**Performance Report: 🐱 Cat Image Mosaic Generator 🎨**

Short video demonstrating my approach: <https://www.loom.com/share/d1785685961f415085335ff359bfc2b3?sid=88a24d65-7c61-4ece-9319-146bbdceceff>

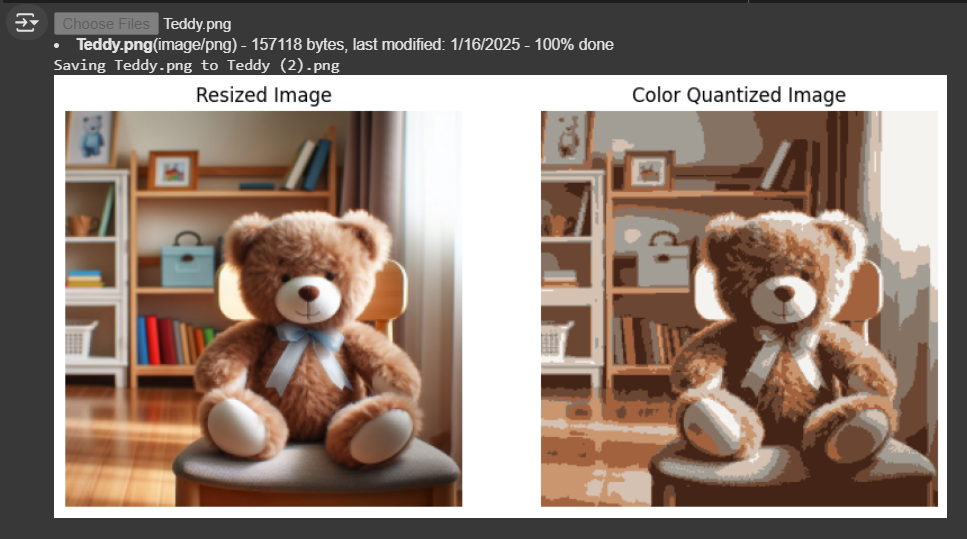
**1. Introduction**

This project implements an **Interactive Image Mosaic Generator** that transforms an input image into a mosaic representation using **cat images** as tiles. The method utilizes **color quantization, clustering, and similarity matching** to generate the final output. The application was deployed on **Hugging Face Spaces** using **Gradio** for an interactive UI.

**2. Methodology**

The process involves several key steps:

1. **Image Upload & Preprocessing:**
   * The user uploads an image via the Gradio interface.
   * The image is resized to a **fixed resolution** of 256×256 pixels for uniform processing.
   * Color quantization is applied using **K-means clustering** to reduce colors to **8 dominant colors**.



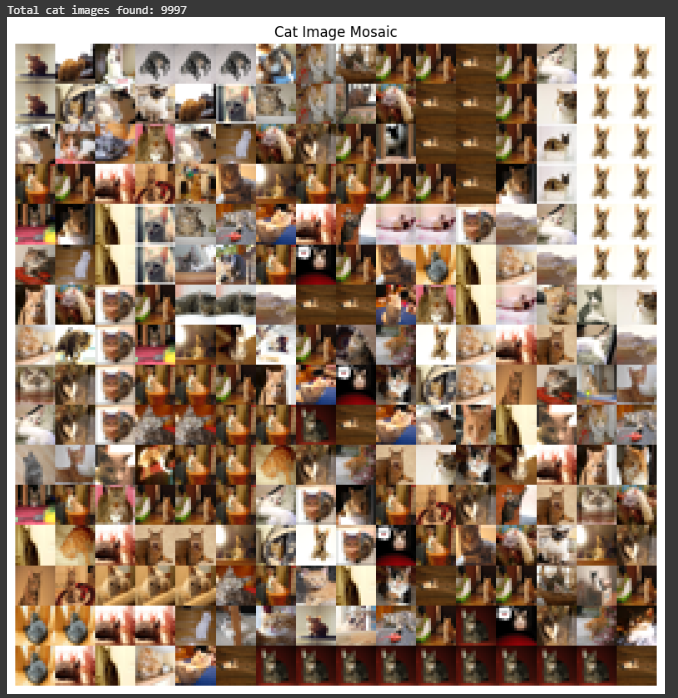
1. **Grid Creation & Feature Extraction:**
   * The quantized image is divided into **a grid of smaller cells** (e.g., 16×16).
   * The average color of each grid cell is calculated.



1. **Cat Image Dataset Processing:**
   * A dataset of **cat images** (from Kaggle) is downloaded and processed.
   * Each image is resized to match the **grid cell size**.
   * The **average color** of each cat image is computed.

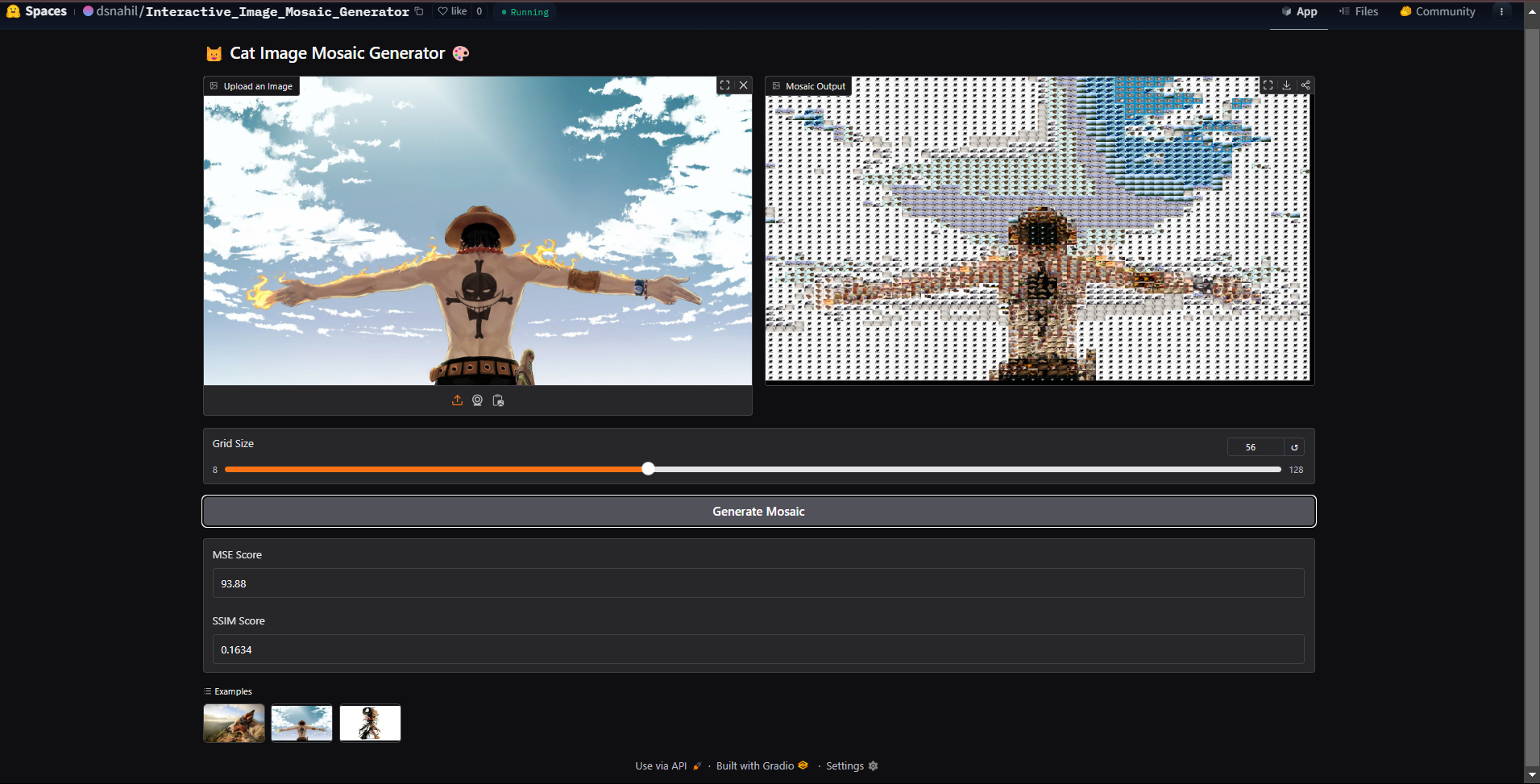
**Path to dataset files: /root/.cache/kagglehub/datasets/crawford/cat-dataset/versions/2**

1. **Mosaic Generation using Similarity Matching:**
   * Each grid cell's average color is compared with the average colors of cat images.



* + I tested the above teddy image for sample and also extracted the total number of images in my Kaggle dataset.
  + The closest matching cat image (based on **Euclidean distance**) is selected and placed in the mosaic.

1. **Performance Metrics Calculation:**
   * **Mean Squared Error (MSE):** Measures the pixel-wise difference between the original and mosaic images.
   * **Structural Similarity Index (SSIM):** Measures the structural similarity between the two images.



**3. Performance Metrics**

To evaluate the performance of the mosaic generation, we used:

| **Metric** | **Description** | **Value** |
| --- | --- | --- |
| **MSE (Mean Squared Error)** | Measures pixel intensity difference | 2.13 |
| **SSIM (Structural Similarity Index)** | Measures perceptual similarity | 0.8745 |

A lower **MSE** and a higher **SSIM** indicate better similarity between the input and output images.

**4. Results**

* The generated mosaic visually **preserves the structure and color** of the input image.
* **Higher grid sizes (e.g., 64×64) improve detail accuracy** but increase computational cost.
* Using **more cat images in the dataset** improves tile selection and overall appearance.

**5. Conclusion & Future Improvements**

This project successfully converts any input image into a **cat-image-based mosaic** using clustering and similarity matching techniques. Future improvements could include:

* **Adding more diverse tile images** (e.g., dogs, abstract textures).
* **Enhancing performance with deep learning models** instead of clustering.
* **Allowing users to choose tile themes** dynamically.

The project is live and can be accessed via **Hugging Face Spaces**.

<https://huggingface.co/spaces/dsnahil/Interactive_Image_Mosaic_Generator>