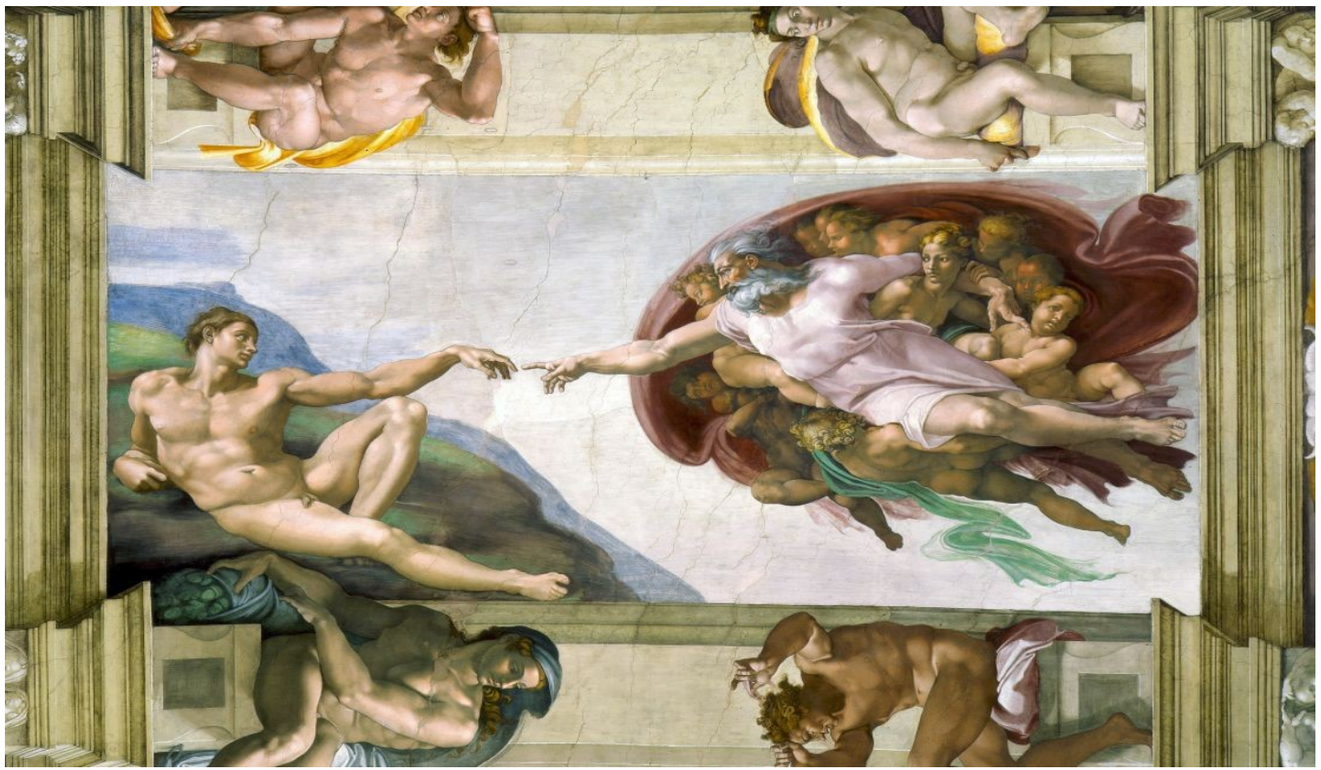
Project Leonardo: Using Image Recognition to Find Similar Art

Erdős Data Science Bootcamp Fall 2024

Team Members: Greyson Meyer, Sun Lee, Dawson Kinsman, Oulin Yu, Botan Cevik

GitHub Link: GitHub Repository



Overview of Project Leonardo

- **Objective**:

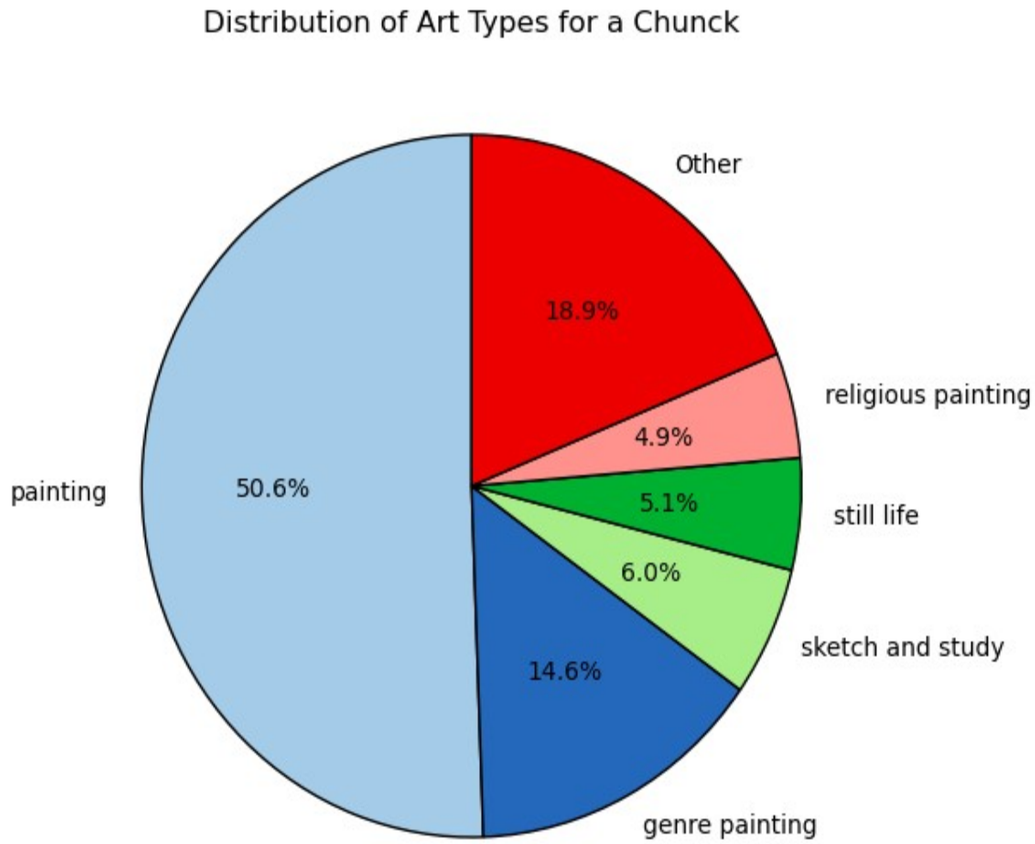
  To develop a recommendation system that identifies and suggests visually similar paintings based on color and composition analysis.
- **Motivation**:
  - Assist art curators, scholars, and art enthusiasts in finding visually similar artwork.
  - Create a new way of exploring online art databases using input images rather than text prompts.
- **Approach**:
  - Leveraging k-means clustering for color and composition features to recommend and visualize matches.

Key Objectives

- Extract and store meaningful visual features (color and composition) from a large dataset of images.
- Use k-means clustering to analyze dominant colors and structural features (contours and shapes).
- Recommend visually similar images based on extracted features and visualize top matches.
- Assess the clustering performance using metrics like silhouette scores.

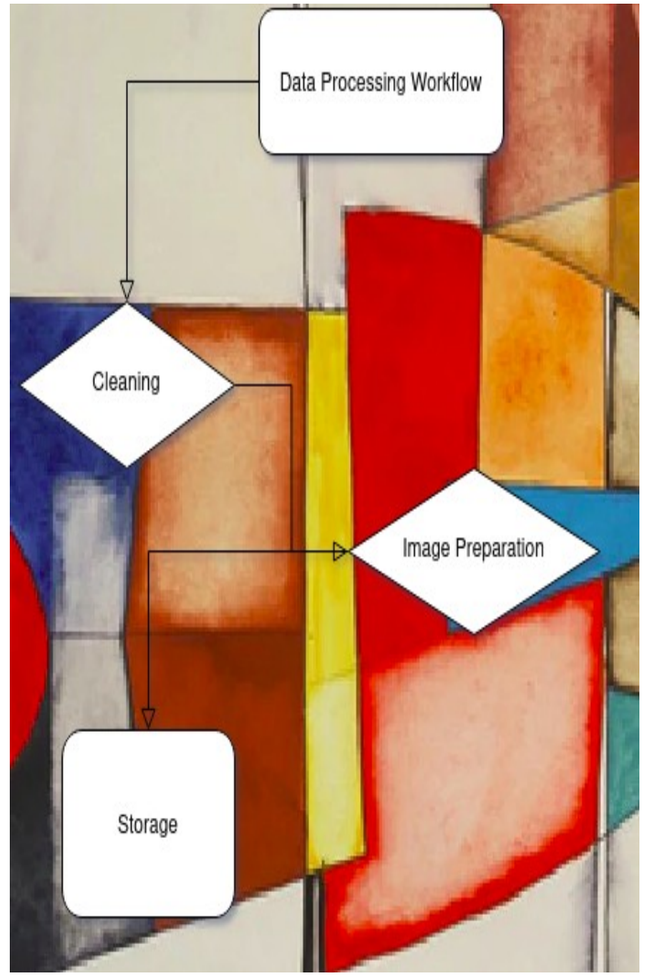
Dataset

- **Source**:
  - OmniArt Dataset (Restricted Version)
- **Contents**:
  - Over 15 online artwork collections and user-generated art.
- **Focus:**
  - Paintings and painting-adjacent entries.



Data Processing Workflow

- **Cleaning**:
  - Removed missing artist names, broken URLs, and irrelevant categories.
- **Image Preparation:**
  - Downloaded and resized images to consistent dimensions (e.g., 200x200 pixels).
- **Storage**:
  - Organized cleaned dataset into Parquet files for efficient retrieval and analysis.



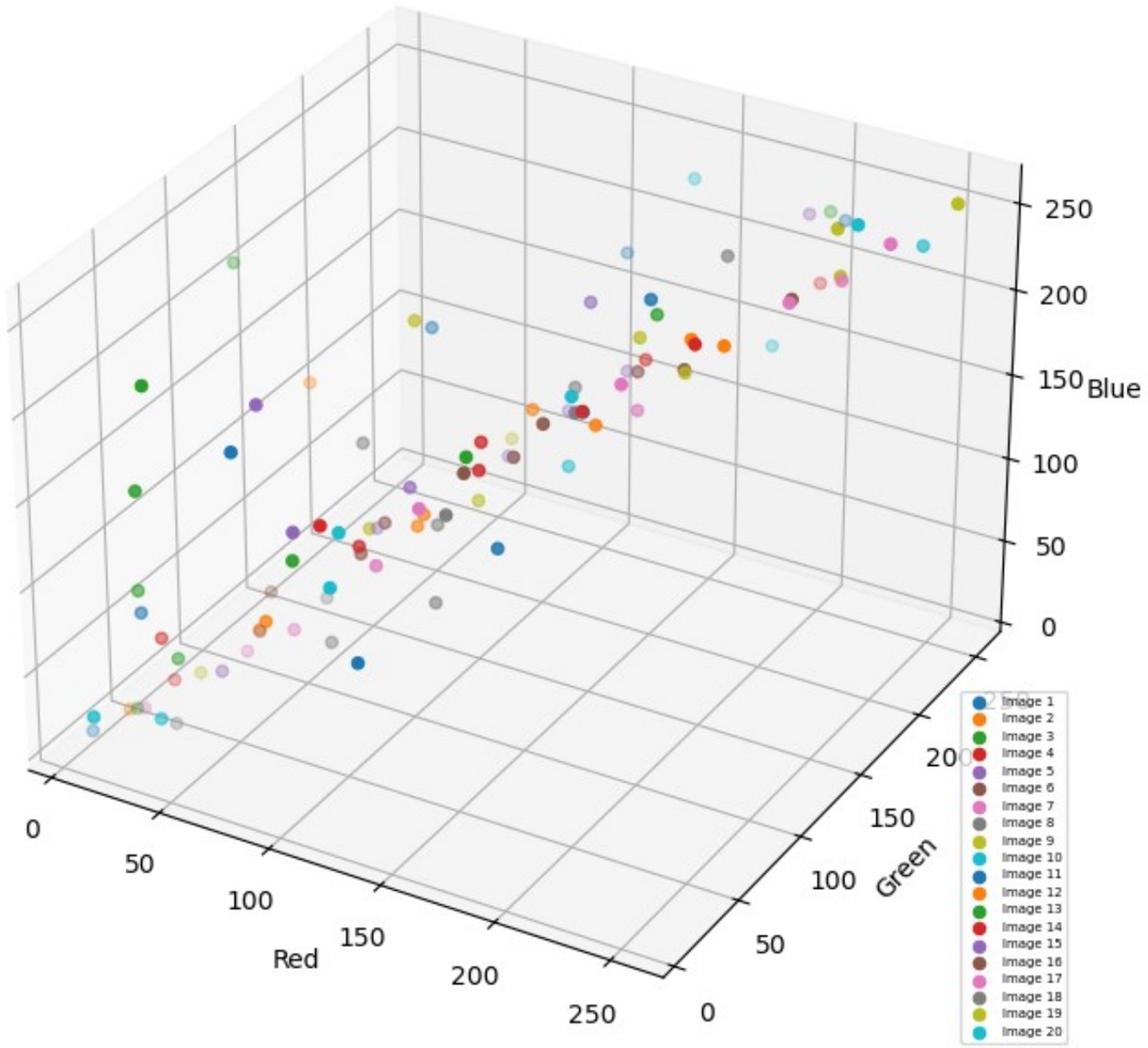
Feature Extraction with K-Means

- **Color Analysis:**
    - Clustering pixel values to identify dominant colors.

- **Composition Analysis:**
    - Using contours and shapes to determine spatial layout and structural patterns.

- **Performance Metric:**
    - Silhouette scores to validate clustering quality.

Example of Color Distribution of Images



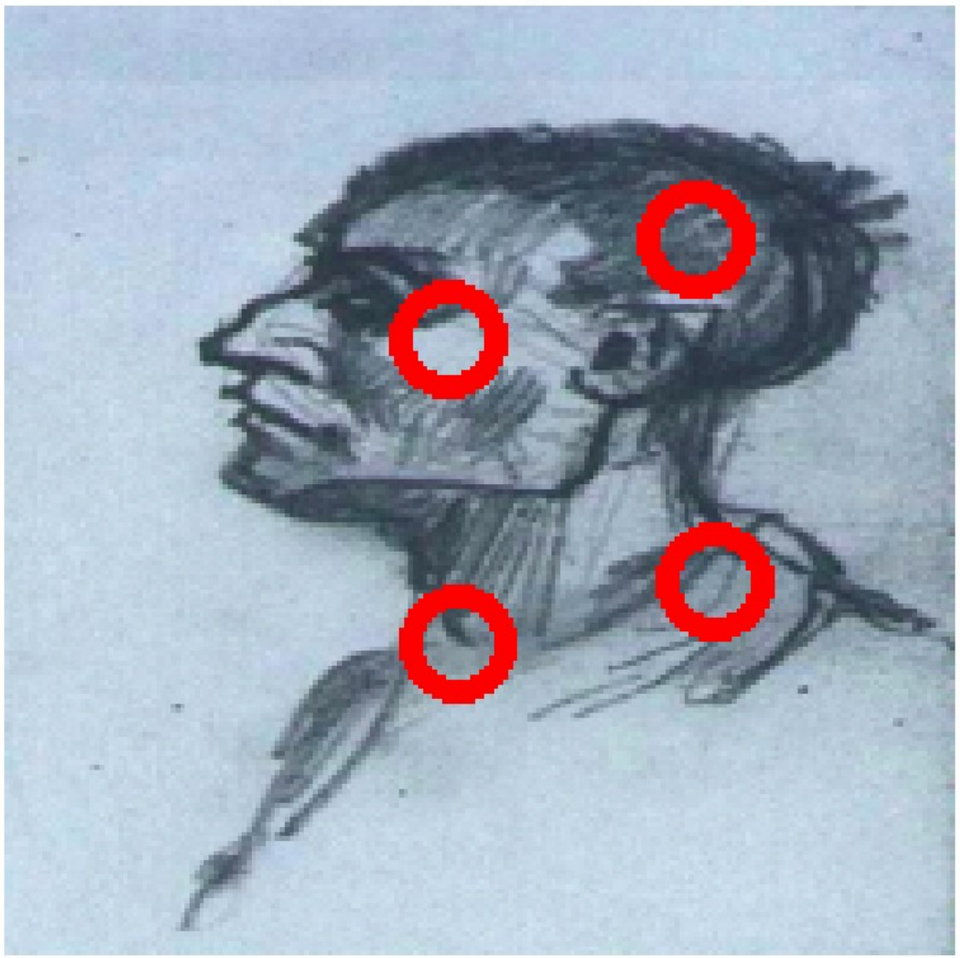
Color Clustering Examples

Example of Canny Edge Detection

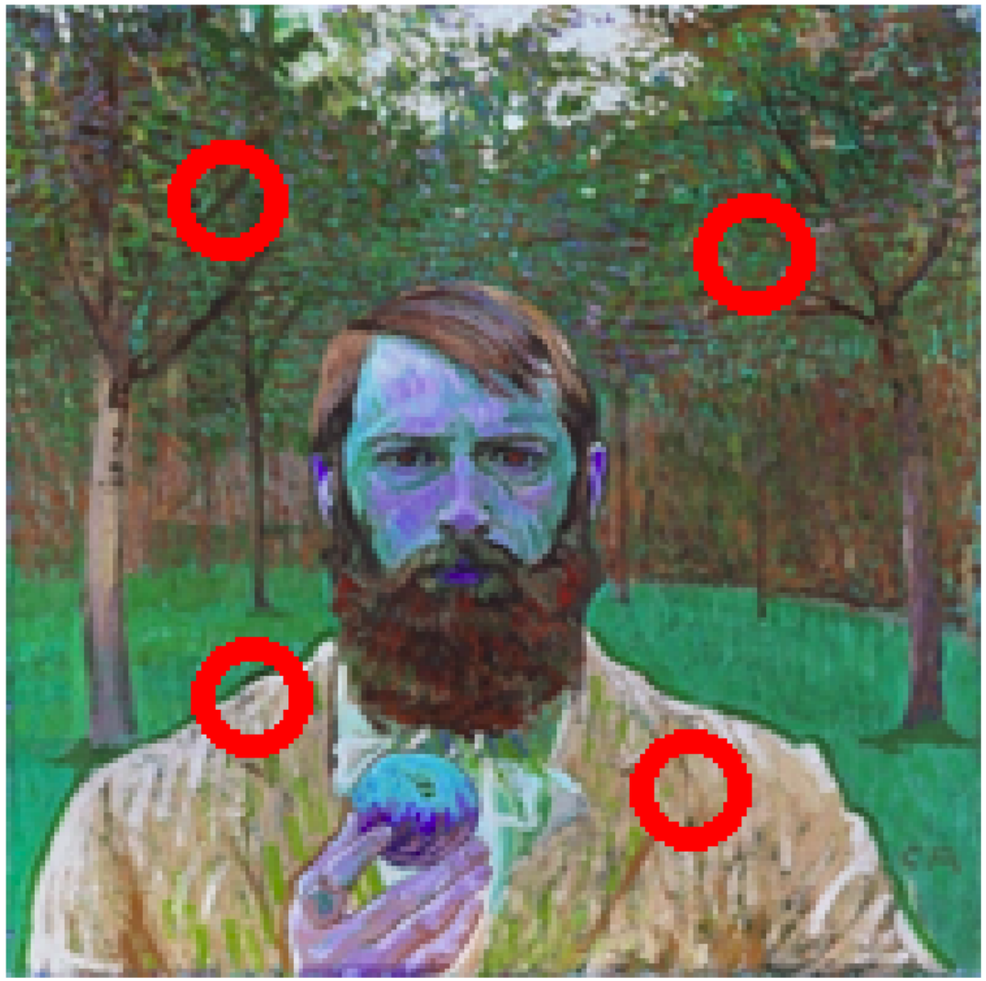
Composition Clusters

Composition clustering examples

Composition Clusters on Image 15 for chunk 14



Composition Clusters on Image 19 for chunk 14



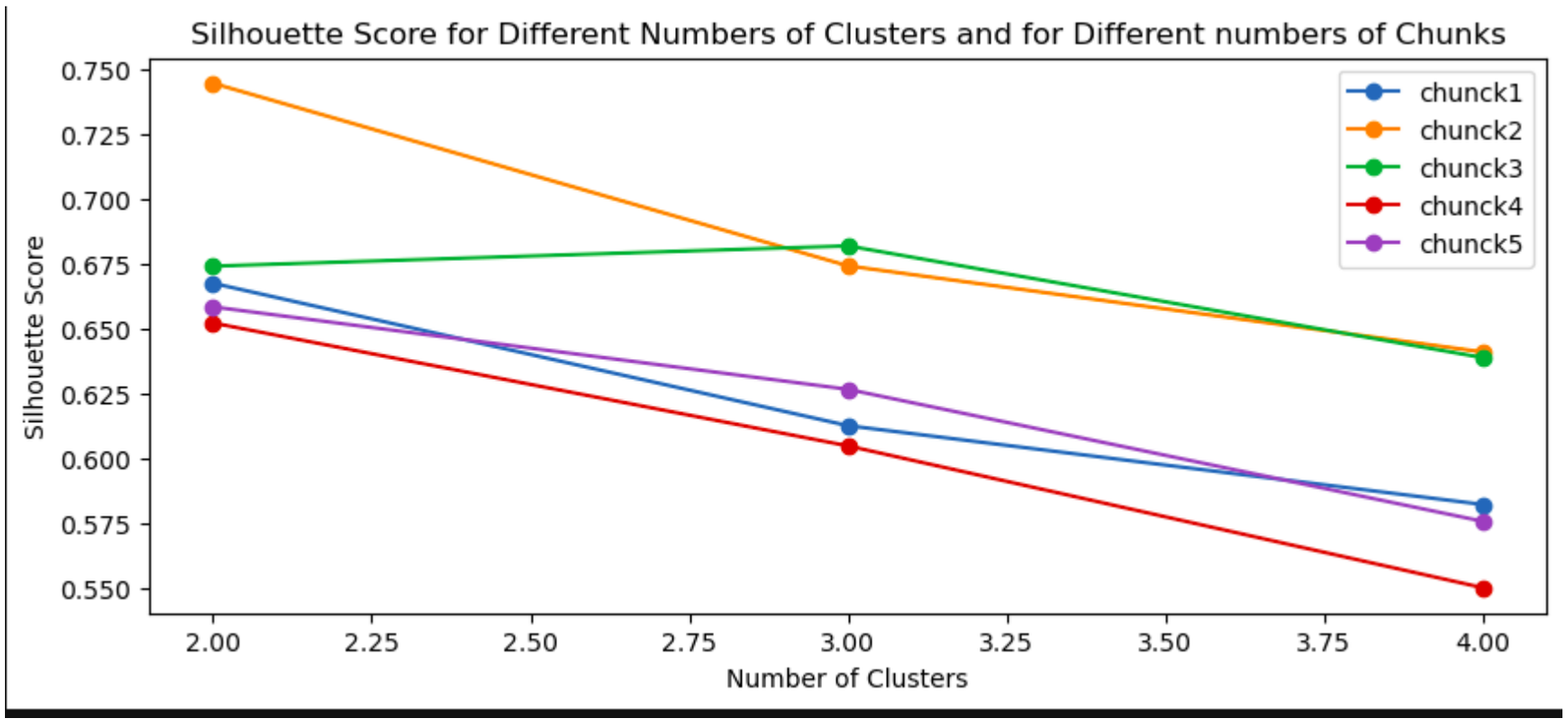
Recommendation System

- Input Image → Extracted Features (Color & Composition).
- Compare extracted features with dataset images.
- Output: Recommended images based on color, composition, and overall similarity.

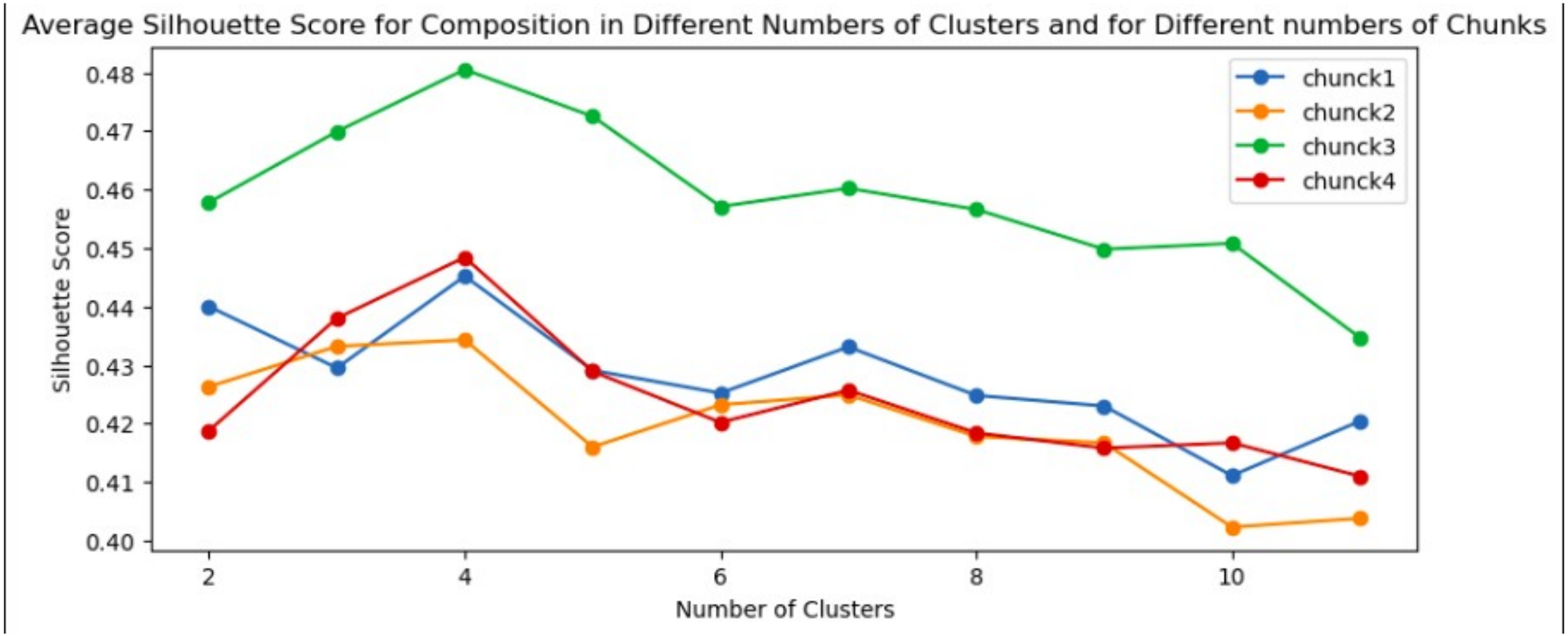
Performance Evaluation

- **Metric**:
    - Silhouette score to measure clustering quality.
- **Observations**:
    - Optimum number of clusters (k) identified using silhouette analysis.
    - Recommendations align closely with input features.

- We worked on randomly chosen chunks of image data to evaluate the optimal number of clusters for both color and composition features.
- By calculating the average silhouette score for each chunk, we assessed how well the images were clustered for different values of K.
- Our analysis revealed that the optimal number of clusters for color was 2 and for composition was 4, as it consistently yielded the highest silhouette scores across the chunks.
    - Here is the graph of average silhouette scores for color:



Here is the graph of average Silhouette Score for Composition:



Concern on the optimal number of cluster for Color:

- **Oversimplification**:
    - If you were to choose k=2, it would imply that you're grouping all the color data into just two major clusters. This might not fully capture the richness and diversity of colors in the artwork.

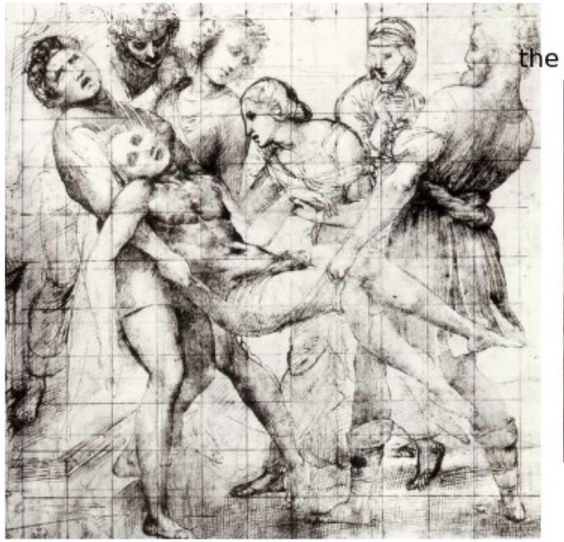
- **Loss of Detail:**
    - Choosing fewer clusters may ignore important differences in the image's color distribution. For example, a painting might have multiple distinct color regions (e.g., light and dark shades, or complementary color schemes), and k=2 might not capture the full complexity of these color variations. With only two clusters, nuances between these regions might be lost, making the clustering too basic for accurate comparisons.

Results

- **Examples :**

three dominican saints (detail) by giovanni battista piazzetta

modello for the entombment by raffaello sanzio



the battle of anghiari (detail) by leonardo da vin

Input image

przydrozne irysy/roadside irises by stokrotas

winter sofia by kalinatoneva

winter sofia by kalinatoneva

Input image

Future Iterations

- Integrate deep learning for advanced feature extraction (e.g., style, texture).
- Adaptive clustering for varying complexity.
- Interactive user interface for personalized recommendations.

Streamlit App

Project Leonardo · Strea

https://project-leonardo-wna6tfncgzxmvdwlzcblua.streamlit.app

90%

Project Leonardo

Statistically driven approach to find similar arts.

Image loader

Select an image...

Drag and drop file here

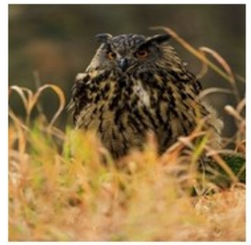
Limit 200MB per file • PNG, JPG, JPEG

Browse files

or input image URL

https://static.streamlit.io/examples/owl.jpg

Load Image



Number of clusters

4

Composition

0.50

0.00

1.00

Color

Art type

All

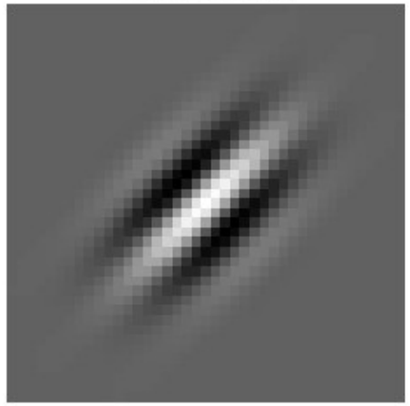
Why didn’t we use Gabor Filter?

- In the context of our image similarity model, we decided not to apply Gabor filters for feature extraction. While Gabor filters are powerful for texture analysis and can capture frequency and orientation information, they tend to remove important structural and color details in an image. Specifically, Gabor filters perform well in detecting local features such as edges and textures, but they may discard crucial high-level semantic information, which is important for accurately comparing images based on composition and color distribution.

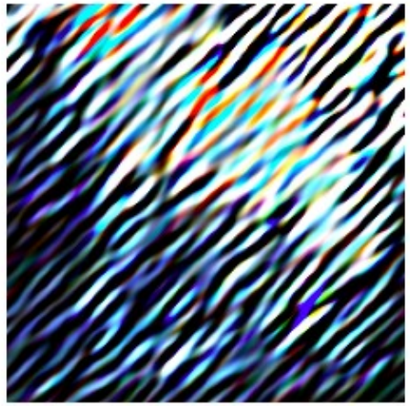
- For our image similarity model, which focuses on identifying more global features like color clusters and overall composition, using Gabor filters would have likely distorted or diminished essential details that are necessary for a more accurate similarity assessment.

Resized Image

Gabor Kernel

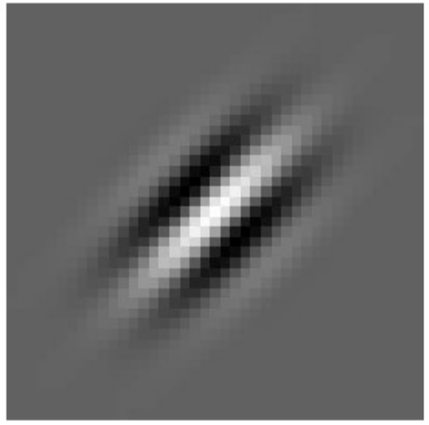


Filtered Image



Original Image

Gabor Kernel



Filtered Image

Acknowledgements

We would like to thank:

@Erdos Institute