A Modular and Tunable Image Mosaic Reconstruction System Based on Visual Feature Matching

Qing Zhang
Department of Mathematics
University of Michigan
Ann Arbor, MI, USA
zqingrr@umich.edu
Github url: https://github.com/Titus-Z/Stats507-Final-Project.git

***Abstract*—This paper presents a modular and tunable system for image mosaic reconstruction using visual feature matching. Traditional mosaicking methods primarily rely on color averages, which limits the structural fidelity of the output. To address this, our system incorporates a variety of handcrafted and learned visual features, including color histograms, HOG, edge density, and shallow CNN responses. It supports flexible feature combinations, FAISS-based fast nearest-neighbor search, and quantitative evaluation using PSNR, SSIM, and LPIPS. We also implement an automatic parameter tuning module to explore optimal feature weights. Experiments demonstrate that hybrid features can significantly improve visual quality while balancing computational cost.**

***Index Terms*—Image mosaic, visual similarity, shallow CNN, FAISS, feature fusion, LPIPS**

I. Introduction

Image mosaic is a technique that reconstructs a target image by dividing it into small patches (tiles) and replacing each tile with a visually similar image selected from a gallery. This process combines the technical rigor of computer vision with the aesthetic appeal of artistic image composition. It has been widely applied in creative visual design, digital exhibitions, and data-driven visualizations.

However, most existing mosaic generation methods rely on simple RGB color averages as the primary matching criterion. In such methods, each tile in the target image is replaced by a gallery image whose mean color most closely matches that of the tile. While computationally efficient, this approach often fails to preserve the structural integrity and perceptual fidelity of the original image.

To address these limitations, this work proposes a modular and tunable image mosaic reconstruction system. Our goal is to explore how different visual features impact the quality of reconstruction, and to introduce a scoring-driven tuning framework that automatically optimizes feature combinations. The system supports multiple handcrafted and learned features, including color histograms, edge maps, HOG descriptors, and shallow CNN responses.

Considering practical limitations in computing and storage, we selected a subset of approximately 50,000 images from the Tiny-ImageNet dataset to serve as the gallery pool. The final objective is to construct a complete image mosaic system that can generate visually coherent, structurally faithful mosaics efficiently, with built-in support for automated parameter tuning.

II. System Overview and Modules

The proposed image mosaic system is composed of three core modules: feature extraction, visual matching, and reconstruction evaluation.

A. Feature Extraction Module

The system supports six types of image features, which can be linearly combined with user-defined weights:

- **RGB Mean**: Channel-wise average RGB values; simple and efficient.
- **HSV Mean**: More aligned with human color perception.
- **HOG**: Histogram of Oriented Gradients; captures local edge and shape structure.
- **Color Histogram**: Describes color distribution; efficient and useful for style matching.
- **Edge Density**: Measures the proportion of edge pixels using the Canny operator.
- **Shallow CNN Feature**: Extracted from the first two convolutional layers of VGG16.

RGB and HSV features provide basic color-level information and are mainly used as auxiliary components in fusion. HOG captures directional patterns and local structure; color histograms reflect tone distribution without spatial encoding; edge density gives a global structure cue.

CNN features are extracted from early VGG16 layers to capture texture and edge information. However, the raw feature size is extremely high (e.g., 262,144 dimensions), which is computationally costly. To address this, global average pooling (GAP) is applied to reduce the dimensionality (e.g., down to 64D) while preserving meaningful texture activations.

B. Matching Module

To balance matching accuracy and computational efficiency, three strategies are implemented:

- **Brute-force Matching**: Computes distances to all gallery images; accurate but slow.
- **KMeans Clustered Matching**: Clusters gallery features and searches within clusters.
- **FAISS-based ANN Matching**: Uses Facebook's FAISS library for efficient high-dimensional search (default method).

KMeans enables local search within visually similar clusters, reducing computation for moderately sized galleries. FAISS (Facebook AI Similarity Search) is an industrial-grade library designed for scalable approximate nearest-neighbor (ANN) search, supporting inner-product or cosine-based similarity. It is critical for large-scale feature retrieval in this project.

C. Reconstruction and Evaluation Module

Tiles are replaced based on the nearest-match gallery image, and reconstructed into a full mosaic. We adopt three automated evaluation metrics to assess visual quality:

- **PSNR (Peak Signal-to-Noise Ratio)**: Measures pixel-wise fidelity.
- **SSIM (Structural Similarity Index)**: Evaluates perceived structure and contrast.
- **LPIPS (Learned Perceptual Image Patch Similarity)**: Computes perceptual distance using CNN features.

All reconstructed images are saved for visual inspection. While numerical scores provide consistent benchmarks, we observe that subjective perception still plays an important role. Bridging automated metrics with human visual judgment remains an open direction for future work.

a) Evaluation Formulas.: **PSNR** is computed as:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX^2}{\text{MSE}} \right)$$

where MSE is the mean squared error between original and reconstructed image pixels, and $MAX = 255$ for 8-bit RGB images.

SSIM between two image patches x and y is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where μ_x, μ_y are means, σ_x^2, σ_y^2 are variances, and σ_{xy} is the cross-covariance. C_1 and C_2 are small constants to stabilize the division.

LPIPS is computed using deep feature maps extracted from a pretrained CNN (e.g., AlexNet). Let $f_l(x)$ and $f_l(y)$ denote normalized features of images x and y at layer l, and w_l be learned weights:

$$\text{LPIPS}(x, y) = \sum_l w_l \| f_l(x) - f_l(y) \|_2^2$$

This combination of metrics enables both pixel-level, structural, and perceptual evaluation of the reconstructed mosaic.

III. Model Tuning and Automated Search

To explore the optimal combination of features, we develop an automatic tuning framework:

- Users specify search ranges for feature weights.
- The system generates mosaics for all weight combinations and evaluates their quality using SSIM, LPIPS, and PSNR.
- All output images and scores are logged for inspection.
- The best-performing configuration is returned for reuse.

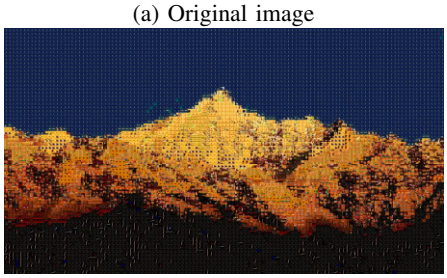
This module enables reproducible, quantitative performance optimization and significantly reduces manual trial-and-error efforts.

IV. Experiments and Results

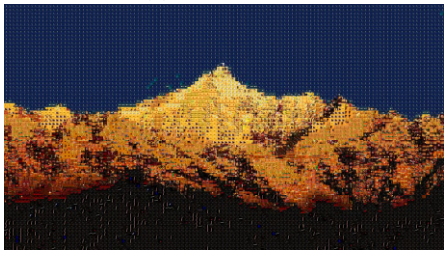
A. Example result

We compared the FAISS-based matching method with KMeans in our example notebook. The results showed that FAISS achieved comparable visual quality while being over 100 times faster than KMeans. Based on this, we adopted FAISS as the default matching strategy in the pipeline.

(a) Original image



(b) FAISS result



(c) KMeans result

Fig. 1: Visual comparison of matching methods: FAISS vs. KMeans.

B. Experiment result

To evaluate the effectiveness of different feature combinations, we conduct a set of experiments using a high-resolution

target image and approximately 50,000 resized gallery images from the Tiny-ImageNet dataset.

We compare seven feature weight configurations, including single-feature baselines and multi-feature fusion strategies:

Run	Weights	PSNR	SSIM	LPIPS	Time (s)
ex_01	w_cnn: 1.0	17.60	0.4227	0.2656	142.25
ex_02	w_color: 1.0	17.23	0.4117	0.2580	42.75
ex_06	w_hog: 0.1, w_color: 0.9	17.26	0.4076	0.2563	93.52
ex_04	w_edge: 0.3, w_color: 0.7	18.79	0.4072	0.2494	54.82
ex_00	w_color: 0.6, w_hog: 0.3, w_edge: 0.1	17.50	0.3999	0.2898	106.73
ex_05	w_cnn: 0.5, w_color: 0.5	17.66	0.3912	0.2249	197.12
ex_03	w_hog: 0.5, w_color: 0.5	16.88	0.3856	0.3623	89.01

TABLE I: Performance comparison of different experiments with their corresponding feature weights.

From Table I, we conclude the following:

- The pure CNN feature (ex_01) achieves the highest SSIM score (0.4227), indicating strong structure preservation, but it requires significantly more time (142.25s) for processing.
- The hybrid combination (ex_00), integrating color histogram, HOG, and edge density, achieves a good trade-off between quality (SSIM 0.3999) and runtime (106.73s), making it a practical default choice.
- Pure color features (ex_02) and lightweight fusions (ex_06) also deliver decent SSIM scores (0.41) while maintaining fast runtimes, which are suitable for efficiency-oriented scenarios.
- Combining CNN and color (ex_05) offers acceptable quality (SSIM 0.3912), but the computation time (197.12s) is the highest among all trials.
- Feature sets relying solely on HOG or edge (ex_03, ex_04) show lower performance and limited reliability, suggesting they are insufficient when used alone.

C. Model tuning

To enhance the quality of mosaic reconstruction, we explore a lightweight model tuning strategy based on feature weight optimization. Specifically, we adjust the combination weights of different visual features (e.g., CNN, color histogram, HOG, edge density) to search for configurations that yield the best visual fidelity for a given target image.

Initially, we planned to implement a full grid search across a wide parameter space. However, due to limitations in hardware performance, runtime, and storage, we opted for evaluating a smaller set of representative feature weight presets. This practical compromise still allows us to demonstrate the tunability and adaptability of our system on different target images. In the future, the tuning module can be extended to support full grid search or more advanced automated search methods such as Bayesian optimization or reinforcement learning.

Summary

Through a series of experiments, we demonstrate the effectiveness and flexibility of our mosaic reconstruction system. FAISS-based matching achieves comparable quality to KMeans while being significantly faster, making it our default choice. Among different feature configurations, CNN-based methods yield the highest visual similarity but are computationally expensive. Lightweight combinations such as color + HOG + edge provide a good balance between quality and speed. Finally, our tuning module, though currently limited to a few presets, showcases the potential of optimizing feature weights for different target images and paves the way for more advanced search strategies in future work.

V. Conclusion and Future Work

In this paper, we present a modular and tunable image mosaic system that leverages multiple types of visual features for patch matching. Unlike traditional methods that rely solely on color averages, our approach supports feature fusion and automated quality evaluation using SSIM, LPIPS, and PSNR.

Through a series of experiments with diverse feature combinations, we demonstrate that shallow CNN features offer the best reconstruction quality, while hybrid combinations such as color+HOG+edge achieve a strong balance between visual fidelity and efficiency. The inclusion of an automatic parameter tuning module further improves usability and repeatability of the pipeline.

Despite its effectiveness, the current system has limitations. Feature extraction remains computationally intensive for large-scale galleries, and the optimal weight setting may still vary across different target images.

In future work, we plan to explore adaptive tuning strategies that dynamically adjust feature weights based on image content. In addition, incorporating semantic-level features or using lightweight CNNs may further enhance both performance and efficiency.

VI. Reference

References

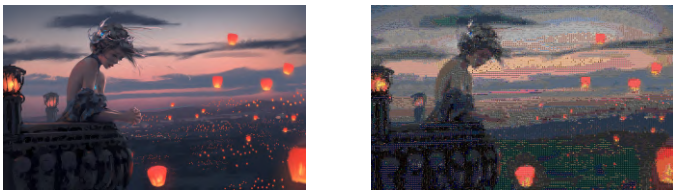
[1] Hugging Face, *Image Similarity with Transformers*, Hugging Face Blog, 2023. Available: https://huggingface.co/blog/image-similarity

[2] Facebook AI Research (FAIR), *FAISS: A library for efficient similarity search and clustering of dense vectors*, GitHub Repository. Available: https://github.com/facebookresearch/faiss

APPENDIX

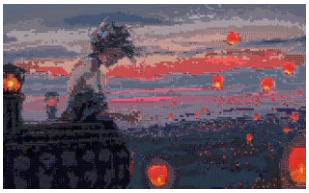
A. Experiment Visual Results

This appendix presents visual outputs from our tuning experiments (`experiment_00` to `experiment_06`). Each result is accompanied by its corresponding feature weight configuration and evaluation metrics. The original target image is included for comparison.

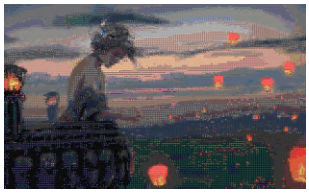


Original Image

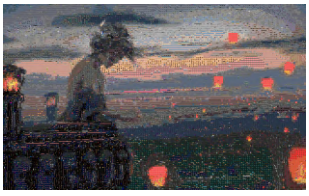
ex_00



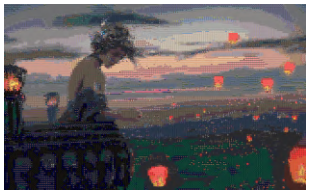
ex_01



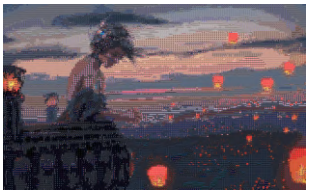
ex_02



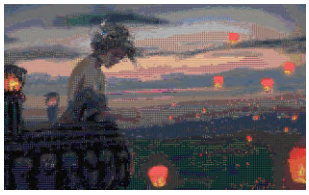
ex_03



ex_04



ex_05



ex_06

Fig. 2: Visual comparison of the original image and mosaic results from `experiment_00` to `experiment_06`.

B. File Organization

- **mosaic_pipeline.py**: Core pipeline with all main modules (feature extraction, fusion, tiling, matching, reconstruction).
- **evaluator.py**: Computes PSNR, SSIM, and LPIPS scores.
- **image_prepare.ipynb**: Preprocesses and resizes 50,000 Tiny-ImageNet images.
- **Full pipeline example.ipynb**: Runs inference using the best manually selected configuration.
- **Experiment.ipynb**: Batch testing of various feature combinations.
- **Model tuning.ipynb**: Performs model tuning for a single image.