Pixel Rearrangement Art Generator
Digital Image Processing Mini Project Proposal

Aadesh Tikhe
BCA Semester 5
Enrolment Number: A71004823070
Assisted by: Dr. C Kalpana
Amity University Mumbai

Abstract

This project presents a novel approach to artistic image transformation through pixel-level rearrangement based on grayscale intensity sorting. The system takes two input images: a source image and a destination template. By analyzing and sorting pixels by grayscale intensity values from both images independently, and then mapping the sorted source pixels to the sorted destination pixel positions, the system creates an artistic effect where the source pixels reorganize to form the destination image while retaining original color values. This demonstrates key digital image processing concepts including grayscale conversion, pixel manipulation, sorting, and contour detection. The method also enables optional pixel-migration animation.

1 Introduction

Digital image processing enables creative transformations beyond filtering and enhancement. This project explores an unconventional method: pixel rearrangement art. Unlike traditional morphing or style transfer that modifies intensity, this method preserves original pixel colors while reorganizing them spatially based on intensity rank mapping.

Applications include generative art, privacy-preserving transformations, educational visualization of sorting, and creative digital filters.

2 Literature Review

Pixel sorting gained attention through Kim Asendorf (2010). Traditional DIP literature focuses on histogram operations and spatial filtering, not pixel reordering. The presented method differs by modifying spatial arrangement alone while preserving all color values. It intersects with histogram analysis, sorting algorithms, and pixel-level manipulation.

3 Methodology

3.1 Image Preprocessing

1. Load source and destination images and resize them to equal dimensions.
2. Convert both images to grayscale using the weighted RGB formula:

$$Gray = 0.299R + 0.587G + 0.114B$$

3. Generate histograms to study intensity distribution.

3.2 Pixel Sorting and Mapping

1. Extract all pixels with coordinates and grayscale values.
2. Sort both sets by grayscale intensity.
3. Map the i-th source pixel to the i-th destination coordinate.

3.3 Image Reconstruction

1. Place each source pixel into its mapped destination location.
2. Construct output image.
3. Optionally generate frames for a pixel-migration animation.

3.4 Enhancement Using DIP Techniques

- Canny edge detection.
- Contour detection using CHAIN_APPROX_NONE and CHAIN_APPROX_SIMPLE.
- Morphological operations.
- Harris corner detection for feature distribution analysis.

3.5 Tools and Technologies

Python, OpenCV, NumPy, Matplotlib, Jupyter or VS Code.

4 Expected Results and Analysis

- Original source and destination images.
- Rearranged artistic output.
- Visual comparison grid.
- Histograms of grayscale intensity.
- Contour detection results on at least 5 image pairs.
- Harris corner detection on source, destination, and rearranged images.

5 Challenges

5.1 Technical Challenges

Computational complexity of sorting millions of pixels, memory usage, and the need for dimension matching.

5.2 Visual Quality Challenges

Intensity clustering, possible loss of fine detail, and color mismatch relative to destination structure.

5.3 Implementation Challenges

Parameter tuning, animation smoothness, handling grayscale or transparent images.

6 Conclusion and Future Scope

6.1 Conclusion

This project demonstrates a creative method for image transformation that preserves pixel color while reorganizing spatial structure. It integrates grayscale conversion, histogram study, sorting, contour detection, and corner detection.

6.2 Future Scope

Potential extensions include region-based sorting, interactive animation, alternative color spaces, video processing, CNN-based segmentation for selective rearrangement, GUI or mobile application versions.

7 References

1. Gonzalez and Woods, Digital Image Processing.
2. OpenCV Documentation.
3. Asendorf, Pixel Sorting (2010).
4. Bradski and Kaehler, Learning OpenCV.
5. Nixon and Aguado, Feature Extraction and Image Processing.