Hackathon Submission (Level-1-Solution)

Use Case Title: Color Detection from Images

Student Name: Danish P

Register Number: 712523205015

Institution: PPG Institute of Technology

Department: Information Technology

Date of Submission: 17/05/2025

---

1. Problem Statement

Designers, developers, and artists frequently need to identify precise colours from images to ensure visual consistency, maintain brand identity, or evaluate accessibility. Traditional image editing tools are often too complex or time-consuming for this simple task. There is a need for a lightweight, user-friendly application that allows users to detect and identify colour names and RGB values directly from any point in an uploaded image.

Proposed Solution

We propose a **Colour Detection Application** that enables users to:

- Upload any image.

- Click on any point in the image or select coordinates.
- Extract the RGB values of the selected pixel.
- Match the colour with a predefined dataset of colour names.
- Display the name of the closest colour and a visual preview.

This tool aims to streamline colour identification, especially for brand-focused designs, UI work, and accessibility checks.

Key Features:

- Upload and display images.
- Detect RGB values on user selection.
- Match colour using a dataset (colors.csv).
- Display colour name, RGB values, and colour swatch.
- Clean and responsive web-based user interface.

Technologies & Tools Considered

Python - Core programming language

Streamlit - For building the interactive web app

OpenCV - Image processing and pixel detection

NumPy - Array manipulation and pixel access

Pandas -Handle the colour dataset (colors.csv)

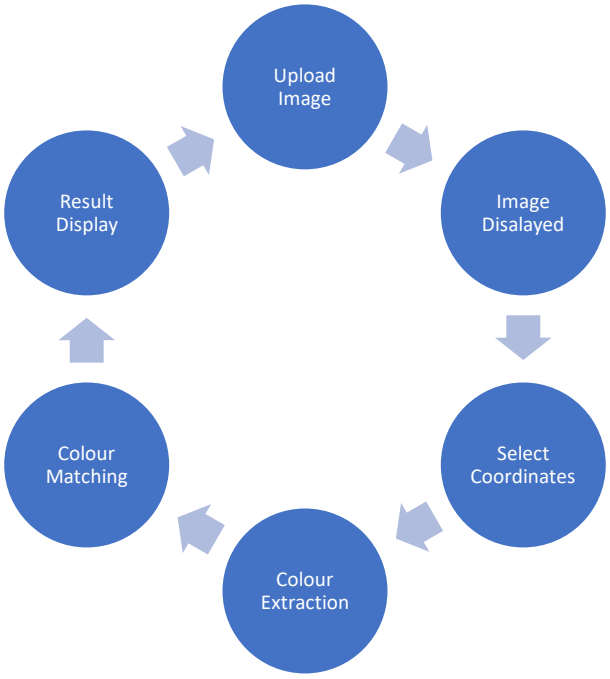
Pillow (PIL)- Image format compatibility

GitHub- Version control and documentation

2. Solution Architecture & Workflow

Architecture Components:

- **Frontend**: Streamlit UI for uploading images, displaying results, and handling input.
- **Backend**: Python script to process image, extract pixel data, and match colour.
- **Data Layer**: colors.csv containing predefined colour names and their RGB values.



3. Feasibility & Challenges

Feasibility

- The proposed solution uses lightweight, open-source tools.
- Easy to deploy on local machines and platforms like Streamlit Cloud.
- Minimal hardware requirements.
- Modular and extensible for future features.

Potential Challenges & Mitigations

Challenge	Solution
Pixel-accurate selection in browser	Use slider-based coordinate selection or integrate OpenCV GUI for local use

Challenge	Solution
Inaccurate colour matching	Upgrade to perceptual distance algorithms (e.g., CIEDE2000)
Large image files	Auto-resize images or limit upload size
UI compatibility	Ensure responsiveness across devices using Streamlit features

4. Expected Outcome & Impact

Expected Outcomes:

- Functional web app for real-time colour detection
- RGB and colour name display
- Easy-to-use, intuitive UI
- Well-documented GitHub repository

Impact:

- Helps designers and developers maintain visual and branding consistency
- Useful in digital content accessibility testing
- Aids educators and students in design-related fields

- Reduces dependency on heavy tools like Photoshop for simple colour identification

5. Future Enhancements

- Click-based colour detection (real-time OpenCV window)
- HEX value display for use in CSS/web
- Colour blindness simulation for accessibility checking
- Webcam/live camera feed input
- Option to export detected colours as palettes
- Mobile-friendly or native app versions
- Machine learning for enhanced colour name prediction