even Semester of the academic year 2020-2021. We also declare that this project-based lab report is of our own effort and it has not been submitted to any other university for the award of any degree.

Date:

Place:

ACKNOWLEDGEMENT

Our sincere thanks to **Dr.VIJAYA SRI KOMPALLI** in the Project for her outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **Dr.P.VIDYULLATHA** Course Co-coordinator for **18CS3065 , BIG DATA ANALYTICS** course in the Computer Science and Engineering Department for providing us with adequate planning and support and means by which we can complete this project-based Lab.

We express our gratitude to **Mr. V. HARIKIRAN**, Head of the Department for computer science and Engineering for providing us with adequate facilities, ways and means by which we can complete this project-based Lab.

We would like to place on record the deep sense of gratitude to the honorable Vice Chancellor, K L University for providing the necessary facilities to carry the project-based Lab.

Last but not the least, we thank all Teaching and Non-Teaching Staff of our department and especially our classmates and our friends for their support in the completion of our project-based Lab.

Name of the student
MAGANTI SREELEKHA
180030523

TABLE OF CONTENTS

CHAPTERS	PAGE NO
ABSTRACT	6
CHAPTER 1: INTRODUCTION	6-8
1.1 INTRODUCTION	6
1.2 PROBLEM DEFINITION	7
1.3 SCOPE	7
1.4 PURPOSE	7
1.5 PROBLEM AND EXISTING TECHNOLOGY	7
1.6 PROPOSED SYSTEM	8
CHAPTER 2: REQUIREMENTS & ANALYSIS	8
2.1 PLATFORM REQUIREMENTS	8
2.2 MODULE DESCRIPTION	8
CHAPTER 3: DESIGN & IMPLEMENTATION	
3.1 ALGORITHMS	9
3.2 PSEUDO CODE	10-11
CHAPTER 4: SCREENSHOTS	12-14
CHAPTER 5: CONCLUSION	15

ABSTRACT

Colour detection is the process of detecting the name of any color. For humans this is an extremely easy task but for computers, it is not straightforward. Human eyes and brains work together to translate light into color. Light receptors that are present in our eyes transmit the signal to the brain. Our brain then recognizes the color. Since childhood, we have mapped certain lights with their color names. We will be using the somewhat same strategy to detect color names.

INTRODUCTION

In this colour detection Python project, we are going to build an application by implementing an algorithm through which you can automatically get the name of the colour by clicking on them. So for this, we will have a data file that contains the colour name and its values. The Dataset contains 865 colour names along with their RGB and hex values. Then we will calculate the distance from each colour and find the shortest one.

PROBLEM DEFINITION:

We can't name every color present, so to make this simple we compared with 865 colors available with us with their respective RGB values. We implement an algorithm by using pycharm interface which enables to access command prompt, by which we will be giving the input file which will be of a JPG file. When the code runs then the given file is opened to select the point where you want to find the color.

SCOPE:

Color detection is the basic and important step for proceeding in computer vision. Some special type of spectacles can be made which will make use of computer vision (image processing) along with neural networks to provide an artificial vision to blind people.

PURPOSE:

The main purpose of this project is how to detect a color and recognize objects. Which can be used as a tool in various image editing and drawing apps.

PROBLEM AND EXISTING TECHNOLOGY:

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. But the problem in RGB model is while mixing light of these 3 base colors, you could create any color perception. But such a color set does not exist. RGB does a pretty good job of covering a large part of the color gamut, but not all (RGB fails at saturated cyan and yellow, for example).

PROPOSED SYSTEM:

Color detection serves as the basis for further processing such as sorting of products and quality assessment. The proposed system intends to utilize this ability of the **color** sensor in order to automate further processing tasks such as sorting & arrangement of products and quality assessment.

REQUIREMENTS AND ANALYSIS

Before starting with this Python project with source code, you should be familiar with the computer vision library of Python that is OpenCV and Pandas. OpenCV, Pandas, and numpy are the Python packages that are necessary for this project in Python.

SYSTEM REQUIREMENTS:

A System with min of 4GB RAM , and any python IDE can run this program.

MODULE DESCRIPTION:

To detect the color in the image we write a simple Python code using OpenCV and pandas (python libraries) by implementing an algorithm that is RGB model.

DESIGN AND IMPLEMENTATION

ALGORITHM:

The Project Folder Contains 3 Files:

- **colordetection.py** - main source code of the project.
- **color.jpg** - sample image for experimenting.
- **colors.csv** - a file that contains dataset.

STEP 1: Taking an image from the user

STEP 2: we read the CSV file with pandas

STEP 3: Set a mouse callback event on a window

STEP 4: Create the draw_function

STEP 5: Calculate distance to get color name

STEP 6: Display image on the window

STEP 7: Run the python file

STEP 8: Detect the color

PSEUDO CODE:

```
#180030523
import cv2
import pandas as pd
img_path = 'color.jpg'
csv_path = 'colors.csv'

# reading csv file
index = ['color', 'color_name', 'hex', 'R', 'G', 'B']
df = pd.read_csv(csv_path, names=index, header=None)

# reading image
img = cv2.imread(img_path)
img = cv2.resize(img, (800,600))

#declaring global variables
clicked = False
r = g = b = xpos = ypos = 0

#function to calculate minimum distance from all colors and get the most
matching color
def get_color_name(R,G,B):
    minimum = 1000
    for i in range(len(df)):
        d = abs(R - int(df.loc[i,'R'])) + abs(G -
        int(df.loc[i,'G'])) + abs(B - int(df.loc[i,'B']))
        if d <= minimum:
            minimum = d
            cname = df.loc[i, 'color_name']

    return cname

#function to get x,y coordinates of mouse double click
def draw_function(event, x, y, flags, params):
    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)
```

```

# creating window
cv2.namedWindow('image')
cv2.setMouseCallback('image', draw_function)

while True:
    cv2.imshow('image', img)
    if clicked:
#cv2.rectangle(image, startpoint, endpoint, color, thickness)-1 fills entire
rectangle
    cv2.rectangle(img, (20,20), (600,60), (b,g,r), -1)

#Creating text string to display( Color name and RGB values )
    text = get_color_name(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)

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

cv2.destroyAllWindows()

```

SCREENSHOTS:

colordetection.py - C:\Users\del\l\Desktop\bda pro\colordetection.py (3.9.2)

File Edit Format Run Options Window Help

```
#180030523
import cv2
import pandas as pd
img_path = 'color.jpg'
csv_path = 'colors.csv'

# reading csv file
index = ['color', 'color_name', 'hex', 'R', 'G', 'B']
df = pd.read_csv(csv_path, names=index, header=None)

# reading image
img = cv2.imread(img_path)
img = cv2.resize(img, (800,600))

#declaring global variables
clicked = False
r = g = b = xpos = ypos = 0

#function to calculate minimum distance from all colors and get the most matching color
def get_color_name(R,G,B):
    minimum = 1000
    for i in range(len(df)):
        d = abs(R - int(df.loc[i,'R'])) + abs(G - int(df.loc[i,'G'])) + abs(B - int(df.loc[i,'B']))
        if d <= minimum:
            minimum = d
            cname = df.loc[i, 'color_name']

    return cname

#function to get x,y coordinates of mouse double click
def draw_function(event, x, y, flags, params):
    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)

# creating window
cv2.namedWindow('image')
cv2.setMouseCallback('image', draw_function)

while True:
    cv2.imshow('image', img)
    if clicked:
        #cv2.rectangle(image, startpoint, endpoint, color, thickness)-1 fills entire rectangle
        cv2.rectangle(img, (20,20), (600,60), (b,g,r), -1)

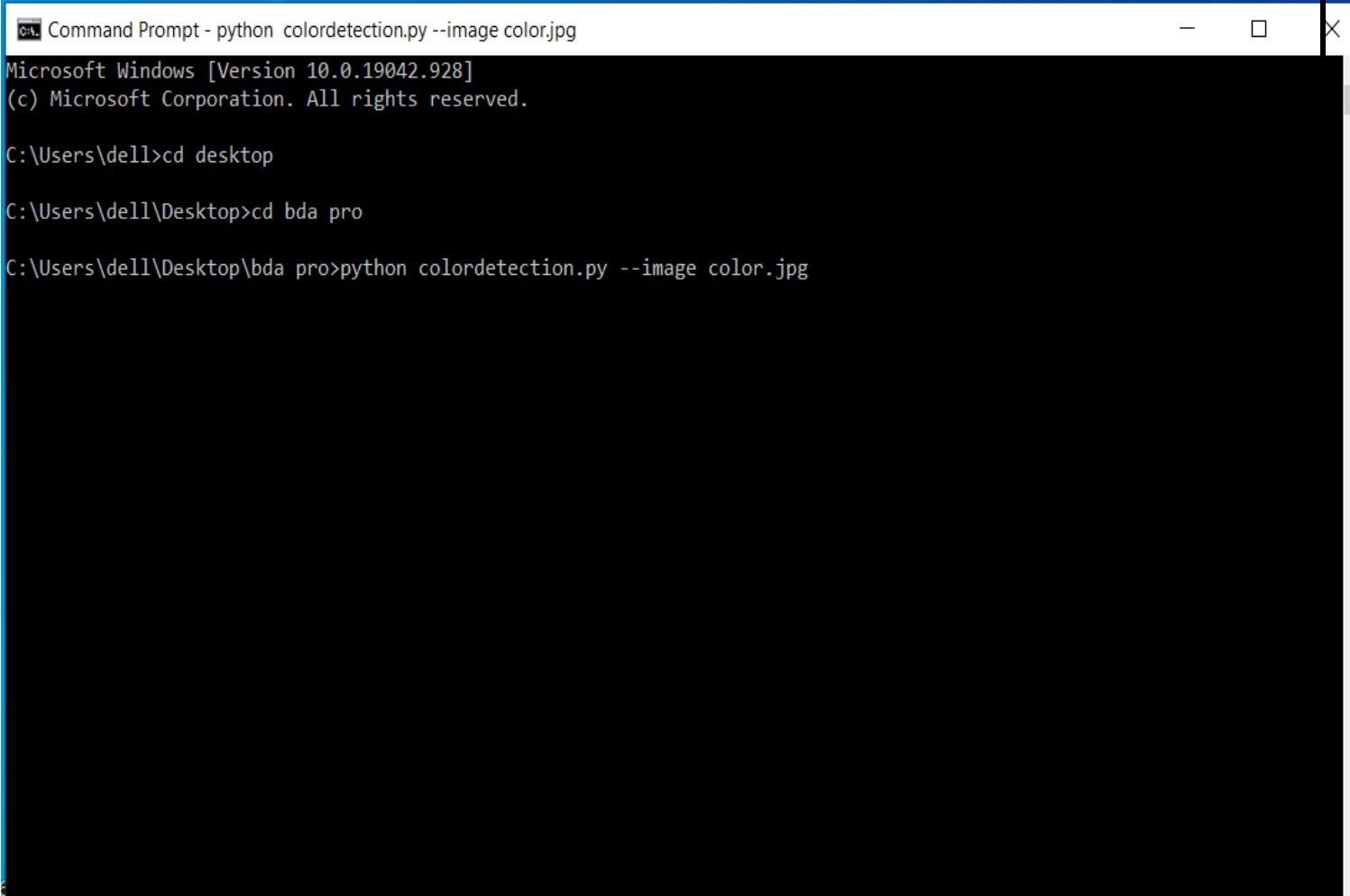
        #Creating text string to display( Color name and RGB values )
        text = get_color_name(r,g,b) + ' R=' + str(r) + ' G=' + str(g) + ' B=' + str(b)
        #cv2.putText(img,text,start,font(0-7),fontScale,color,thickness,lineType )
        cv2.putText(img, text, (50,50), 2,0.8, (255,255,255),2,cv2.LINE_AA)

        #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)

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

cv2.destroyAllWindows()
```

OUTPUTS:



```
Command Prompt - python colordetection.py --image color.jpg
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dell>cd desktop
C:\Users\dell\Desktop>cd bda pro
C:\Users\dell\Desktop\bda pro>python colordetection.py --image color.jpg
```

image

— □ ×

Mandarin duck...



image

— □ ×

Blue (Crayola) R=20 G=95 B=246



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.