ABSTRACT

The foremost stage in many of the image processing applications is Color detection. It is significantly used in applications such as self-driving cars, object detection, traffic signal detection, skin tone detection and object tracing. While tracing an object in motion, color is constant than any other attributes. This paper gives an approach to detect the label of the color by placing the cursor and double clicking at that position of the image and tracks the red, green and blue color objects using bounding box property. By examining the RGB values of every pixel in the image, the color of the pixels is recognized. All the objects of interest in the video are detected and tracked by a rectangular bounding box using HSV color model. The results of this implementation can be used in self driving cars to detect traffic signal, in some industrial robots to perform pick-and-place task in separating colored objects and as a tool in various drawing and image editing applications.

In this project, We build an application through which we can automatically get the name of the color by double clicking on it. We have a data file that contains the color name and its values. Then we calculate the distance from each color and find the shortest one. We extract the color RGB values and the color name of a pixel. This also tracks three different colors Red, Blue and Green from a video. If there's any color from Red, Blue and Yellow or all the three at the same time in the live stream, rectangular boxes of Red color for tracking of Red, blue rectangular box for Blue color and Green for Green color will bound the respective color objects and the name of the color is displayed on top of it. This helps in recognizing colors and in robotics. This type of system is used in driverless cars to detect traffic and vehicle back light and take decision to stop, start or continue driving. This also has many applications in industry to pick and place different colored objects by the robotic arm. This is done using OPENCV using python programming language.

LIST OF FIGURES

1.INTRODUCTION

1.1 Project Objective:

For a robot to visualize the environment, along with the object detection, detection of its color is also very crucial . Color detection is necessary to recognize objects, it is also used as a tool in various image editing and drawing apps. Some real world applications are: it is used in self driving cars to detect traffic signals and is also used in some industrial robots, to perform pick-and-place task in separating different colored objects. This design system can be implemented in various fields for various purposes such as Defence, industrial purposes, games, automation, security, monitoring etc. Even these systems can also play a vital role in field of radar and navigating such as detecting, tracking of a moving colored object etc. Detection of color plays an important role even in the field of medical, in detection of color of skin, identification of a face, recognizing license plate.

1.2 Project Outline:

Color is one of the salient features of a picture. The detection of color in a live stream or in a graphic image can be employed in numerous scientific and industrial applications. Color detection is the elemental step in many image processing applications. In today’s graphical world, videos and images are omnipresent, it has been a complicated task to perform computer vision using robust and economical computer devices. Open-source computer vision library (OpenCV) paves a way to serve the rising demand of high-quality video and image processing. An image is always represented in the form of a matrix containing its pixel values. Images are often illustrated using several color models like CMYK, gray-scale, RGB, HSV etc. In this system, RGB model is employed to label the colors in a picture. Red, Green and Blue lights are mixed in numerous ways to generate ample amounts of colors. In RGB model image is represented in a matrix of PxQx3 pixels with P rows and Q columns of pixels in a picture. On which different operations can be applied to label the color in an image and while tracing an object in

3.RGB COLOR MODEL

3.1 Introduction:

The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.

The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors.

RGB is a device-dependent color model: different devices detect or reproduce a given RGB value differently, since the color elements (such as phosphors or dyes) and their response to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time. Thus an RGB value does not define the same color across devices without some kind of color management.

Typical RGB input devices are color TV and video cameras, image scanners, and digital cameras. Typical RGB output devices are TV sets of various technologies (CRT, LCD, plasma, OLED, quantum dots, etc.), computer and mobile phone displays, video projectors, multicolor LED displays and large screens such as Jumbotron. Color printers, on the other hand are not RGB devices, but subtractive color devices. This article discusses concepts common to all the different color spaces that use the RGB color model, which are used in one implementation or another in color image-producing technology.

4. HSV COLOR MODEL

4.1 Introduction:

A **color model** is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components. When this model is associated with a precise description of how the components are to be interpreted (viewing conditions, etc.), the resulting set of colors is called "color space." This section describes ways in which human color vision can be modeled. Unlike the RGB color model, which is hardware-oriented, the HSV model is user-oriented, based on the more intuitive appeal of combining hue, saturation, and value elements to create a color.

HSV is a cylindrical color model that remaps the RGB primary colors into dimensions that are easier for humans to understand. Like the Munsell Color System, these dimensions are hue, saturation, and value. Hue, saturation, and value are the main color properties that allow us to distinguish between different colors. Using color effectively is one of the most essential elements in photography, as color can draw the viewer's eye to your composition and affect the mood and emotional impact your photo.

It is important to note that the three dimensions of the HSV color model are interdependent. If the value dimension of a color is set to 0%, the amount of hue and saturation does not matter as the color will be black. Likewise, if the saturation of a color is set to 0%, the hue does not matter as there is no color used. Because the hue dimension is circular, the HSV color model is best depicted as a cylinder. This is illustrated in the interactive example below, where all possible color mixes are represented within the bounds of the cylinder.

of these three, all the different colors can be made. The following interactive allows you to play around with RGB.

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 many ways we can define a color 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. Csv file includes 865 color names along with their RGB and hex values.

Steps for detecting color in an image:

Here are the steps to build an application in Python that can detect colors:

5.3 Algorithm:

Image Capture: The first step is to fetch a high-quality image with resolution. To loadan image from a file we use Cv2.imread(). The full path of the image has to be given as input or else the input has to be in our working directory.

Img=cv2.imread(img path)

Extraction of RGB Colors: In this process, 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. The intensity value of each color ranges between 0(lowest) to 255(highest). By combining any 3 primary colors at different intensity levels a large variety of colors are produced.

For Example: If the intensity value of the colors is 0, this combination corresponds to black. If the intensity value of the primary colors is 1, this combination corresponds to white.

Index= ["color", "color_name", "hex", "R", "G", "B"]. This function calculates the minimum distance from the coordinates. The minimum distance is calculated by considering moving towards the origin point is circulated among all colors to find the most matching color. The pandas library serves as an important utility to perform various operations on comma-separated values. We need to read the csv file and upload it into the pandas data frame.

D = abs(R-int(csv.loc[i ,"R"])) + abs (G-int (csv.loc[i ,"G"])) + abs (B- int (csv.loc [i,"B"]))

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. The color name and its intensity level will be displayed in a rectangular box.

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

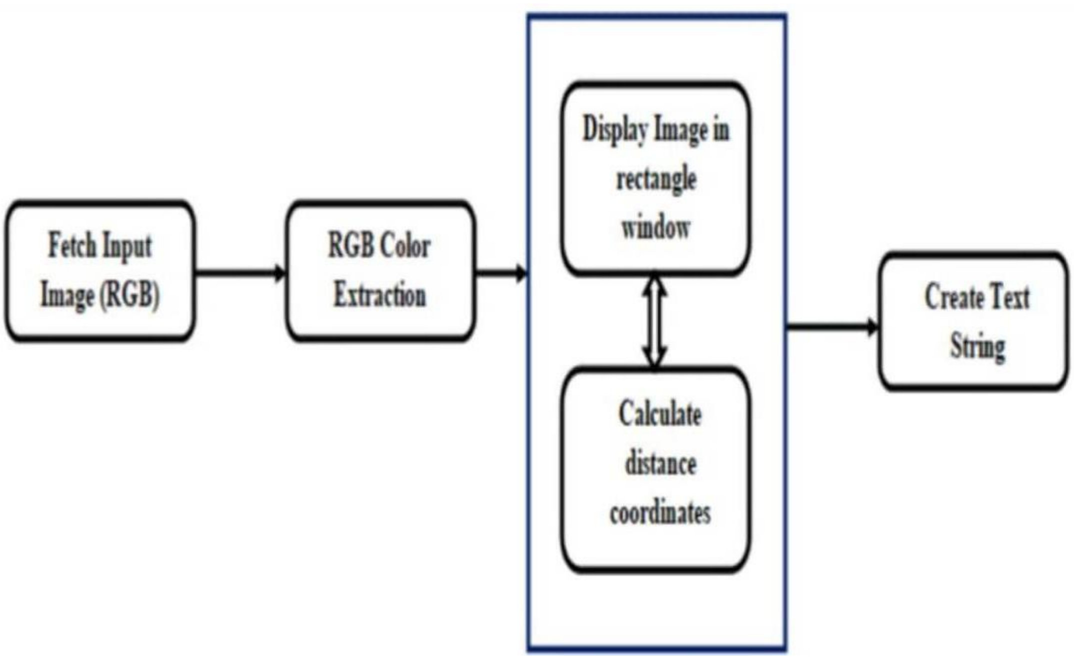

Figure 5.1 Architecture diagram for color detection

Image processing is one of the main reasons why computer vision continues to improve and drive innovative AI-based technologies. From self-driving cars to facial recognition technology—computer vision applications are the face of new tech.
Data annotators draw these rectangles over images, outlining the object of interest within each image by defining its X and Y coordinates. This makes it easier for machine learning algorithms to find what they're looking for, determine collision paths, and conserves valuable computing resources. Bounding boxes are one of the most popular image annotation techniques in deep learning. Compared to other image processing methods, this method can reduce costs and increase annotation efficiency.

6.8.5 Using Bounding Boxes for Object Detection:

But how does object detection work in relation to bounding boxes for this we require looking at object detection as two components: object classification and object localization. In other words, to detect an object in an image, the computer needs to know what it is and where it is.
Take self-driving cars as an example. An annotator will draw bounding boxes around other vehicles and label them. This helps train an algorithm to understand what vehicles look like. Annotating objects such as vehicles, traffic signals, and pedestrians makes it possible for autonomous vehicles to maneuver busy streets safely. Self-driving car perception models rely heavily on bounding boxes to make this possible.
However, it's important to note that a single bounding box doesn't guarantee a perfect prediction rate. Enhanced object detection requires many bounding boxes in combination with data augmentation techniques.

6.8.6 Common Use Cases for Bounding Boxes:

There are a variety of use cases for image processing and bounding boxes. Some of the more popular ones include:

- Self-driving cars
- Insurance claims
- Ecommerce
- Agriculture
- Healthcare

Bounding boxes are used in all of these areas to train algorithms to identify patterns. An insurance company may leverage machine learning to document insurance claims for car accidents, while an agriculture company could use it to identify what stage of growth a plant is in.

6.9. Output:

To achive the goal the first step is to capture the video through webcam.This is taken as input .The image frames are accessed from the video stream.The HSV color model can handle changes caused by lighting.Hence the image stored in RGB format has to be transformed into HSV .This color model describes color in terms of the amount of gray and their brightness value.Hue value is extended from 0-179,Saturation value is extended from 0-255 and Value is extended from 0-255 respectively.The corresponding is made by defining the range of each color (red,green and blue).Noise causes internal imperfections,hence it has to be removed.Morpholoagical technique called diation is used for this purpose.To specifically detect red,green and blue and discard others ,bitwise and is performed between the image frame and mask.All the points which are having same color or intensity have to be bounded by a rectangular box .As the object with red, green or blue color advances in the live recording, the bounding box alse advances with it .Here the goal for detecting and tracing the red ,green and blue colored object is achieved