Restoring Colors to a Grayscale Image

Python for Research

August 4, 2025

1 Introduction

Colorization of black-and-white (grayscale) images is a challenging task in image processing. While early approaches relied on manual painting, modern techniques use deep learning models to automatically assign realistic colors. This project explores different methods of image colorization, from simple rule-based approaches to pre-trained deep learning models.

2 Objective

Develop a Python-based application that takes a grayscale image as input and generates a full-color version using image processing techniques.

3 Description of the Project

Students will work on grayscale images (e.g., historical photographs or artistic black-and-white shots) and implement different approaches to colorization, including:

1. **Manual Colorization**: Assigning colors to specific image regions.
2. **Histogram-Based Color Transfer**: Using a reference image to transfer color information.
3. **Deep Learning-Based Colorization**: Using pre-trained neural networks for automatic colorization.

The goal is to compare different techniques and assess their effectiveness in generating realistic and vibrant colorized images.

4 Workflow

4.1 Step 1: Image Preprocessing

- Convert images to grayscale.
- Resize images to match model requirements.
- Normalize pixel values for deep learning models.

4.2 Step 2: Manual Colorization

Using OpenCV or PIL to apply color masks to specific regions of the image.

4.3 Step 3: Histogram-Based Color Transfer

- Select a reference image with a similar scene or lighting conditions.
- Transfer color histograms from the reference image to the grayscale image.

4.4 Step 4: Deep Learning-Based Colorization

- Use pre-trained deep learning models such as DeOldify or OpenCV's Colorization Model.
- Convert grayscale images into LAB color space.
- Use deep learning to predict color channels and merge with the original grayscale image.

4.5 Step 5: Evaluation and Visualization

- Compare results from different methods.
- Display the original grayscale image, manually colorized image, and AI-generated colorized image side by side.

5 Key Techniques and Libraries

- **OpenCV** (`cv2`) – Image processing, color space conversions.
- **NumPy** (`numpy`) – Image array manipulation.

- **Matplotlib** (`matplotlib.pyplot`) – Visualization.
- **Pre-trained Deep Learning Models**:
  - OpenCV's deep learning colorization model
  - DeOldify (State-of-the-art colorization using PyTorch)

6 Dataset Suggestions

- Public Domain Historical Images: Library of Congress
- Flickr Creative Commons Grayscale Images
- Custom Family Photos (students can use their own images)

7 Expected Outcome

- A Python application that colorizes black-and-white images.
- Comparison of different colorization approaches with insights into their strengths and limitations.
- Enhanced understanding of image colorization techniques, from rule-based to deep learning models.