Colour Detection

3rd-Year Project

Project work submitted to
Indian Institute of Information Technology Kalyani

Supervisor
Dr. Bhaskar Biswas

Certificate

This is to certify that the project work entitled “Colour Detection” by

Ms. Nikita Barua (Roll no: 39/CSE/17053/251),

Ms. Ritu Chauhan (Roll no: 39/CSE/17070/268) and

Mr. Sanny Kumar (Roll no: 39/CSE/17075/273), undergraduate students in the Department of Computer Science and Engineering, Indian Institute of Information Technology, Kalyani, West Bengal-741235, India, as the 3rd year project work, is an original work carried by them under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of IIIT Kalyani and in my opinion, has reached the standards needed for submission.

(Dr. Bhaskar Biswas, Ph.D.)

Assistant Professor

Department of Computer Science and Engineering

Indian Institute of Information Technology Kalyani

Kalyani, West Bengal, 741235, India.

Nikita Barua	Ritu Chauhan	Sanny Kumar
39/CSE/17053/251	39/CSE/17070/268	39/CSE/17075/273

Acknowledgment

First of all, we would like to take this opportunity to thank our supervisor, Dr. Bhaskar Biswas without whose effort this project would not have been possible. We are grateful to him for working tirelessly after us, answering our doubts whenever and wherever possible. His ideas helped to shape the project in its current form. We are most grateful to the Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, Kalyani, West Bengal-741235, India, for providing us this wonderful opportunity to complete our 3rd-year project. We would like to thank our friends for helping with various topics related to Image Visualisation.

Nikita Barua

39/CSE/17053/251

Ritu Chauhan

39/CSE/17070/268

Sanny Kumar

39/CSE/17075/273

Abstract

This project gives an approach to recognize colors in a two-dimensional image using the color thresh-holding technique in OpenCV and Pandas with the help of an RGB color model to detect a selected color by a user in an image. The methods involved in the detection of color in images are the conversion of three-dimensional RGB image into a grayscale image and then subtracting the two images to get the two-dimensional black and white image, using the median filter to filter out noisy pixels, using connected components labeling to detect connected regions in binary digital images and use of bounding box and its properties for calculating the metrics of each labeled region. Further, the color of the pixels is recognized by analyzing the RGB values for each pixel present in the image. The results of this implementation can be used in security applications like a spy robot, object tracking, segregation of objects based on their colors, intrusion detection.

Contents

1. Introduction 6

2. What is OpenCV? 7

3. Why use OpenCV? 8

4. What is pandas in python? 9

5. Why use pandas? 10

6. Requirements 11

    6.1. Hardware 11

    6.2. Software 11

7. Working 12

8. PseudoCode 14

9. Result 17

10. Advantages & Disadvantages 19

11. Application 20

12. Conclusion 21

13. Future Scope 22

14. References 23

1. Introduction

Colour detection is the process of detecting the name of any color. Simple isn't it? Well, 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. 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.

2. What is OpenCV?

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high-resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

3. Why use OpenCV?

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 a webcam by clicking the webcam button. Then the algorithm analyzes the pattern of the framed part of the webcam. The pattern is matched with the defined color pattern by the RGB color model. If the pattern matches the potential pattern of the RGB color model then the system results with the correct output.

4. What is pandas in python?

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.

5. Why use pandas?

Pandas is very efficient with small data (usually from 100MB up to 1GB) and performance is rarely a concern.

However, if you're in data science or big data, chances are you'll encounter a common problem sooner or later when using Pandas — low performance and long runtime that ultimately result in insufficient memory usage — when you're dealing with large data sets.

Indeed, Pandas has its own limitation when it comes to big data due to its algorithm and local memory constraints. Therefore, big data is typically stored in computing clusters for higher scalability and fault tolerance. And it can often be accessed through a big data ecosystem (AWS EC2, Hadoop, etc.) using Spark and many other tools.

Eventually, one of the ways to use Pandas with large data on local machines (with certain memory constraints) is to reduce memory usage of the data.

6. Requirements

6.1. Hardware

1. I3 processor Based Computer or higher
2. Memory: 1GB RAM
3. Hard Drive: 50 GB
4. Monitor
5. Internet Connection

6.2. Software

1. Windows 7 or higher
2. Python
3. Pandas
4. OpenCV

7. Working

White light is a mixture of three basic colors known as primary colors. They are red, blue and green. These colors have different wavelengths. Combinations of these colors at different proportions create different types of colors. When the white light falls on any surface, some of the wavelengths of the light are absorbed by the surface while some are reflected back based on the properties of the surface material. The Colour of the material is detected when these reflected wavelengths fall on the human eye. To detect the color of material three main types of equipment are required. A light source to illuminate the material surface, a surface whose color has to be detected and the receivers which can measure the reflected wavelengths.
In this color detection Python project, we are going to build an application through which you can automatically get the name of the color by clicking on them. So for this, we will have a data file that contains the color name and
its values. Then we will calculate the distance from each color and find the shortest one.

The Dataset

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. The CSV file for our dataset has been taken from this link:

Colors Dataset

The colors.csv file includes 865 color names along with their RGB and hex values.

The project folder contains 3 files:

- **Color_detection.py** – main source code of our project.
- **Colorpic.jpg** – sample image for experimenting.
- **Colors.csv** – a file that contains our dataset.

The system comprises of 4 major modules as follows:

- **Activated Webcam:**
    - The user opened the webcam by clicking the button on the screen.

- **Scan object part within the camera frame:**
    - Algorithm analysis the pattern of the framed part.

- **Matching framed part:**
    - Pattern matched with a defined color pattern by RGB color model

- **System result:**
    - If the pattern matched with a potential pattern of RGB color model then system output the correct result

8. PseudoCode

1. Taking an image from the user.

2. We are using argparse library to create an argument parser. We can directly give an image path from the command prompt:

```
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)
```

3. Next, we read the CSV file with pandas.
The pandas library is very useful when we need to perform various operations on data files like CSV. pd.read_csv() reads the CSV file and loads it into the pandas DataFrame. We have assigned each column with a name for easy access.

```
index=["color","color_name","hex","R","G","B"]
csv = pd.read_csv('colors.csv', names=index, header=None)
```

4. Set a mouse callback event on a window.
First, we created a window in which the input image will display. Then, we set a callback function which will be called when a mouse event happens.

```
cv2.namedWindow('image')
cv2.setMouseCallback('image',draw_function)
```

With these lines, we named our window as ‘image’ and set a callback function which will call the draw_function() whenever a mouse event occurs.

5. Create draw_function.
It will calculate the RGB values of the pixel which we double click. The function parameters have the event name, (x,y) coordinates of the mouse position, etc. In the function, we check if the event is double-clicked then we calculate and set the r,g,b values along with x,y positions of the mouse.

```
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)
```

6. Calculate distance to get a color name.
We have the r,g, and b values. Now, we need another function which will return us the color name from RGB values. To get the color name, we calculate a distance(d) which tells us how close we are to color and choose the one having the minimum distance.
Our distance is calculated by this formula:
d = abs(Red – ithRedColor) + (Green – ithGreenColor)
+ (Blue – ithBlueColor)

```
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
```

7. Display image on the window.

Whenever a double click event occurs, it will update the color name and RGB values on the window. Using the cv2.imshow() function, we draw the image on the window. When the user double clicks the window, we draw a rectangle and get the color name to draw text on the window using cv2.rectangle and cv2.putText() functions.

9. Result

Input:

Outputs:


Tan R=207 G=168 B=153

10. Advantages & Disadvantages

❖ **Advantages:**

- ➢ The system is user-friendly and has a simple interface.
- ➢ Can be used in a manufacturing company
- ➢ It helps in the sorting of objects based on a three-color approach. It also helps in counting objects.

❖ **Disadvantages:**

- ➢ Data needs to be entered properly otherwise, the outcome may won't be accurate.
- ➢ Problem with the lighting of the scene.
- ➢ Time consumption increases with the dataset.

11. Application

- This system can be used by multiple peoples to get counseling sessions online.
- Some of the applications are the light color temperature measurement, RGB LED consistency control, medical diagnosis systems, health fitness systems, industrial process control, etc…
- Color sensors available in the market are AS73211, TCS3200, TCS3400, TCS34715, TCS34727, colorPAL from parallax, SEN-11195, Lego Mindstorms EV3, etc…

12. Conclusion

With the help of an image processing toolbox in the OpenCV and Pandas, the program has been made which can detect red, blue, green, magenta, yellow, cyan colors. Also, the colored object is being enclosed inside a bounded region along with the centroid of that region.

13. Future Scope

1. Computer vision- Color detection is the basic and important step for proceeding in computer vision. Some special types 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.
2. Spy robots- The spy robots are made to identify objects in the place where they are launched. Object's shape, size, color, orientation is of importance to robots.
3. Object Segregation- An object can be segregated(separated) on the basis of its color.
4. Object Tracking- A moving object can be tracked on the basis of its color.

14. References

[1] Alasdair McAndrew, in “An Introduction to Digital Image Processing with OpenCV and pandas Notes for SCM2511 Image Processing1”, School of Computer Science and Mathematics, Victoria University of Technology.

[2] Digital image processing -Gonzalez woods & Eddins

[3] R. S. Berns, “Principles of Color Technology” (3rd edition New York: Wiley, 2000)

[4] G. Wyszecki and W. S. Styles, “Color Science: Concepts and Methods, Quantitative Data and Formulae” (2nd edition New York: Wiley, 1982)

[5] J. L. Vincent, “Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms”, IEEE Transactions on Image Processing, vol. 2, pp. 176–201, 1993.

[6] Devi, H.K.A., (2006). Thresholding: A Pixel-Level Image Processing Methodology Preprocessing Technique for an OCR System for the Brahmi Script. Ancient Asia. 1, pp.161–165.

[7] The Multi-stage Approach to Grey-Scale Image Thresholding for Specific Applications, Van Solihin and C. G. Leedham

[8] Document Image Analysis by Rangachar Kasturi, Louis Lam, Seong - Whan Lee & Ching Y. Suen.

[9] M. Sezgin and B. Sankur, “Survey over Image Thresholding Techniques and Quantitative Performance Evaluation”, Journal of Electronic Imaging, vol. 13, no. 1, pp. 146–168, Jan. 2004.

[10] N. R. Pal and D. Bhandari, “On Object Background Classification”, International Journal Syst. Science, vol. 23, no. 11, pp. 1903–1920, Nov. 1992.