

Report

live demo: https://huggingface.co/spaces/Novadecore/mosaic_generator

This project implements an adaptive image mosaic generator using multi-scale tiles and per-cell blending.

1. Preprocessing

There are two types of input images: the background image that will be divided into a grid and blurred, and the tile images that make up the mosaic.

The background image is cropped to a fixed resolution of 512×512 using the **resize_with_crop** method. It is then resized with OpenCV's `cv2.resize` and quantized into 16 representative colors using the `color_quantization` method, which applies k-means clustering on RGB pixel values.

For the tile images, a directory named `tiles_folder` contains the prepared tile collection. These images are loaded in the class initializer via **_load_tiles**, cropped into squares, and resized to 8, 16, or 32 pixels for further processing.

2. Image Gridding and Thresholding

The **grid_mean_colors** method performs adaptive gridding with block sizes 32×32 -> 16×16 -> 8×8 based on texture variance.

Processing begins with 32×32 blocks. If a block's mean per-channel variance is below 80.0 and the block is full-sized, the block is retained as 32×32.

Otherwise, the block is subdivided into 16×16 cells. A 16×16 block is retained if its variance is below 60.0 and it is full-sized.

Any remaining areas are further divided into 8×8 cells, which are always accepted without variance checking.

For every accepted block, the mean RGB color is calculated and stored. In parallel, all candidate tile images from `tiles_folder` are resized into three size banks (32, 16, 8), and each resized tile's average color is precomputed and cached for later matching.

3. Tile Mapping

For each adaptive block, the **build_mosaic** method selects the best-matching tile from the corresponding size bank (32, 16, 8). The match is determined by computing the Euclidean distance in RGB space between the block's mean color and the precomputed tile colors. The chosen tile, already resized to the correct dimensions, is placed into the mosaic at the block's location. This process produces a multi-resolution layout in which larger tiles represent smoother regions and smaller tiles capture areas with greater detail. To restore fine local detail, per-cell alpha blending is applied between the mosaic and the quantized reference image via **blend_mosaic_with_ref**. The blending factor `alpha` is adjustable by the user, allowing control over the balance between stylized mosaic effects and fidelity to the source image.

4. Gradio Interface

To enable interactive experimentation, a Gradio interface was implemented for the mosaic generation pipeline. The interface allows users to upload an input image, adjust the blending factor via a slider ranging from 0.0 to 0.8, and generate the corresponding mosaic. The pipeline is executed step by step: image resizing and preprocessing, color quantization, adaptive gridding, tile mapping, and optional per-cell blending with the quantized reference image. The final output mosaic is displayed alongside the original input, while quantitative performance metrics (MSE and SSIM) are computed and presented in real time. Example images are also provided within the interface, allowing users to quickly test the system and compare the impact of different blending values.

5. Performance Metric

The `evaluate_performance` method quantitatively assesses the quality of the generated mosaic relative to the original image. Both images are first verified to have identical dimensions. Two complementary metrics are then computed:

Mean Squared Error (MSE): Measures the average squared difference between pixel values of the original and mosaic images. Lower values indicate closer similarity.

Structural Similarity Index (SSIM): Captures perceptual similarity by comparing luminance, contrast, and structural information across the RGB channels. Values range from -1 to 1 , with scores closer to 1 reflecting higher visual fidelity.

The method returns the pair `(mse_val, ssim_val)`, providing both a low-level error measure and a perceptually informed similarity score.

Result

Example	Blend Factor (α)	MSE	SSIM	Observation
1	0.0	1480.94	0.1717	Pure mosaic rendering: large pixel-wise error and low structural similarity.
2	0.3	1467.18	0.1438	Slight blending reduces MSE marginally, but SSIM decreases due to limited structural recovery.
3	0.6	489.37	0.4273	Strong blending significantly lowers MSE and improves SSIM, reintroducing more fine detail and structural fidelity.

The results indicate that blending has a substantial impact on reconstruction quality. At $\alpha = 0.0$, the output relies solely on tile substitution, leading to high pixel error and poor structural similarity. A modest blend ($\alpha = 0.3$) yields little improvement in MSE and even reduces SSIM, suggesting insufficient recovery of local details. By contrast, a stronger blend ($\alpha = 0.6$) dramatically decreases MSE and increases SSIM, confirming that per-cell blending effectively restores fine-grained information. However, excessively high blending may diminish the intended mosaic stylization, so an appropriate balance must be selected depending on whether visual fidelity or artistic effect is prioritized.