Project Title: Color Identification in Images

Akash Madhukar Polkar

Master's Student in Mechatronics | RWU Hochschule Ravensburg-Weingarten
Email: akashpolkar29@gmail.com

Date: 21 November 2024

Project Overview:

The Color Identification in Images project leverages computer vision techniques to detect and identify specific colors in images. The goal is to enable a system that can analyze an image, identify various color regions, and provide relevant information such as color names, RGB values, and HEX codes in real-time.

Flowchart

1. Introduction

The **Color Identification in Images** project aims to develop a system capable of detecting and identifying specific colors present in digital images using computer vision techniques. The goal is to enable real-time color classification, where each region of a color can be highlighted, named, and represented in RGB and HEX formats. This project can serve as an essential tool for tasks involving digital design, brand analysis, and educational applications in color recognition.

2. Features

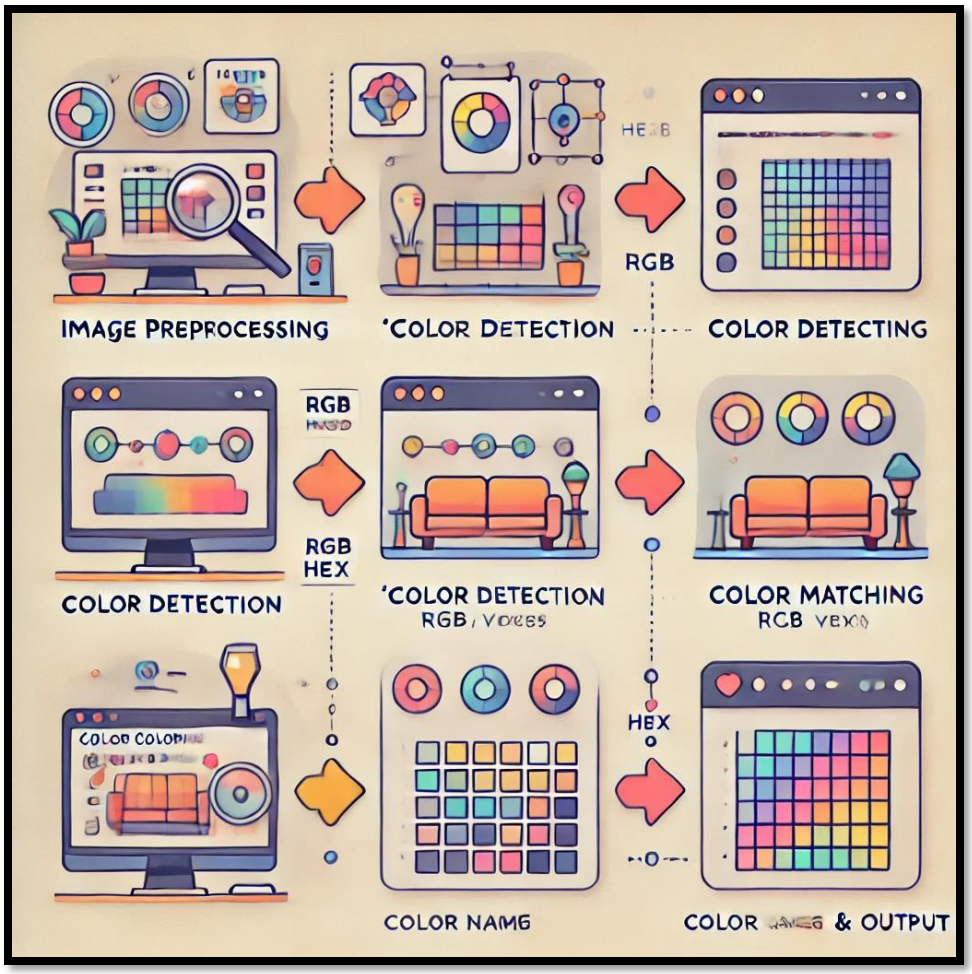
- **Automatic Color Detection**: The system can detect and extract prominent colors from an image, even those present in small areas. This ensures that all significant colors in the image are captured and analyzed.
- **Color Code Representation**: Detected colors are displayed in both RGB and HEX formats. This feature ensures easy integration of the identified colors into other applications, such as web design or graphic design tools.
- **Color Naming**: Once the color is detected, the system matches it with a predefined database of colors and provides a name for the color, such as “Sky Blue” or “Crimson.” This feature simplifies color identification for users who may not be familiar with specific color names.
- **Customizable Thresholds**: The system allows users to fine-tune detection thresholds to adapt to various image qualities and lighting conditions, improving the accuracy of color identification in diverse environments.

3. How It Works

The program follows a series of steps to detect and identify colors in an image:

1. **Image Preprocessing**: The image is loaded into the system, and its size is adjusted. The image is then processed in the RGB color space to extract pixel values.
2. **Color Detection**: Colors are detected by analyzing the image pixel by pixel, identifying distinct color regions in the image.
3. **Color Matching**: Each detected color is compared against a predefined set of color values stored in a CSV file. The closest match is found using the Euclidean distance between the RGB values of the detected color and those in the database.
4. **Color Naming and Output**: The closest match for the detected color is assigned a name from the database. The system displays the color’s RGB and HEX values, along with its

name, on the image. This output is shown in real-time, updating as the user interacts with the image.



Work Flow

4. Application Areas

- **Digital Design and Art**: Designers can use this tool to identify and extract color schemes from images. It is especially useful for web developers, graphic designers, and other professionals in creative fields.
- **Brand Analysis**: The tool can be used to analyze brand logos and digital marketing material to check for consistency in the use of brand colors across different media.
- **Educational Tools**: The system can serve as an interactive teaching tool for understanding color theory, helping students and beginners identify and learn about different colors in images.

5. Technologies Used

This project utilizes a range of technologies, libraries, and tools:

- **Python**: The programming language used for implementing the entire system, chosen for its versatility and strong support for libraries in computer vision and data analysis.
- **OpenCV**: A widely used library for computer vision tasks. OpenCV provides functions for reading images, detecting color values, and handling real-time image display and interaction.
- **NumPy**: Used for efficient handling of numerical operations, particularly for image processing and pixel value manipulation.
- **Pandas**: Employed to manage the predefined color database, allowing efficient access to color data (such as RGB values) for matching detected colors.
- **Color Database**: A CSV file containing predefined colors with their respective names, RGB values, and HEX values. This database is used to compare and find the closest match for any detected color.

6. Code Explanation

The project consists of several key functions:

1. **Loading and Displaying the Image**: The image is loaded using cv2.imread() and resized using cv2.resize() for better visualization.
2. **Color Matching**: The function get_color_name() is used to match detected RGB values with the closest color in the database by calculating the Euclidean distance.
3. **Mouse Event Handling**: The function draw_function() listens for double-clicks on the image. When a double-click is detected, the function retrieves the RGB values of the pixel and displays the closest color name.
4. **Real-Time Output**: The system updates the displayed image in real-time, drawing a rectangle with the identified color name, and showing its RGB and HEX values.

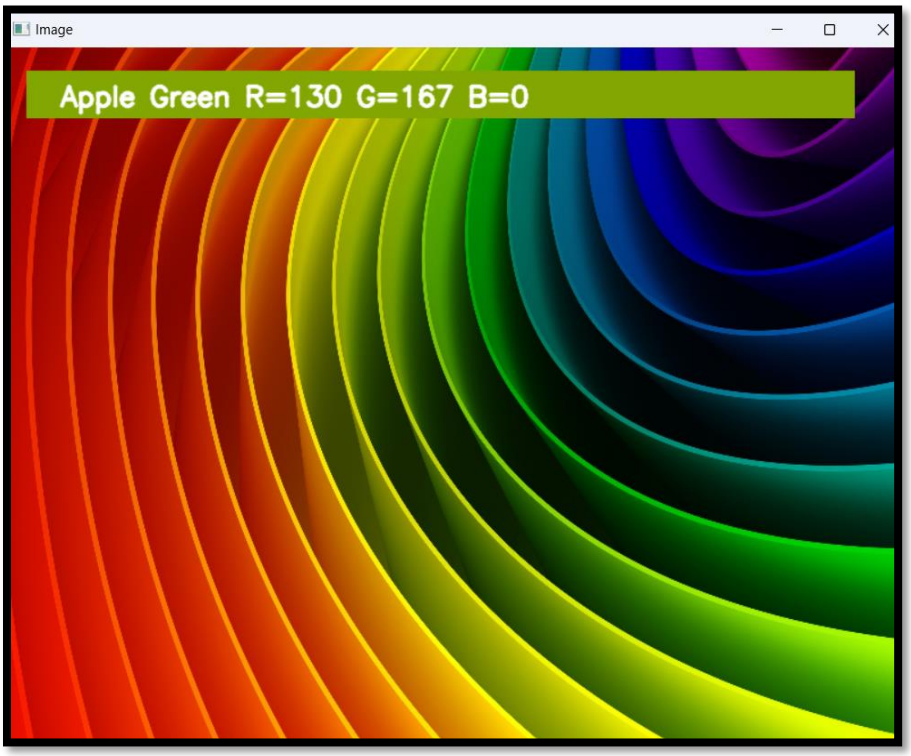
7. Results

Below is the output generated by the program when the user double-clicks on a pixel within the image:

- The color name is displayed along with its corresponding RGB and HEX values.
- A filled rectangle is drawn on the image displaying the selected color.

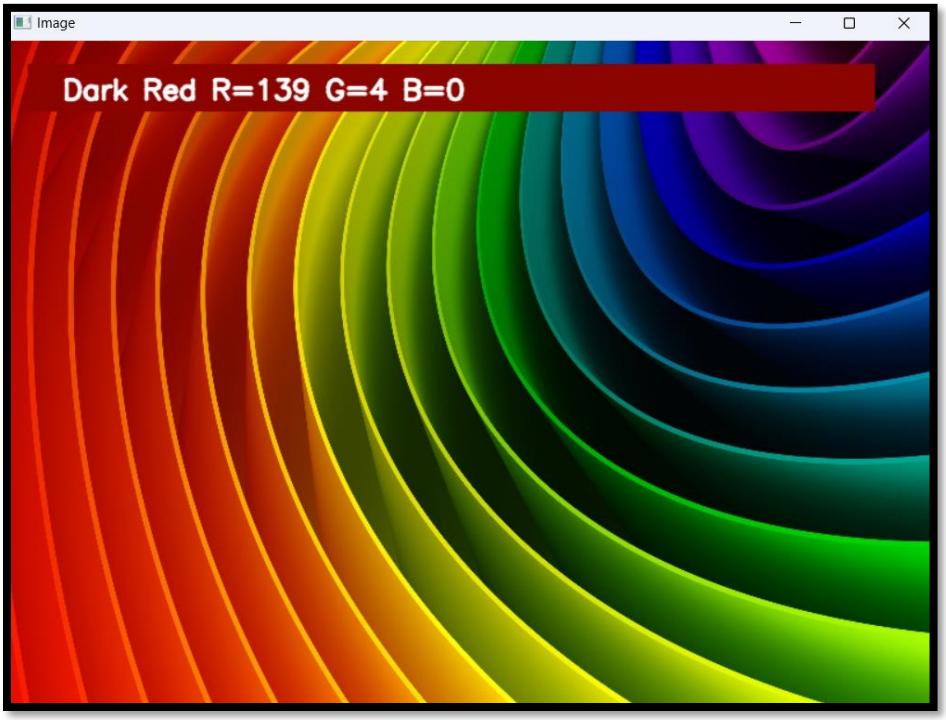
Sample Output:

1. Color Detected:
    - Color Name: Apple Green
    - RGB Value: (130, 167, 0)
    - HEX Code: #76CD2



Result

2. Color Detected:
    - Color Name: Dark Red
    - RGB Value: (139, 4, 0)
    - HEX Code: #8B0000



Result

7. Challenges and Limitations

- **Lighting Conditions**: The detection of colors can be affected by varying lighting conditions in the image, which may cause colors to appear differently.
- **Color Similarity**: In some cases, colors may appear too similar to each other, leading to mismatches. Adjusting detection thresholds can improve accuracy, but some level of ambiguity may persist.
- **Performance**: The system can be further optimized for handling large images or real-time video processing, as the current implementation might not be as efficient in these scenarios.

8. Conclusion

The **Color Identification in Images** project provides an effective and interactive solution for identifying and naming colors in digital images. With applications in digital design, brand analysis, and education, this tool is a valuable asset for various industries. The system's real-time functionality and integration with a color database ensure that it can be used in diverse environments, providing users with accurate color data in an accessible and intuitive way.

9. References

1. OpenCV Documentation. (n.d.). Retrieved from https://docs.opencv.org
2. NumPy Developers. (n.d.). NumPy Documentation. Retrieved from https://numpy.org/doc/
3. Pandas Documentation. (n.d.). Retrieved from https://pandas.pydata.org/