# Internship Report: GenAI-NullClass Image Colorization Project

## Introduction

This internship focused on the development of advanced deep learning methods for **image colorization**, organized into three main tasks:

- Visualization of the colorization process
- Dataset augmentation for improved colorization
- Interactive user-guided colorization

Each task was implemented using Python (PyTorch), with Jupyter notebooks for code, model training, and evaluation. Saved model weights, augmentation scripts, and GUI prototypes support reproducibility and further experimentation.

## Task 1: Visualizing the Colorization Process

**Objective:**
Provide insights into how deep learning models perform image colorization by visualizing intermediate model activations, color difference maps, and performance metrics.

**Key Activities:**

- Developed and trained a deep neural colorization model (architecture details and training logic in `colorization_visualization.ipynb`).
- Extracted and visualized layer-wise **feature maps** to understand the model's progressive perception of color.
- Generated **color difference maps** between predicted and ground-truth images to locate key error areas.
- Plotted training and validation metrics (e.g., loss curves, accuracy) and perceptual metrics such as **SSIM** and **LPIPS** across epochs to monitor learning progress.

**Outcome:**
The visualizations facilitated interpretability, highlighting the model's strengths and potential failure modes at different pipeline stages. This diagnostic approach improved understanding for debugging and future model refinement.

## Task 2: Dataset Augmentation for Improved Colorization

**Objective:**
Boost the quality and generalization of the colorization model via data augmentation techniques tailored for grayscale/colorization datasets.

**Key Activities:**

- Implemented augmentation pipelines in `data_augmentation_colorization.ipynb` using transformations such as:
    - **Random rotations**
    - **Flipping (horizontal/vertical)**
    - **Brightness and contrast adjustments**
- Retrained the colorization model using the augmented dataset and tracked before/after performance differences.
- Conducted comparative evaluation using image quality and perceptual metrics (**SSIM**, **LPIPS**, **colorfulness**).

**Outcome:**
Augmentation led to:

- Minimum model accuracy of 80% (measured on held-out test data), increased further after augmentation.
- Noticeable improvements in the model's robustness to input variations, and better generalization to unseen samples as evidenced by metric improvements.

## Task 3: Interactive User-Guided Colorization

**Objective:**
Enable users to influence the colorization process by specifying region- or point-specific color hints.

**Key Activities:**

- Developed a notebook and a simple GUI interface (e.g., Streamlit app provided in `app/`) to let users:
  - Upload grayscale images
  - Mark/select regions and assign target colors
  - View dynamically colorized outputs reflecting user guidance
- Integrated user hints into the colorization model's inference logic, demonstrating flexible, user-driven results.

**Outcome:**
The system allowed for **personalized colorizations** based on user input, greatly enhancing control over output stylization. The GUI prototype supports both technical experimentation and a user-friendly demo.

## Evaluation & Conclusions

- All models were evaluated against standard metrics (confusion matrix, precision, recall, perceptual scores).
- Detailed metric comparisons and plots are documented within each notebook.
- Models and code are modular, facilitating further research, benchmarking, or extension.

## Repository Structure Overview

| Folder | Purpose |
|---|---|
| `Task1_Colorization_Visualization/` | Visualization notebooks, saved weights, and task documentation |
| `Task2_Data_Augmentation/` | Augmentation, training, evaluation scripts |

| | |
|---|---|
| `Task3_Interactive_User_Guided_Colorization/` | User-guided notebook, GUI application code, weights |
| `requirements.txt` | Python dependencies |
| `README.md` | Project overview and setup instructions |

## Usage Guide

- Navigate to each task's folder and run the Jupyter notebooks for code, experiments, and analysis.
- For Task 3, launch the GUI app (example: `streamlit run app/user_guided_colorization.py`) for hands-on colorization.
- Saved model weights and backup download links are included for quick setup.

## Final Note

This internship project demonstrates advanced computer vision techniques, practical augmentation strategies, and user-centered application design.
For questions, further collaboration, or sharing results, refer to the contact details in the main repository README.