

# Color Identification for Visually Impaired people

Hanuman Vamshi Krishna Amanaganti<sup>1</sup>, Sri Charan Koride<sup>2</sup>, Teja K<sup>3</sup>, Subhash Chandra N<sup>4</sup>

<sup>1</sup> Student, CVR College of Engineering, Ranga Reddy, India

<sup>2</sup> Student, CVR College of Engineering, Ranga Reddy, India

<sup>3</sup> Student, CVR College of Engineering, Ranga Reddy, India

<sup>4</sup> Professor, CVR College of Engineering, Ranga Reddy, India

## Abstract

Finding objects with complex patterns and color is a challenging task for visually impaired people. This project will help them to assist in this task and works in an effort to encourage their individuality without any human interaction. The use of an efficient computer-based system to match multiple colors helps visually impaired people and blind people to distinguish between pattern and their color information. There are three main components in our methodology for detecting Color and their Pattern detection, The proposed computer-based system that detect its color based on RGB scheme.

**Keywords:** *Color Detection, Complex Patterns, Human Interaction, Visually Impaired, RGB Scheme.*

## 1. Introduction

Visual impairment is vision loss to such a degree as to qualify as a significant limitation of visual capabilities resulting from disease, trauma, congenital or degenerative conditions that cannot be corrected by conventional means such as refractive correction or medications. Nowadays, people with visual disability are about 303 million globally, whereby 39 million of them are completely blind and the remaining 264 million are having vision limitations or known as partial blindness. About 90% of them are living in developing countries. [1] As blindness can be defined as a lack of vision, it also can be referred to a loss of vision that cannot be corrected with glasses. Color is meaningless to people who are born blind as they never see color and never understand color. However, for some people who become blind later in life or have partial blindness, color is also important in helping them recognize things. Mostly, visually disabled people face difficulties in recognizing colors in their lives. For instance, to identify color in pedestrian crossing lights, clothes, paints, foods, and etc.

This motivated us to work on the project in order to help visually impaired people detect color without depending on others thereby enabling them to retain their dignity and perform their tasks with ease. Therefore, this application should be simple, user-friendly, practical and affordable to those with visual disability. The objective of this study is achieved by, firstly studying the behavior of disabled people especially for color reorganization

## 2. Proposed Work

In this section we'd like to explain the technologies we've used for development and the dataset used for development and the working of the application.

### 2.1 Dataset Description

The Dataset used for this application is colors.csv.

- 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, we can define colors in approximately 16.5 million ways.

- In the dataset, the rgb values of the color is mapped to their color name. Not all the combinations are mapped, but the ones that can be distinguished by human eye are mapped.
- The dataset contains 864 rows with 6 columns.

## 2.2 Features and Technologies used

We can broadly say that the application facilitates 2 features in main i.e., color detection and the color search. Besides this it also provides the output of the color detected at a point in audio format and the application can use up to 2 cameras if the device has any.

The main technologies used are OpenCV, pandas, numpy, Kivy and KivyMD.

## 2.3 Technology Description

As discussed, the main technologies used are OpenCV, pandas, numpy, Kivy and KivyMD.

OpenCV:

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection [2]. It is used to read the camera feed and display to the user.

Pandas:

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series [7]. As known, it is used to manipulate the dataset.

Numpy:

In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important [8]. In this particular application, this package is used to define the HSV ranges for color search.

Kivy:

Kivy is a graphical user interface opensource Python library that allows you to develop multi-platform applications on Windows, macOS, Android, iOS, Linux, and Raspberry-Pi. In addition to the regular mouse and keyboard inputs, it also supports multitouch events. The applications made using Kivy will similar across all the platforms but it also means that the applications fell or look will differ from any native application. KivyMD is the Material Design of Kivy [5][6].

As the application aims to facilitate 2 different features that can't be provided in a single frame with OpenCV, a framework is to be used for development. There are a no of frameworks that could be used to develop the features in a single screen. But each one of them have their own limitations, some were unstable with OpenCV, some have limited access. Of all the options, Kivy was the best one to choose and was a cross-platform GUI framework. Kivy, along with KivyMD, was used to develop the main user interface of the application.

## 2.4 Features Description

As mentioned, the app expedites 2 main features i.e., color detection and color search. Upon these features it also has a audio output feature, and the app can access 2 main cameras of the device, in which the app is being run on.

#### Color Detection:

The main aim of this feature is to detect the color at a point clicked by the user on the live video being rendered and shown in the Image frame. To include a live video as part of kivy is not an easy task. Firstly, the layout defined should be added with Image widget and the live video should be read by OpenCV. Now that we've the live video read and the target widget to display the video loaded. To connect the source and the target, the frame is to be converted to texture and the image texture has to be populated with it. As we can't load live video in the app, we create a simulation by loading the video by either 30fps or 60fps and this is executed infinitely.

Now that the video is loaded, the frame has to be enabled with the touch input property so that the user can click or touch a point to know the color at that point. For this `on_touch_down` method, defined in Widget class of kivy should be overridden. This method helps us in getting the coordinate clicked. With the help of coordinate, we can get the rgb score with the help of OpenCV package. From the rgb score, we can get the color from the colors dataset by taking the difference of the rgb scores and getting the closest color. This color is displayed in the Label of the kivy, and is displayed to the user. Besides displaying to the user, the color name is also provided as voice output.

There is an interesting button placed at the top of the video frame, for camera switching. Based on the state of the button, the respective video is loaded and the rest is same. Upon clicking, the color name is displayed on the screen and is also as audio.

Initially when the video is loaded for the first time, the color at the center of the frame is shown as the output as most of the users have more concentration towards the center of the frame.

#### Color Search:

The main aim of this feature is to search the entire video frame for a color and highlight the region which has the color given as input. This also follows the same procedure as the color detection to load the video and display to user.

For searching a color, the user has to populate the text field with the color name and click search icon adjacent to the text field. Upon clicking the icon, the color name populated in the textbox is passed to the main class which has the Image frame loaded to search for the color. If the input color is not empty and is present in the color list, containing the HSV ranges, then the image frame loaded is converted from BGR to HSV and the mask of the frame is calculated with the help of lower and upper HSV values predefined in the colors list.

After finding the mask of the color, the contours for the mask are retrieved and with the help of OpenCV, the contours are displayed in the frame. If the color is not in the colors list or the color value is null, either the contours are not drawn or the existing contours are erased.

Similar to the former feature, the camera switch button works as smoothly to this feature also. Upon clicking the button, the respective camera feed is loaded and by giving the input color name, you can get the regions where the input color is evident.

In both of these features, initially the primary camera feed is rendered and later according to the button click event.

### 3. System Design

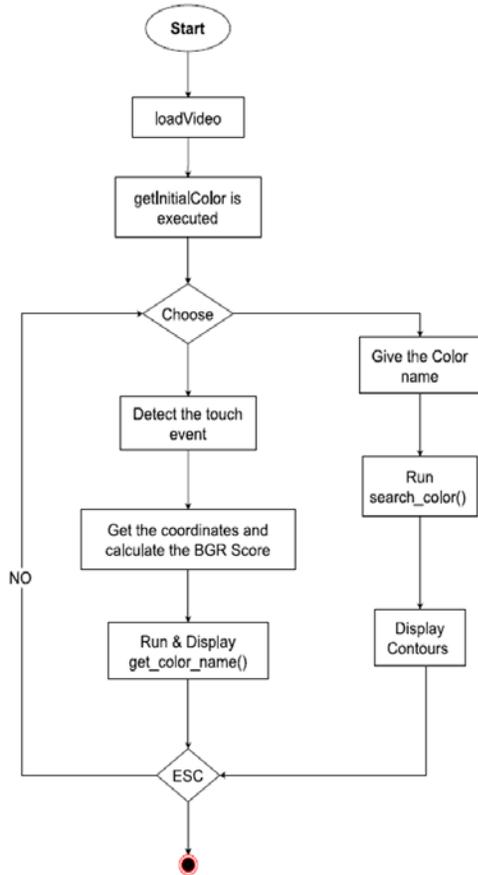


Fig 1 Activity Diagram

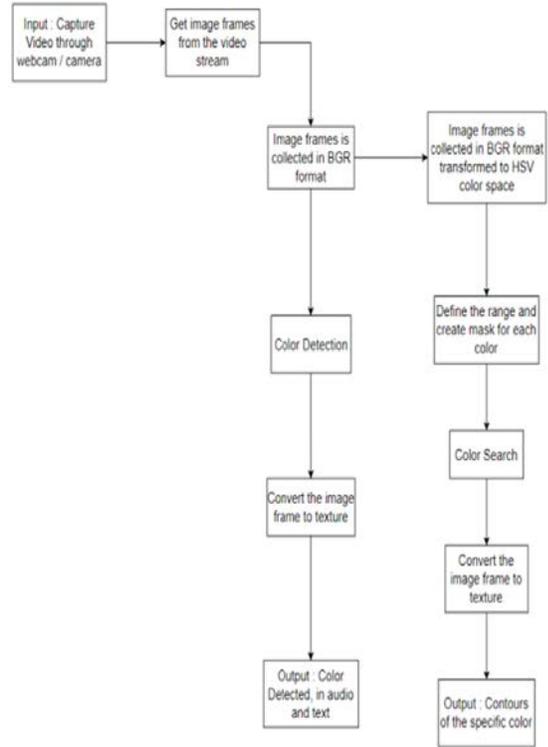


Fig 2 System Architecture

#### 4. Results

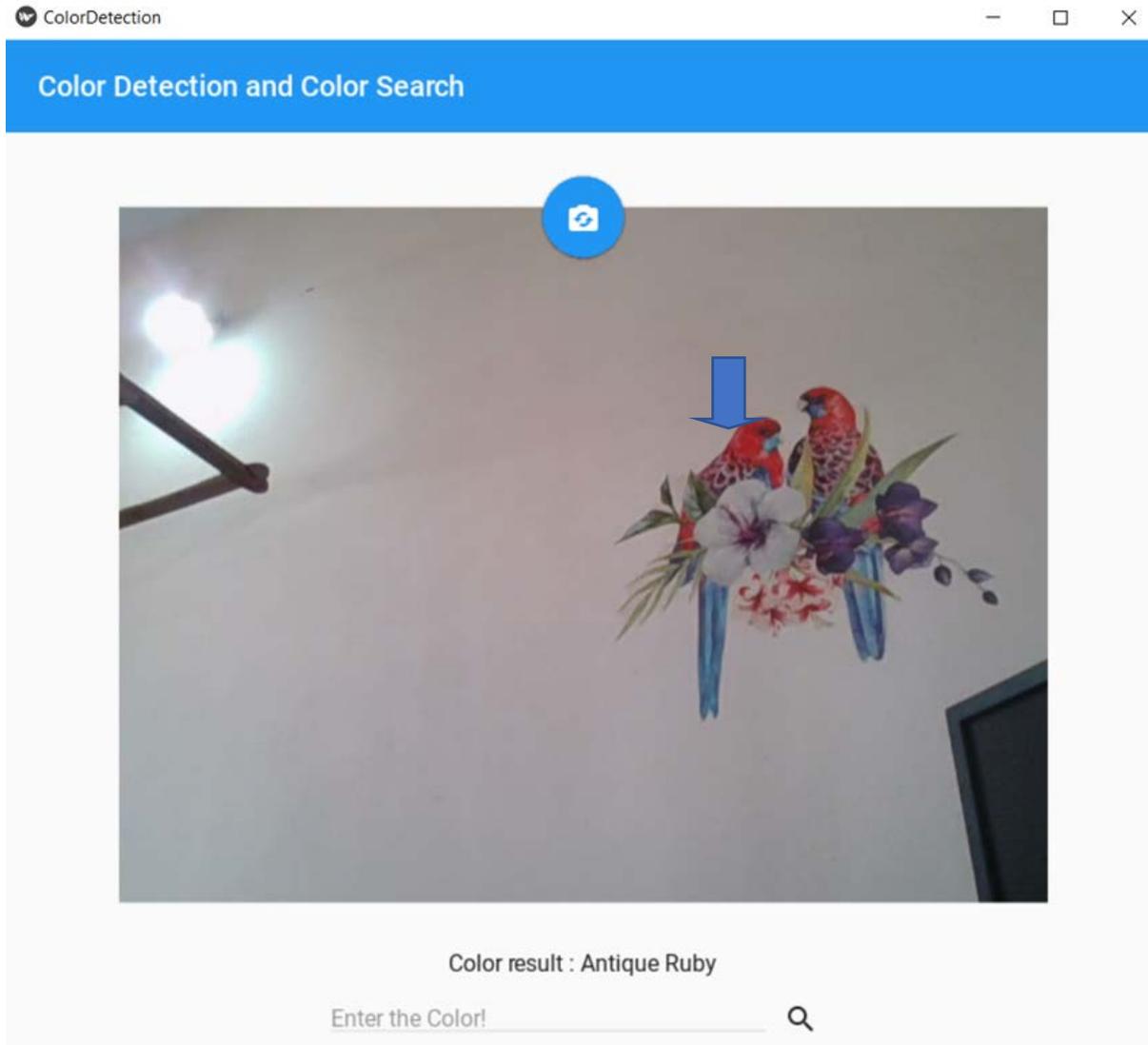


Fig 3 Result for the color detection at a selected point.

The color of the pixel on the parrot's neck is displayed as "antique ruby" in this output (mouse pointer not visible, pointed to by the blue arrow).

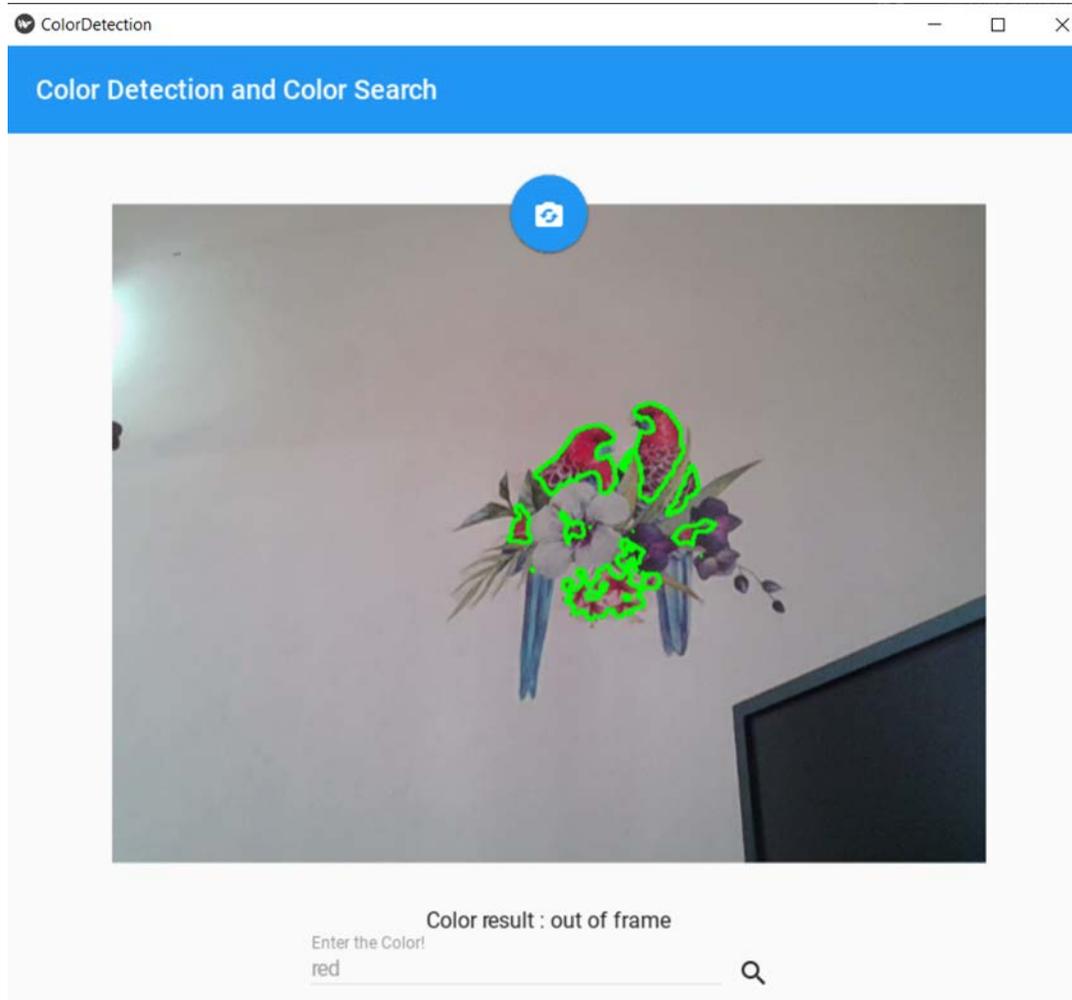


Fig 4 Result for drawing contours around a given color.

All the regions having red color are highlighted by drawing contours around them

## 5. Conclusions

We can draw the following conclusions from this paper. This application can accurately detect color of a point pointed to by the user using a mouse. The application can read out the detected color to the user i.e., the app also helps the user with voice-based output. This application can detect any region with the color that the user desires to see in the frame i.e., if user gives color name as an input, the application draws contours around the color part of the frame. The contours are drawn using the “Topological Structural Analysis of Digitized Binary Images by Border Following” algorithm. This application can use secondary camera if the device has any. We’ve also concluded that OpenCV is not stable with java-based application development platforms like Android Studio, Bizness Apps..., as OpenCV is built with python, the success rate of the app deafens. Some of the positive points of the application are a wide range of color detection that can be differentiated by a naked human eye and the speed of the application. It almost responds to click and search instantly giving a user a better user experience

## **6. Future Scope**

The application can be deployed in multiple platforms like android, iOS etc. It can be used in automated vehicles to detect signal colors. Computer vision- 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. 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 robot. Object Segregation- An object can be segregated (separated) on the basis of its color. Object Tracking- A moving object can be tracked on the basis of its color.

## **References**

- [1] Visual impairment and blindness. (n.d.). Retrieved May, 15th, 2018, <http://www.who.int/mediacentre/factsheets/fs282/en/>
- [2] [https://www.tutorialspoint.com/opencv/opencv\\_overview.htm](https://www.tutorialspoint.com/opencv/opencv_overview.htm)
- [3] <https://buildozer.readthedocs.io/en/latest/>
- [4] <https://pypi.org/project/pyttsx3/>
- [5] <https://kivy.org/doc/stable/>
- [6] <https://kivymd.readthedocs.io/en/latest/>
- [7] <https://pandas.pydata.org/>
- [8] <https://numpy.org/>