Intern Assignment: ColorFinder - Python

Overview

Your task is to develop a Python program that allows a user to upload an image, analyze the image, and detect the major colors present in it. The program should then return the major colors in RGB or any valid web format, ensuring that the color extraction process is efficient. The program will return the image name and a list of major colors in a structured JSON format.

Assignment Details

Input

- An image file uploaded by the user. Supported formats include JPEG, PNG, GIF, etc.

Output

- A JSON response containing:
  - **Image Name**: The name of the uploaded image.
  - **Major Colors**: A list of the major colors present in the image.
    - The colors should be represented in either RGB or HEX format.
    - The number of major colors should vary between 4 and 10.

Key Requirements

1. **Major Color Extraction**:
   - The program should detect between 4 and 10 major colors in the image using an efficient approach.
   - Noise around the image (e.g., subtle color variations or shadows) should be ignored.
   - Use **KMeans clustering** for color extraction.
2. **Output Format**:
   - The output should be returned as a JSON object containing the image name and a list of detected major colors.
   - Example output format:

```
{
  "image_name": "sample_image.jpg",
  "colors": [
    "rgb(123, 45, 67)",
    "rgb(255, 220, 180)",
    "rgb(100, 120, 130)",
    "rgb(90, 60, 40)"
  ]
}
```

Tools and Libraries

- **Python**: Core language for this assignment.
- **Libraries**: (just suggested not necessary to use)
  - `Pillow` for image processing.
  - `scikit-learn` for KMeans clustering.
  - numpy for numerical operations.
  - `json` for returning the output in JSON format.

Steps and Guidelines

1. Image Processing and Color Extraction

- **Load and Preprocess Image**:
  - Use `Pillow` to load the uploaded image and convert it to RGB format.
  - Optionally, resize the image to speed up processing for larger images.
- **Extract Pixel Data**:
  - Convert the image into a numpy array containing RGB pixel values.
- **Apply KMeans Clustering**:
  - Use **KMeans clustering** to group the pixels into clusters that represent the major colors.
  - Set the number of clusters between 4 and 10.
- **Convert RGB to HEX**:
  - Convert the RGB values of the cluster centers into HEX or RGB format to represent the major colors.

2. Return the Major Colors

- After clustering, convert the cluster centers to a valid color format (either RGB or Hex).
- Return the color list along with the image name in a JSON format.

3. Example JSON Response

- The program should return a JSON response like the following:

  ```json { “image_name”: “sample_image.jpg”, “colors”: [ “rgb(123, 45, 67)”, “rgb(255, 220, 180)”, “rgb(100, 120, 130)”, “rgb(90, 60, 40)” ] }
  ```

4. Evaluation Criteria

- Correctness

Ensure the program correctly identifies and returns the major colors in the range of 4-10 colors.

- Performance

The program should process images efficiently, particularly for larger images.

- Code Quality

The code should be clean, modular, well-commented, and easy to understand.

- Error Handling

Handle cases such as invalid image formats or very small images gracefully.

5. Submission

Create a GitHub repository with the following:

- Python script (color_extractor.py or similar) containing the code.
- Sample images for testing the functionality.
- A README.md file explaining how to run and test the code.
- Ensure the code runs efficiently and produces correct results within the given constraints.

Note: Interns are encouraged to be creative and innovative in their approach to designing the portal. The goal is to create a user-friendly and visually appealing platform for accessing brand information efficiently.
