Color Detection Using Python and OpenCv

Raghav Khanna

Computer Science & Engineering

Lovely Professional University

Jalandhar, India

[raghav.12008119@lpu.in](mailto:raghav.12008119@lpu.in)

ABSTRACT:

The main objective of this application is the methodology for identifying the shades of colors with an exact prediction .A study says, a normal human can able to clearly identify nearly 1 million shades of colors. But in the case of human having could be able to see only 1% (i.e.10,000 colors) from the normal humans. While painting pictures, a painter needs to identify the color patterns exactly or else the reality of image is not clear.

Keywords : RGB value, OpenCv

## INTRODUCTION

Before going into the speculations of the project it is important to know the definition of color detection. It is obvious that humans perform this action naturally and do not put any effort in doing so. While it is not the case for computers.

Human eyes and brain work in co-ordination in order to translate light into color. Light receptors that are present in eyes transmit the signal to the brain which in turn recognizes the color. There is no exaggeration in saying that humans have mapped certain lights with their color names since childhood. The same strategy is useful in detecting color in this project.

Three different colors Red, Green and Blue are being tracked by utilizing the fundamentals of computer vision. After successful compilation when I execute

the code a window redirects to the image displayed on it whose path is given as an argument.

Additionally, I obtain the color of the pixel along with the composition of three different colors red, blue and green values. It is helpful in recognizing colors and in robotics. One of the applications of color detection by computer vision is in driver less cars. This system is useful in detecting traffic and vehicle backlights and takes decisions to stop, start and continue driving. This also have much application in industry to pick and place different colored object by the robotic arm. Color detection is also used as a tool in various image editing and drawing apps.

## METHODS AND MATERIAL

Image Capture:

The first step is to fetch a high-quality image with resolution. Our project begins by capturing real-time video frames from the camera. I use the OpenCV library to access the video feed from the camera, allowing us to work with each frame in real-time.cap = cv2.VideoCapture(0)

Extraction of RGB Colors:

In this phase, the 3 layered colors are extracted from the input image. All the color images on screens such as televisions, computer, monitors, laptops and mobile screens are produced by the combination of Red, Green and Blue light.

Each primary color takes an intensive value 0 (lowest) to 255 (highest). When mixing 3 primary colors at different intensity levels a variety of colors are produced. For Example: If the intensity value of the primary colors is 0, this linear combination corresponds to black. If the intensity value of the primary colors is 1, this linear combination corresponds to white.

get\_limits Function:

The get\_limits function plays a crucial role in our color detection process. It computes the lower and upper limits of the HSV (Hue, Saturation, Value) color space for a specified color. Here's how it works: I start by converting the provided color into an HSV color representation. This conversion is important because it allows us to represent colors in a way that's more suitable for color detection tasks. Next, I determine the hue of the color. Hue is a key component of the HSV color space and represents the actual color itself. Based on the hue value, the get\_limits function defines a range of acceptable hue values for the color, considering the wrap-around effect for colors like red that span the 0-180 hue range. The function then constructs the lower and upper HSV limits for the specified color, which can be used to create a color mask for that color in the frame.Color

Mask Creation:

The color mask is generated using the calculated lower and upper HSV limits obtained from the get\_limits function. This mask isolates the pixels in the frame that match the specified color within the defined HSV range. This step is crucial for identifying the regions of interest in the video frames that correspond to the specified color.

Conversion to HSV Color Space:

Before proceeding with color detection, we transform the RGB color representation into the Hue, Saturation, and Value (HSV) color space. This conversion is crucial because it offers advantages for color-based object detection:Hue (H): Represents the actual color. Hue values range from 0 to 180 (0-360 degrees), covering the entire color spectrum.Saturation (S): Defines the vividness or purity of the color. A saturation value of 0 corresponds to grayscale, while higher values represent more vibrant colors.Value (V): Reflects the brightness of the color. A value of 0 indicates black, while higher values indicate brighter colors.This conversion to HSV allows us to describe and detect colors more effectively in various lighting conditions and across a wide range of colors.

Image Display with Shades of Color:

The rectangle window is used to display the image with shades of color. After the double-click is triggered, the RGB values and color name is updated.

To display an image Cv2.imshow () method is used. By using cv2.rectangle and cv2.putText () functions, the color name and its intensity level can be obtained.

text=getColorName(r,g,b) + 'R='+str(r) + 'G='+str(g) + 'B=' +str(b).

## SYSTEM ARCHITECTURE

**A diagram of a flowchart

Description automatically generatedFigure 1 :** Architecture Diagram

The above architecture shows the capability for the project. It consists of a well defined sequence diagram that is abstracted from the source code. It leverages the rich capabilities of the technology such as OpenCv library in python.

The above architecture makes the process more efficient based on principles and properties related to each other. As we know that Red, Green and Blue are the primary colors that can be mixed to produce different colors. The present color detection project takes the path of an image as an input and looks for the composition of three different colors red, green and blue in the given image.

## EXPERIMENTAL RESULTS



(a)



(b)



(c)

**Figure 2:** (a) Original input image of nature (b) Output image with Color intensity RGB values as R=49 G=52 B=21 for Olive Drop (c) Output image with Color intensity RGB values as R=18 G=88 B=186 for Sapphire



(a)



(b)

(c)

**Figure 3:** (a) Original input image of Salt Lake (b) Output image with Color intensity RGB values as R=3 G=9 B=97 for Royal Blue (c) Output image with Color intensity RGB values as R=252 G=229 B=13 for Golden Yellow

## CONCLUSION

In this paper I defined to get the required color field from an RGB image. In this various steps are implemented using openCv platform. The main positive point of this method is its color differentiation of a mono color.

In the future scope, the detection of the colour detection techniques has different other applications like facial detection, color conversion for grey scale image etc. that can also be implemented.

## FUTURE WORK

In existing system there is no exact color representation of colors with accuracy. In proposed system, I 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.

## VI. REFERENCES

[1]. Weiming Hu, Xue Zhou, “Active Counter-Based Visual Tracking by Integrating Colors, Shapes and Motions”, IEEE TRANSACTIONS ON

IMAGE PROCESSING, VOL. 22, NO. 5, MAY 2013

[2]. G.M. Snoek, “Evaluating Color Descriptors for Object and Scene Recognition”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 32,

NO. 9, SEPTEMBER 2010

[3]. Claudia Nieuwenhuis, “Spatially Varying Color Distributions for Interactive Multi Label Segmentation”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL 35, NO. 5, MAY 2013

[4]. Kok -Meng Lee, “Effects of Classification Methods on Color-Based Feature Detection with Food Processing Applications”, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, VOL. 4, NO. 1,

JANUARY 2007

Project repo: <https://github.com/coder2200/Computer-Vision>