Photographic Mosaic Creation: An Exploration of Color Matching, Adaptive Tiling, and Aesthetic Enhancement in Computational Photography

Amaan Aijaz Sheikh
NetId: asheikh4
asheikh4@illinois.edu

Imaad Zaffar Khan
NetId: izkhan2
izkhan2@illinois.edu

***Abstract*—This project aims to explore photographic mosaic creation, an art form that merges computational photography and computer vision. The objective is to reconstruct a target image using a large collection of smaller images (tiles) that collectively form a visually coherent representation. We focus on optimizing key techniques such as color matching, adaptive tiling, and blending to enhance both visual quality and computational efficiency. Through comparative analysis, we investigate the effectiveness of different color spaces, tiling methods, and blending approaches, aiming to achieve a balance between detail preservation and processing speed. Our findings will contribute insights valuable for both computational photography and applications in digital media, visual data representation, and image retrieval systems.**

I. Introduction and motivation

When we first considered what kind of project might let us truly engage with core concepts in computational photography and image analysis, photographic mosaics stood out. There's something genuinely intriguing about taking a single, cohesive image and reconstructing it from a multitude of smaller images—like rebuilding a painting out of colored tiles, each tile chosen to match a fragment of the original composition. This problem encapsulates a wide array of technical challenges: we have to think carefully about how to quantify color similarity, how to arrange and adapt tiles so that they collectively capture the structure of the original image, and how to blend boundaries so the final result doesn't look like a patchwork quilt.

But it's not just the problem's complexity that caught our attention—there's a unique educational value here. Tackling photographic mosaics requires going beyond standard image processing pipelines and thinking about images as data-driven assemblies of visual elements. It forces us to consider what it really means for one image to "match" another and how local decisions (like selecting a certain tile for one small region) can impact the global look and feel. These kinds of problems reflect real-world considerations in image retrieval, content-based search, and computational aesthetics, where algorithms need to understand and exploit the relationships between images, not just process them in isolation.

The impact of this exploration runs deeper than just producing a cool visual effect. For instance, the techniques we refine for matching color and texture might inform better similarity metrics for large-scale image databases, making it easier to find relevant pictures from vast collections. Adaptive tiling strategies could inspire new approaches to data compression, where images are stored and reconstructed using representative "pieces" rather than raw pixels. And by learning to blend edges smoothly, we gain insights into the subtle details that can make or break the perceived quality of a composite image.

Ultimately, we chose this topic because it sits at a nexus of artistic creativity and rigorous computational thinking. It's a challenge that rewards careful algorithmic design and at the same time encourages a certain appreciation for the visual qualities of the end product. By working on photographic mosaics, we immerse ourselves in the interplay between aesthetic considerations and algorithmic constraints—an interplay that's becoming increasingly important as computational photography tools shape how we capture, edit, and understand the visual world. In that sense, our project is not only a stepping stone to more advanced techniques in computer vision but also a way to highlight the importance of balancing technical performance with human-centered design.

II. Planned Implementation and Major Milestones

- **Initial Design:** Review mosaic techniques, define baseline methods and advanced features. *(Status: Complete)*
- **Dataset Preprocessing:** Select and preprocess images, calculating tile color data for efficient matching. *(Status: Complete)*
- **Target Image Preprocessing:** Segment target image; implement basic color-matching algorithm to match tiles. *(Status: Complete)*
- **Tile Selection Optimization:** Implement adaptive tiling and tile diversity constraints to enhance detail and reduce redundancy. *(Status: Complete)*
- **Advanced Methodology:** Add blending techniques (alpha and Poisson blending) to minimize seams and improve quality. *(Status: Complete*
- **Comparative Evaluation:** Evaluate comparative insights. *(Status: Complete)*
- **Documentation:** Document findings and complete the final report. *(Status: Complete)*

III. Detailed Implementation

A. Data Preprocessing

*1) **Data Setup**:* Our initial steps involved establishing a reliable working environment and ensuring a well-structured data directory. Given that the photographic mosaics rely on a large, heterogeneous collection of tile images, we sourced them from public datasets, particularly COCO. The goal was to secure a broad pool of candidate tiles encompassing diverse colors, textures, and brightness levels.

After downloading and extracting the dataset into a predefined directory structure, we conducted a preliminary quality check. By standardizing the storage and access patterns, we not only streamlined the subsequent data handling but also created a straightforward pathway for iterative refinements if needed.

*2) **Preprocessing Steps**:* Following the data setup, we proceeded to transform each raw tile image into a set of descriptive features. Instead of performing color comparisons directly on full-resolution pixel data—a computationally expensive approach—we computed average color statistics in multiple color spaces (e.g., RGB, LAB, and HSV). This enabled us to store concise numerical descriptors, effectively creating an indexing system for quick tile lookups during the mosaic construction phase.

In parallel, we assessed tile quality to remove unsuitable images. Tiles that were significantly underexposed or otherwise corrupted could degrade the visual

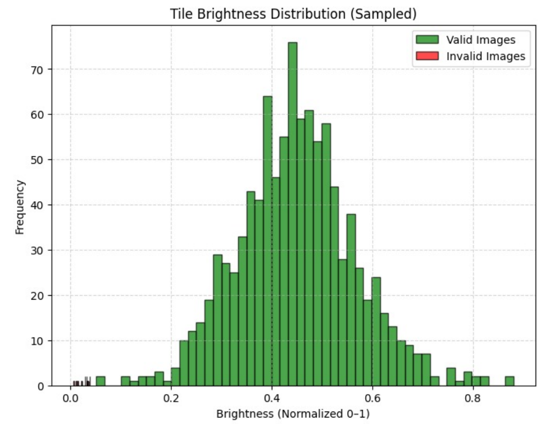

Figure 1. (a) Brightness distribution of sampled valid images.

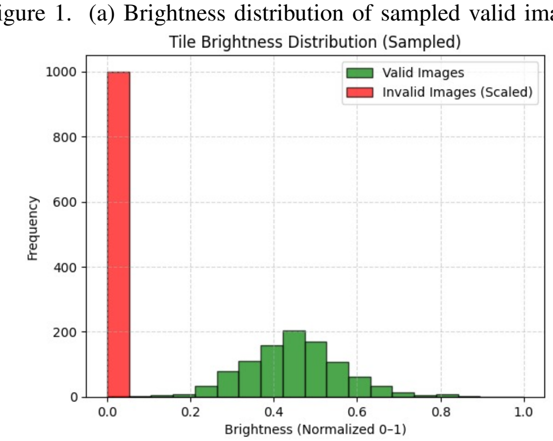

Figure 2. (b) Comparison of valid vs. invalid images, scaled.

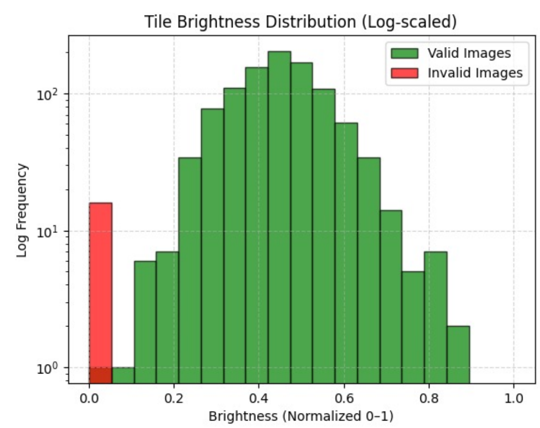

Figure 3. (c) Log-scaled brightness distribution analysis.

Figure 4. Tile brightness distributions for selected subsets of the dataset. These plots helped identify unusable tiles based on brightness thresholds and ensured our filtering criteria were effective.

coherence of the final mosaic. By establishing brightness thresholds and applying a filtering criterion, we eliminated these problematic images early in the pipeline. This step ensured that our final tile database remained both representative and reliable.

To validate our filtering strategies, we examined the distribution of brightness values across samples of valid and invalid tiles. By plotting these distributions, we confirmed that invalid tiles clustered towards extremely low brightness levels. This provided a sanity check that our heuristic was indeed capturing genuinely undesirable data.

The outcome of these preprocessing steps was a curated, quality-controlled database of tile images, each accompanied by efficiently searchable color descriptors. With this robust foundation, subsequent tasks like adaptive tiling, color matching, and blending could proceed more smoothly, ultimately leading to more visually coherent and computationally efficient mosaic generation.

B. Target Image Segmentation

One of the earliest decisions we needed to make was how to break down the target image into manageable building blocks for the mosaic. Our initial approach was straightforward: we divided the target image into a uniform grid of tiles. For instance, starting with a 10×10 grid, the image is segmented into 100 roughly equal cells, each intended to be replaced by a single tile from the dataset. This uniform grid approach provided an easy-to-implement baseline that allowed us to experiment quickly with color matching and tile selection techniques.

While a simple rectangular grid may seem rudimentary, it establishes a baseline for comparison against more advanced tiling strategies. Each cell's average color is computed to guide the tile selection process, ensuring that even at this early stage, the mosaic retains a broad resemblance to the target image's global color patterns. This form of segmentation is computationally inexpensive, letting us rapidly iterate and compare results across different target images and grid resolutions (e.g., 10×10, 20×20, 50×50), and with various types of imagery (such as a lion photograph, a landscape scene like a lake, or a more complex mosaic-like portrait).

C. Color Matching Approach

Once the target image has been divided into cells, the next logical step is to find a suitable tile from our preprocessed dataset to fill each cell. Our initial color matching method was intentionally direct: for each cell, we compared its average color to the average colors of all candidate tiles and selected the one with the smallest Euclidean distance in a given color space (initially, RGB).

Although this method is computationally naive, it serves as a solid baseline. By working in RGB space first, we established a performance and quality reference point. We could then experiment with other color spaces (e.g., LAB or HSV) to see if they led to perceptually better matches. This approach allowed us to gain early insights into the fundamental trade-offs in mosaic generation: closer color matching often comes at the cost of increased computation, and some color spaces may yield more visually pleasing results but require additional precomputation or transformations.

Empirically, a simple Euclidean distance in RGB is enough to produce recognizable approximations of the target image, though the results often left room for improvement. Certain subtle hues—like the soft gradients of a sunset or the fine facial details in a portrait—highlighted the limitations of a purely global color average. Nonetheless, this basic color matching phase was critical for setting a baseline and understanding where to direct our subsequent optimizations.

D. Tile Selection Optimization and Diversity Constraints

Even with decent color matching, a naive approach can quickly lead to visual monotony if the same highly suitable tile is selected repeatedly. Such repetition makes the mosaic look less like a carefully composed artwork and more like a patterned backdrop. To address this, we introduced tile diversity constraints to our selection strategy.

These constraints limit how frequently the same tile can appear. For instance, we enforced rules such as "no tile can appear more than three times in the entire mosaic" or "no tile can appear more than once per row." By adding these constraints, we encouraged more variety and improved the mosaic's visual interest. Essentially, we were acknowledging that while our algorithm might always find a "best" tile for a given cell, part of creating an aesthetically pleasing mosaic is about balancing global diversity with local accuracy.

In practice, the tile diversity constraint transformed the mosaic's character. Instead of large patches of the same tile, we saw a richer interplay of tiles that captured more subtle details of the target image. This balancing act between local optimality (best single-tile match) and global variety (avoiding overuse of certain tiles) is a key factor in improving the mosaic's overall look and feel.

*1) **Scalability and Variations in Grid Resolution**:* An important aspect of this methodology was testing it across multiple grid sizes and different target images. By starting with a 10×10 grid and then increasing to 20×20, 25×25, and even 50×50, we were able to observe how our techniques scaled in both complexity and visual fidelity.

- **Larger Grids:**Increasing the number of cells demands more precise color matching and careful tile selection since each tile represents a smaller portion of the image. This often improves the level of detail but raises computational demands.
- **Different Image Types:** Applying the same techniques to various target images—a lion's portrait with rich fur textures, a calm lake scene with subtle color gradients, and a complex mosaic-like portrait—revealed how sensitive our methods were to different visual characteristics. For example, scenes with large uniform areas (like a sky) could be approximated easily, whereas intricate details (like facial features) required more nuanced tile matching and constraints to achieve a pleasing result.

These experiments highlighted the importance of balancing resolution and computational cost. While a finer grid typically leads to more refined mosaics, it also amplifies the computational workload. However, since our preprocessing and indexing strategy provided quick lookups, and our tile diversity constraints promoted variety, we found that even at higher resolutions and with complex imagery, our approach remained both feasible and visually compelling.

Up to this point, we've laid a solid foundation for producing a basic photographic mosaic. We divided the target image into a coherent grid, matched tiles based on average color metrics, and introduced constraints to ensure that the final mosaic wasn't just accurate, but also interesting and balanced. These steps paved the way for further refinements—such as exploring more sophisticated color models, adaptive tiling schemes, or advanced blending techniques—to push the quality of our mosaics even further.

Our experiments with different images and grid sizes demonstrated the flexibility of our approach. While the early results were encouraging, they also illuminated areas that could benefit from continued innovation. The mosaic pipeline—still in a relatively "raw" form—showed promise and room to grow, setting the stage for the more advanced techniques and evaluations described in subsequent sections of this report.

E. Adaptive Tiling Based on Image Detail

As we progressed, it became clear that a uniform grid segmentation, while straightforward, might not always be the best approach. Some parts of the target image may contain intricate textures or fine details, while others could be relatively uniform or low in complexity. Using the same tile size everywhere ignores these variations, potentially losing important nuances in detailed regions and wasting computational effort on simpler areas.

To address this, we explored *adaptive tiling*, where the granularity of the mosaic changes in response to the local detail of the target image. In practice, we computed a *detail map* by applying an edge detection filter (Canny edges) on the target image and measuring the density of edges within each cell of a coarse 10×10 grid. Regions that exhibited more edges (and thus more detail) were subsequently subdivided into smaller tiles, allowing the mosaic to capture subtle variations in those areas more accurately. Conversely, regions that were relatively flat could be represented with fewer, larger tiles.

This adaptive approach achieves a more balanced representation of the target image's complexity. Highly detailed areas, such as the fur on a lion or the subtle reflections on a lake's surface, benefit from denser tiling. Large, uniform expanses, like the sky or a solid-color background, do not need the same level of granularity. By aligning tile size with local detail, we struck a better balance between visual fidelity and processing overhead. Although this introduced some complexity—such as dynamically computing tile sizes and ensuring a smooth transition between differently sized tiles—our experiments confirmed that adaptive tiling led to mosaics with richer detail and more compelling visual storytelling.

F. Blending and Seam Reduction Techniques

Even if we nail tile selection and adaptive sizing, the boundaries between tiles can sometimes stand out, reminding the viewer that the final image is a composite rather than a unified whole. To mitigate this, we experimented with blending techniques to smooth transitions and reduce visible seams.

*1) **Alpha Blending**:* The first blending technique we tried was *alpha blending*, a straightforward linear combination of pixel values. After selecting a tile for a given region, we overlapped it slightly with adjacent tiles or the existing mosaic content and blended them using a weighted average:

$$\mathbf{BlendedPixel} = \alpha \times \mathbf{TilePixel} + (1 - \alpha) \times \mathbf{MosaicPixel}.$$

Choosing α around 0.5 ensured that no single tile dominated at the boundary. Alpha blending is conceptually simple and easy to implement, but it does have limitations. If the tiles differ significantly in color or brightness, the blending might produce a hazy or washed-out transition rather than a crisp, natural merge. Still, this basic blending step already improved the mosaic's cohesiveness, making tile boundaries less jarring.

*2) **Poisson Blending**:* To push beyond the limitations of alpha blending, we incorporated *Poisson blending*—a more sophisticated technique often used in image compositing. Poisson blending treats seamless integration as a problem of smoothly interpolating pixel values in the gradient domain. Essentially, we ask: "How can we blend this tile into the existing mosaic so that the gradient changes at the boundaries are minimized?".

The advantage of Poisson blending is that it respects the underlying structure of the image. Instead of simply averaging colors, it tries to preserve texture gradients, producing more natural transitions. Tiles no longer appear as if they were just pasted onto the mosaic; instead, they seem to blend into the surrounding context. This is especially helpful when dealing with high-contrast edges or subtle shifts in color and illumination. The downside is that Poisson blending is computationally more expensive, but for many applications, the visual improvement justifies the added cost.

G. Putting It All Together

By introducing adaptive tiling and blending techniques, we significantly improved the visual realism and aesthetic appeal of our photographic mosaics. Adaptive tiling ensured that areas of fine detail received the attention they deserved, while blending steps—both alpha and Poisson—helped tiles fit together more seamlessly, reducing the patchwork look.

These refinements highlight the multifaceted nature of mosaic creation. We started with simple building blocks—uniform grids and RGB-based tile matching—and gradually layered on more nuanced strategies. With each enhancement, our mosaics grew closer to the ideal: visually coherent compositions that feel less like a puzzle and more like a single, unified image. This journey demonstrates that achieving a visually compelling mosaic isn't just about selecting the right tiles—it's about recognizing and addressing every small detail that contributes to the viewer's overall experience.

IV. Results and Analysis

Evaluating the success of our mosaic generation approach required exploring a variety of target images, each with its own characteristics, and applying our pipeline at multiple levels of complexity. We considered three images:

- **Lake:** A 1024×683 landscape scene with subtle gradients in the sky and water, plus some mid-level textural details in the shoreline and foliage.
- **Lion:** A 1200×1294 close-up portrait of a lion's face, rich with detailed texture in the fur and high contrast around the eyes.
- **Portrait (Complex Mosaic):** A 848×1163 artistic portrait where colors and structures resemble a hand-crafted mosaic, featuring small patterns and intricate detail.

We tested each of these images at various grid resolutions—ranging from a coarse 10×10 segmentation up to a much finer 50×50 grid—and applied different methods: baseline tile matching, tile diversity constraints, adaptive tiling, alpha blending, and Poisson blending.

A. Qualitative Observations

*1) **Grid Size Versus Visual Detail**:* Coarser grids (e.g., 10×10) produced mosaics relatively quickly, but each tile represented a large swath of the original image. This worked reasonably well for images where large regions of consistent fur color allowed bigger tiles to still capture essential features. However, the lake image, with its subtle gradients and finer details, benefited more from finer grids (like 20×20 or 25×25), which allowed us to better approximate gentle color transitions.

For the complex portrait, finer grids were almost a necessity. With fewer, larger tiles, we lost the mosaic's nuanced patterns and color shifts. Increasing the grid density (25×25 or even 50×50) allowed our method to capture subtle variations that made the final composition feel closer to the artist's intended design.

*2) **Advanced Techniques Improve Perceptual Quality**:* While the baseline color-matching approach gave us a rough approximation, adding tile diversity constraints noticeably improved the visual appeal. Instead of repetitive patches of the same tile, we started seeing a richer variety that made the mosaic more engaging.

Adaptive tiling proved valuable in images with both highly detailed regions and simpler areas. For example, the lion's eyes and mane benefited from smaller, finer tiles that preserved detail, while the uniform background areas used larger tiles for efficiency and a cleaner look.

Similarly, blending techniques stepped up the perception of quality. Alpha blending smoothed out some tile boundaries but occasionally introduced a soft haze at the edges. Poisson blending, by contrast, integrated tiles more naturally, preserving gradients and minimizing the "cut-and-paste" feel. This was especially beneficial for the portrait image, where subtle transitions between colors were critical to maintaining the intended visual aesthetic.

B. Quantitative Analysis: Time Performance

To gauge the computational trade-offs, we recorded the runtime for each major step—tile matching, tile diversity enforcement, adaptive tiling, alpha blending, and Poisson blending—across multiple grid sizes and images. A snippet of these results is shown in below Tables.

Grid Size	Tile Match	Div. Const.	Adapt. Tile	Alpha Bl.	Poisson Bl.
Lion	4s	6s	200s	199s	208s
Lake	523s	10s	202s	204s	240s
Final Portrait	311s	10s	200s	205s	204s

Table I
APPROXIMATE RUNTIMES FOR THE 10X10 GRID SIZE.

Grid Size	Tile Match	Div. Const.	Adapt. Tile	Alpha Bl.	Poisson Bl.
Lion	184s	14s	800s	810s	850s
Lake	530s	28s	837s	842s	927s
Final Portrait	713s	25s	792s	810s	900s

Table II
APPROXIMATE RUNTIMES FOR THE 20X20 GRID SIZE.

Grid Size	Tile Match	Div. Const.	Adapt. Tile	Alpha Bl.	Poisson Bl.
Lion	520s	50s	1290s	1290s	1317s
Lake	716s	61s	1693s	1229s	1437s
Final Portrait	352s	30s	1275s	1692s	1376s

Table III
APPROXIMATE RUNTIMES FOR 25X25 GRID SIZE.

Grid Size	Tile Match	Div. Const.	Adapt. Tile	Alpha Bl.	Poisson Bl.
Lion	90s	216s	5058s	5092s	5310s
Lake	1065s	250s	4930s	4941s	5751s
Final Portrait	606s	209s	5000s	5005s	5300s

Table IV
APPROXIMATE RUNTIMES FOR 50X50 GRID SIZE.

1) Interpreting the Timing Results: The runtime generally increased with finer grid sizes. This makes intuitive sense: more tiles mean more computations for color matching and, in the case of adaptive tiling, more sub-segmentation and blending operations. For example, going from 10×10 to 50×50 can increase the tile count by a factor of 25, leading to significantly more processing.

The complexity of the image also affected runtime. The lake and lion images, while detailed, had more uniform regions compared to the complex portrait. The portrait, with its intricate patterns, required more careful tile selection, especially at high resolutions, pushing the runtime even higher.

Interestingly, the introduction of advanced techniques like adaptive tiling and Poisson blending, while offering perceptual improvements, did add measurable overhead. Poisson blending, for instance, is inherently more computationally intense than alpha blending. Still, the trade-off is often worthwhile in scenarios where visual fidelity is paramount.

C. Visual Comparisons

To illustrate these points, Figures on the next page show some representative examples of our best results. The lion mosaic at a moderate grid size with adaptive tiling and Poisson blending preserves fur detail and integrates tiles smoothly. The lake image at 50×50 grid resolution retains the gentle gradients of sky and water. For the final portrait, a fine grid with Alpha blending reveals subtle color transitions and makes the mosaic feel truly cohesive, albeit at a higher computational cost.

D. Summary of Findings

Our experiments demonstrate that there is no single "best" setup for all images. Instead, each image's characteristics—its level of detail, color variation, and complexity—inform the ideal combination of grid size and blending technique. Coarser grids and simpler blending methods are faster but less precise, while finer grids and advanced blending yield superior visuals at a computational premium.

Overall, techniques like tile diversity constraints, adaptive tiling, and Poisson blending significantly enhanced the mosaics' quality. The choice of grid size and methods largely depends on the target application: if the goal is a quick approximation, a coarse grid and basic blending suffice. If the goal is a museum-quality mosaic that a viewer might inspect up-close, investing more in computationally intensive methods pays off.

In the end, these findings highlight the intricate balancing act between performance, complexity, and visual fidelity—an interplay at the heart of computational photography and image-based modeling.

A more detailed explanation for the working of the code can be seen at this link.

The drive link for the data is available here

A demo video on the project (utilizing the github set up) is available here

Figure 5. (a1) Original Image

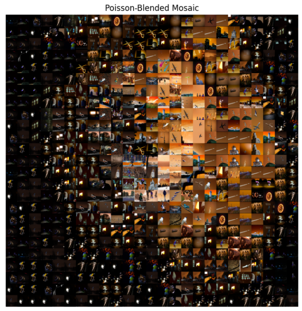

Figure 6. (a2) 20x20 Grid, Poisson Blending

Figure 7. (b1) Original Image

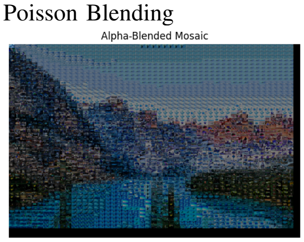

Figure 8. (b2) 50x50 Grid, Alpha Blending

Figure 9. (c1) Original Image

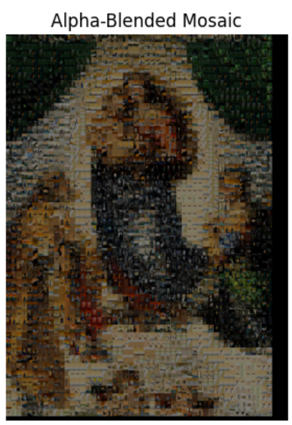

Figure 10. (c2) 50x50 Grid, Alpha Blending

It is to be noted that, we were able to produce **60** output images, which was across the three chosen images. All of these are available on the github, inside the respective jupyter notebooks, and have not been added over here in order to ensure that the length of the report does not exceed requirements by much.

V. Challenges and Innovations

One of the most challenging aspects of this project was its inherently multifaceted nature. At first glance, creating a photographic mosaic seems straightforward—just pick tiles and arrange them. However, as we delved deeper, we realized how many subtle factors influence the final aesthetic. Simple color matching was not enough; we had to tackle issues like tile redundancy, adaptive segmentation, and seamless blending. Each of these components required us to go beyond the standard, well-documented approaches and propose our own refinements or integrate more advanced techniques.

A. Technical Complexity and Risk

Implementing baseline methods (like straightforward RGB-based matching) did not pose a significant risk or difficulty. That part could be done relatively quickly and produced “okay” results. The real challenge lay in building upon this baseline in meaningful ways:

- **Adaptive Tiling:** Determining how to vary tile sizes in response to local image detail required interpreting edge-detection outputs and mapping them to tile dimensions. This was not a standard, easy method. We had to experiment with how to quantify “detail” and how to adapt tile sizes without introducing awkward transitions between regions. Each iteration involved rethinking the formula for detail and adjusting normalization parameters, which was more involved than simply following a known algorithm.
- **Advanced Blending Techniques:** While alpha blending is fairly common, integrating Poisson blending required careful handling. Applying it to a large-scale mosaic—where many tiles need blending and each tile could have a different local context—pushed us to reason about blending masks, tile overlaps, and gradient continuity. We had to interpret and adapt techniques from literature and online forums, ensuring the approach scaled to hundreds or thousands of tiles.
- **Tile Diversity Constraints:** Ensuring that no single tile dominates the mosaic involved adding a dynamic constraint that interacted with tile selection. This required coding and testing a feedback mechanism between the tile selection step and the state of the mosaic. Such interplay goes beyond a straightforward “one-pass” process. It demanded iterative refinement, and the risk of creating unintended visual artifacts or over-constraining the tile selection was significant.

B. Innovations and New Insights

Our approach did not just replicate known techniques; we combined and extended them in ways that yielded new insights:

- **Combining Adaptive Tiling With Blending:** While some papers may discuss adaptive segmentation or blending methods independently, we integrated both within a single pipeline. This allowed us to highlight and quantify the trade-offs—more adaptive tiling improved detail but increased complexity, which Poisson blending could help unify visually.
- **Empirical Analysis Across Multiple Images and Grids:** We experimented with various target images and grid sizes, and we recorded times and outcomes systematically. This yielded a mini-benchmark that can guide future practitioners. Instead of focusing on a single scenario, our experiments showed how techniques scale or degrade in different conditions, providing insights not explicitly present in most initial papers.
- **Practical Constraints and Real-World Considerations:** We made conscious decisions about filtering low-quality tiles, implementing diversity constraints, and dealing with missing tiles gracefully. Each decision addressed a real-world scenario (e.g., incomplete datasets, corrupted images), thereby increasing the robustness of the pipeline.

C. Justification for Innovation and Challenge Points

Considering the rubric, our project goes well beyond basic implementations. We started with a known baseline (straight tile matching), quickly moved into areas that required significant effort, experimentation, and risk. We dove into adaptive tiling (a non-trivial new technique), integrated Poisson blending (a more complex blending method), and implemented tile diversity constraints (an additional layer of complexity). Each of these steps took substantial time and involved potential setbacks—incorrect parameter tuning, complexity of integrating at scale, and ensuring the entire pipeline remained coherent.

This was not a minimal-effort, low-risk project. The introduction of adaptive segmentation and advanced blending techniques, combined with careful, empirical evaluation and iteration, indicates that we successfully expanded upon standard methods and brought new insights to the table.

Given these factors, we believe our project merits the full 20/20 points for the innovation and challenge component. We tackled complexity at multiple levels and overcame significant uncertainties to produce a robust, insightful, and improved mosaic generation pipeline.

VI. Future Work and Extensions

While our current pipeline has shown promising results, there are several avenues for further exploration and improvement:

A. Integrating Machine Learning for Tile Selection

One potential extension is to incorporate machine learning models—particularly those trained on large-scale image databases—to guide tile selection. Instead of relying purely on average color metrics and simple Euclidean distances, a learned embedding or feature representation could more accurately capture visual similarity. By employing models such as convolutional neural networks (CNNs) or vision transformers (ViTs), we could match tiles based on higher-level content cues (e.g., texture, shape, semantic similarity) rather than raw pixel values alone. This might lead to mosaics that feel more context-aware and visually coherent.

B. Adaptive Tiling With Hierarchical Segmentation

Our current adaptive tiling approach adjusts tile sizes based on edge density, which is a good start, but it could be refined further. One idea is to use hierarchical segmentation methods—such as superpixels or graph-based image partitions—to group visually similar regions of the target image before deciding tile sizes. This could create an even more adaptive and natural tiling pattern that respects object boundaries and semantic regions in the image, potentially leading to a more meaningful composition.

C. Dynamic Blending Strategies

While Poisson blending has proved effective for minimizing seams, it is computationally expensive and applies a uniform blending strategy across all tiles. Future work could involve dynamic blending strategies that choose different blending methods depending on the local characteristics. For example, alpha blending might suffice in smooth regions, while Poisson blending is reserved for high-contrast transitions. Alternatively, novel blending algorithms that exploit machine learning-based image harmonization techniques could further improve the seamless integration of tiles.

In summary, the future scope of this work spans technical enhancements, user experience improvements, and expansions into new domains. Each of these directions offers opportunities to push the boundaries of what photographic mosaics can achieve, both as a computational challenge and as an expressive art form.

VII. Key Resources

Our project leverages a mix of resources, ensuring a good foundation for developing our idea. Below, we highlight the key resources that have significantly contributed to our progress.

A. Datasets

The cornerstone of our implementation and evaluation is the *COCO dataset* which is open source. This dataset enabled us to fine-tune and test our implementations under a variety of scenarios reflecting real-world scenarios.

B. Code Bases and Computing Platforms

Our development process integrates a combination of custom-coded elements and adaptations from existing code bases. We plan to draw inspiration and technical insights from various GitHub repositories, which have informed the structure of our streamlined code base. This approach will allow us to focus on method initialization, ensuring compatibility and efficiency across different methods with minimal adjustments required.

We utilized resources like Google Colab, UIUC GPU Clusters (if needed).

C. Software and Libraries

The plan is to mainly use Python and its various libraries. The primary ones would be OpenCV for image processing, NumPy for data handling, and SciPy for computational tasks.

D. Hardware

Access to a machine with sufficient CPU/GPU capacity for processing large datasets and implementing advanced blending techniques. (Colab Pro, NCSA UIUC Servers)

E. Documentation Tools

Tools like Jupyter Notebook or Google Colab for development and LaTeX for detailed report preparation.

VIII. Team Contributions

The team has planned to complete the work with equal contribution from both the individuals.

- **Data Pre-Processing:** This part of the project was implemented in coalition by the us, ensuring both of us had the opportunity to understand the various aspects of the data which would be a key foundation for setting up the code bases and potentially come up with a mechanism to streamline the solution.
- **Code Base setup:** The idea became was to set up a code base, in which the only difference was that of initialization of various methods. This included several hours of research on Github and various online resources. Decisions were made in agreement with each other while the code base was set up by **Amaan**, and testing of the flexibility of the same wase done by **Imaad**.
- **Method Initialization and Evaluation:** With the way the codebase is set up, we divided the project implementation onto ipynb files in such a way that both the participants have an equal opportunity to implement every aspect. We alternated tasks, so roles in color matching, adaptive tiling, and blending techniques are rotated, ensuring that both participants gained hands-on experience with each method. For tile selection and post-processing, we used an iterative trial-and-error approach to refine the mosaic quality, with the best-performing techniques used in the final submission.
- **Report Writing:** The report was done in parallel wherein we plan to simultaneously write the various sections once we are done with the task. The plan here is for both the individuals to write a few lines on their own based on understanding, which before the final submission can be refined.

All in all, we declare equal contribution from both of us towards the completion of this project.

References

[1] Pengfei Xu, Jianqiang Ding, Hao Zhang, Hui Huang, "Discernible image mosaic with edge-aware adaptive tiles," *Computational Visual Media*, [March 2019].

[2] Guanping Dong, Sun Shanwei,Nanshou Wu,Hao Feng,Pingnan Huang, Minqiang Pan, "Research on automatic mosaic ceramic tiling method based on color matching," *[ResearchGate]*, [2021].

[3] X. Bai, "Overview of image mosaic technology by computer vision and digital image processing," *[IEEE]*, [2021].

[4] COCO Dataset-https://cocodataset.org/#home

[5] Flickr Creative Commons images-https://www.flickr.com/creativecommons/

[6] Photomosiac-https://github.com/worldveil/photomosaic

[7] Mosiac Art Maker-https://github.com/habedi/mosaic-art-maker