

Hackathon Submission Template (Level-1-Solution)

Use Case Title: Color Detection from Images

Student Name: Anupriya.S

Register Number: 511323205006

Institution: Kingston Engineering College

Department: Information Technology

Date of Submission: 19-05-2025

1.Problem Statement

Designers, developers, and artists often need to extract precise color values from images for brand consistency, content analysis, or digital accessibility. However, manually identifying colors from images is inefficient and error-prone. The problem is to build a Color Detection Application that allows users to upload image, click any point in it, and instantly view the RGB value and closest color name, improving workflow speed and accuracy.

2.Proposed Solution

The proposed solution is a web-based Color Detection Application built using OpenCV and Streamlit. The application enables users to upload an image and click on any area to detect the color at that pixel. It retrieves the RGB values, compares them with a CSV dataset of known color names, and displays the closest match. The UI also shows a color-filled box for easy visual reference, making it practical for design-related tasks.

3.Technologies & Tools Considered

- Programming Language: Python.
- Libraries/Frameworks: OpenCV, Pandas, NumPy, Streamlit.

APIs/Other Tools:

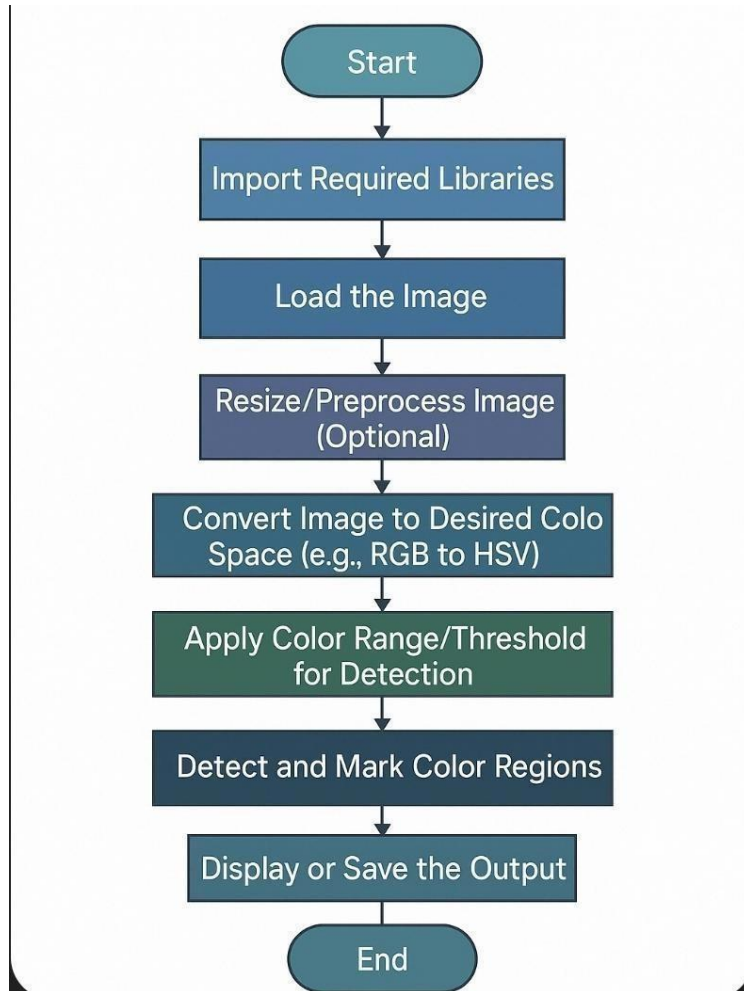
- colors.csv dataset for matching color names.
- GitHub for version control and code hosting.

4.Solution Architecture & Workflow

Major Components:

- Image Upload Module: Accepts user images via a Streamlit file uploader.
- Pixel Click Event Handler: Captures mouse click and fetches pixel RGB values.
- Color Matching Algorithm: Compares extracted RGB with values from a dataset.
- UI Display Module: Shows RGB, color name, and filled color box.

5.Workflow:



6.Feasibility & Challenges

- **Feasibility:** The solution is technically feasible using widely available libraries like OpenCV and Streamlit. It requires only basic image processing and data matching techniques, making it suitable for rapid prototyping and deployment.

● Challenges:

- Handling incorrect or unsupported file formats
- Matching RGB values accurately despite slight differences
- UI responsiveness during high-resolution image processing
- Ensuring real-time detection performance in browser-based usage

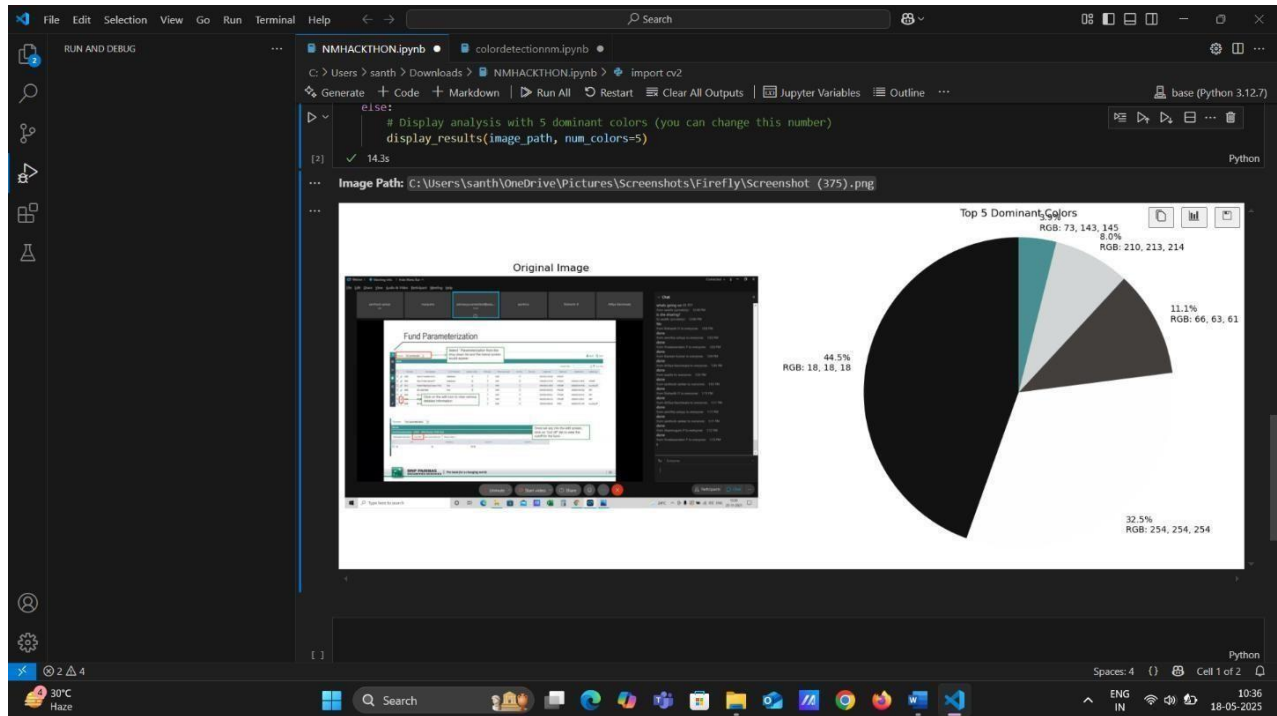
7.Expected Outcome & Impact

This project will deliver a functional tool that simplifies color detection from images. Users can interactively identify colors, supporting brand consistency, content design, and accessibility checks. It benefits graphic designers, UI/UX developers, and anyone needing precise color analysis.

8.Future Enhancements

- 1) Add HEX and HSV color format detection.
- 2) Integrate accessibility checks for contrast and readability.
- 3) Enable color palette extraction from the entire image.
- 4) Allow saving of favorite detected colors.
- 5) Support drag-and-drop for image uploads.
- 6) Build mobile-friendly version of the application.

9.Expected Output



10.Application Features

- Upload and display an image using a user-friendly interface.
- Detect and capture RGB values from any clicked point in the image.
- Compare the RGB value with a dataset of color names.