



## Phase\_2

Submitted to:

**Dr. Mahmoud Ibrahim Khalil**

**Eng: Ali Ahmed Ali**

**Eng: Dina Zakaria Mahmaud**

Submitted by:

<b>Nagy Ahmed Nagy</b>	<b>23p0365</b>
<b>Abdelrhman Mohammed Mahmoud Salah</b>	<b>23p0370</b>
<b>Ahmed Mosta Gomaa</b>	<b>23p0375</b>
<b>John George Mikhael</b>	<b>23P0266</b>
<b>Peter Maged Shokry</b>	<b>23p0192</b>

## Contents

Introduction .....	3
Milestone 1: Preprocessing and Artifact Preparation for Jigsaw Puzzle Analysis .....	3
1. Objective .....	3
2. Techniques Attempted and Justification .....	4
2.1 Contrast Limited Adaptive Histogram Equalization (CLAHE) .....	4
2.2 Gaussian Smoothing for Noise Reduction .....	4
2.3 Morphological Cleaning.....	4
2.4 Sobel Edge Detection .....	4
2.5 Patch-Based Image Splitting .....	4
3. Failure Cases and Limitations.....	4
3.1 Weak or Missing Contours .....	4
3.2 Noise-Induced False Contours .....	5
3.3 Patch Boundary Fragmentation .....	5
4. Suitability of Produced Artifacts for Later Assembly .....	5
Phase Two: Puzzle Assembly and Edge Matching.....	5
1. Overview of Phase Two .....	5
2. 2×2 Puzzle Assembly Using Exhaustive Edge Matching.....	5
Objective.....	5
Methodology.....	6
Reconstruction and Visualization .....	6
Ground Truth Mapping and Accuracy Evaluation .....	6
Results and Discussion .....	6
Summary.....	7
3. 4×4 Puzzle Assembly Using Beam Search.....	7
Objective.....	7
Methodology.....	8
Border Cost Design .....	8
Precomputation of Adjacency Costs.....	8
Beam Search Assembly Strategy.....	9

## Milestone 2: Preprocessing & Artifact Preparation

Reconstruction and Visualization .....	9
Ground Truth Mapping and Accuracy Evaluation .....	9
Results and Discussion .....	9
Summary.....	10
4. 8×8 Puzzle Assembly Using Beam Search.....	10
Objective.....	10
Methodology.....	11
Border Cost Function .....	11
Precomputation of Adjacency Costs.....	11
Reconstruction and Visualization .....	11
Ground Truth Mapping and Accuracy Evaluation .....	12
Results and Discussion .....	12
Summary.....	13
5. Performance Comparison Across Puzzle Sizes .....	13
6. Phase Two Discussion.....	13
7. Phase Two Conclusion .....	14
Final Conclusion.....	14

Repo: [https://github.com/Nagy-Ahmed-cs/Jigsaw\\_Puzzle/](https://github.com/Nagy-Ahmed-cs/Jigsaw_Puzzle/)

### Introduction

This project is conducted as a group assignment involving teams of four to five students and aims to reinforce both fundamental and advanced topics covered in the Computer Vision course. The project provides practical experience in designing and implementing complete solutions using **classical computer vision techniques**, without relying on machine learning or deep learning approaches.

The main objective of the project is to automatically analyze jigsaw puzzle pieces from digital images. This includes preprocessing input images, segmenting individual puzzle pieces, extracting their contours, and identifying potential matches between puzzle edges based on contour shape similarity. The system is intended to emulate the human cognitive process of solving jigsaw puzzles by comparing complementary edges, but in an automated and algorithmic manner.

To accomplish this task, a robust image processing pipeline must be developed. The pipeline focuses on enhancing image quality, suppressing noise, and accurately extracting puzzle piece contours. These contours are then described using rotation-invariant shape descriptors, allowing meaningful comparison between edges regardless of their orientation. Suitable geometric distance metrics are applied to evaluate similarity and suggest potential neighboring matches.

This project evaluates the application of several key computer vision concepts, including image preprocessing, enhancement, segmentation, contour extraction, shape description, and rotation-invariant matching. Although the edge matching component extends beyond the exact topics covered in lectures, it encourages students to expand their understanding and apply classical computer vision principles to a real-world problem through advanced geometric analysis.

The project is divided into two milestones. The first milestone focuses on preprocessing and preparing the jigsaw puzzle images for further analysis, while the second milestone builds upon these results to identify and visualize likely matches between puzzle pieces. Throughout the project, students are required to collaborate using GitHub, produce well-documented code and reports, and demonstrate their system's performance on both clean and challenging input cases.

### Milestone 1: Preprocessing and Artifact Preparation for Jigsaw Puzzle Analysis

#### 1. Objective

The objective of Milestone 1 is to prepare the jigsaw puzzle image dataset for subsequent analysis and assembly. Since accurate puzzle edge matching depends heavily on the quality of extracted contours, this phase focuses on designing a robust preprocessing pipeline using classical computer vision techniques. The pipeline aims to enhance puzzle piece boundaries, reduce noise,

segment relevant regions, and extract reliable contours suitable for rotation-invariant shape analysis in later stages.

## **2. Techniques Attempted and Justification**

### **2.1 Contrast Limited Adaptive Histogram Equalization (CLAHE)**

CLAHE was applied to grayscale image patches to correct illumination inconsistencies and enhance local contrast. Jigsaw puzzle images often contain shadows or low-contrast regions that obscure edge information. CLAHE was chosen because it improves edge visibility without excessively amplifying noise, making it suitable for images with uneven lighting.

### **2.2 Gaussian Smoothing for Noise Reduction**

Gaussian smoothing was applied prior to thresholding to suppress high-frequency noise that could interfere with segmentation. Although edge-preserving filters such as bilateral filtering were considered, Gaussian smoothing provided stable results with lower computational cost, making it appropriate for processing a large number of image patches efficiently.

### **2.3 Morphological Cleaning**

A morphological opening operation using a  $3 \times 3$  structuring element was applied to remove small isolated noise in the thresholded masks. This step improves segmentation quality by preserving meaningful structures while eliminating minor artifacts that could result in false contour detection.

### **2.4 Sobel Edge Detection**

Sobel filters were applied in both horizontal and vertical directions, and the gradient magnitude was computed and normalized. Sobel edge detection highlights structural boundaries and provides clear edge representations that support accurate contour extraction.

### **2.5 Patch-Based Image Splitting**

Each full puzzle image was divided into an  $N \times N$  grid of smaller patches. The preprocessing pipeline, contour extraction, and descriptor computation were applied independently to each patch. This approach simplified the processing workflow and enabled efficient localized analysis, although it introduces certain limitations discussed later.

## **3. Failure Cases and Limitations**

### **3.1 Weak or Missing Contours**

Some patches exhibited low contrast or lacked clear puzzle boundaries, resulting in weak or missing contours even after enhancement. While CLAHE improved edge visibility in many cases, it could not fully resolve extreme low-contrast scenarios.

### 3.2 Noise-Induced False Contours

Lighting variations occasionally introduced noise that produced false contours after thresholding. Morphological opening removed most small artifacts; however, in extreme cases, some noise-induced contours remained.

### 3.3 Patch Boundary Fragmentation

Due to uniform patch-based splitting, some puzzle pieces were divided across patch boundaries. This produced incomplete contours that do not represent the full shape of the puzzle piece. This limitation will be addressed in later phases by adopting full-piece segmentation rather than grid-based partitioning.

## 4. Suitability of Produced Artifacts for Later Assembly

The artifacts generated in Milestone 1 are well suited for subsequent analysis and assembly:

- **Clean binary masks** support reliable and deterministic contour extraction.
- **Enhanced grayscale images and Sobel edge maps** emphasize boundary information required for shape analysis.
- **Extracted contours** provide accurate geometric representations of puzzle pieces.
- **Fourier descriptors** offer rotation- and scale-invariant representations necessary for edge matching in Milestone 2.
- **Organized output folders** improve reproducibility, debugging, and integration with later stages of the project.

## Phase Two: Puzzle Assembly and Edge Matching

### 1. Overview of Phase Two

The objective of Phase Two is to assemble jigsaw puzzle pieces by identifying likely neighboring pieces based on edge similarity. Building on the preprocessing and contour extraction performed in Phase One, this phase focuses on defining edge comparison strategies, evaluating puzzle piece compatibility, and reconstructing the puzzle layout. All methods rely exclusively on classical image processing and geometric analysis techniques, without the use of machine learning or deep learning models.

### 2. 2×2 Puzzle Assembly Using Exhaustive Edge Matching

#### Objective

The objective of the 2×2 experiment is to validate the proposed edge-matching strategy on a small-scale puzzle where all possible piece arrangements can be evaluated exhaustively. This experiment serves as a baseline for assessing the effectiveness of the border similarity metric before extending the approach to larger puzzle configurations.

## Methodology

Each scrambled  $2 \times 2$  puzzle image is divided into four equal-sized pieces. The pieces are converted from the BGR color space to the LAB color space to improve robustness to illumination variations, as LAB separates luminance information from chromatic components.

An exhaustive permutation-based search is employed to determine the correct arrangement of puzzle pieces. All possible permutations of the four pieces ( $4! = 24$  configurations) are evaluated. For each permutation, the pieces are arranged into a  $2 \times 2$  grid, and a matching cost is computed based on the similarity between adjacent piece borders.

Edge similarity is calculated using pixel-wise absolute differences along shared borders. Horizontal compatibility is evaluated between left–right neighbors, while vertical compatibility is evaluated between top–bottom neighbors. The total cost for a permutation is obtained by summing the border differences across all internal boundaries. The permutation with the minimum total cost is selected as the optimal solution.

## Reconstruction and Visualization

Once the optimal permutation is identified, the solved puzzle image is reconstructed by concatenating the corresponding puzzle pieces according to the selected grid arrangement. The reconstructed images are saved for visual inspection, enabling qualitative evaluation of the assembly results.

## Ground Truth Mapping and Accuracy Evaluation

When a ground truth (correct) image is available, an additional validation step is performed. Each scrambled puzzle piece is matched to its corresponding ground truth piece using pixel-wise difference comparison. This mapping allows the solved puzzle layout to be expressed using true piece identifiers.

Performance is evaluated using a neighbor-based accuracy metric, which measures the proportion of correctly matched adjacent piece pairs relative to the total number of internal boundaries. This metric focuses on local adjacency correctness rather than requiring a fully correct global reconstruction.

## Results and Discussion

The exhaustive search approach produces reliable results for  $2 \times 2$  puzzles due to the limited number of possible configurations. The use of the LAB color space improves robustness to lighting variations, while pixel-based border comparison provides an effective similarity measure for small puzzle sizes.

However, the factorial growth in the number of permutations makes this approach computationally infeasible for larger puzzles. Despite this limitation, the  $2 \times 2$  experiment

successfully demonstrates the validity of the proposed edge-matching cost function and establishes a solid foundation for developing scalable assembly strategies.



### Summary

The  $2 \times 2$  puzzle assembly experiment confirms that classical image processing techniques combined with exhaustive permutation search can effectively reconstruct small jigsaw puzzles. The insights gained from this experiment guide the design of scalable methods for assembling larger puzzles, which are addressed in the following sections.

### 3. $4 \times 4$ Puzzle Assembly Using Beam Search

#### Objective

The objective of the  $4 \times 4$  experiment is to extend the puzzle assembly approach to a larger and more realistic puzzle size, where exhaustive permutation search becomes computationally infeasible. This experiment evaluates the scalability of the edge-matching strategy by introducing a guided search method that balances solution quality and computational efficiency.

## Methodology

Each scrambled  $4 \times 4$  puzzle image is divided into sixteen equal-sized pieces. Similar to the  $2 \times 2$  case, all pieces are converted from the BGR color space to the LAB color space to reduce sensitivity to illumination variations and to better separate luminance from chromatic information.

Unlike the  $2 \times 2$  experiment, where all permutations can be evaluated, the number of possible configurations for a  $4 \times 4$  puzzle ( $16!$ ) is prohibitively large. To address this challenge, a beam search strategy is employed. Beam search incrementally constructs the puzzle grid while maintaining only a fixed number of the most promising partial solutions at each step. This allows exploration of high-quality candidates without evaluating the full search space.

## Border Cost Design

To measure compatibility between adjacent puzzle pieces, a composite border cost function is used. This function consists of two components:

### 1. Color Difference Term

The absolute difference between corresponding pixels along adjacent borders is computed to capture color continuity between neighboring pieces.

### 2. Gradient Consistency Term

A gradient-based term is added to enforce smooth transitions across boundaries. This term compares the gradient at the border with the gradient just inside the piece, penalizing sharp discontinuities.

The total border cost is computed as the sum of the color difference and a weighted gradient difference term. This formulation improves robustness against false matches caused by similar colors but inconsistent edge structure.

## Precomputation of Adjacency Costs

To improve efficiency during the search process, all pairwise horizontal and vertical border costs between puzzle pieces are precomputed and stored in cost matrices. Horizontal costs represent left–right compatibility, while vertical costs represent top–bottom compatibility. These precomputed matrices allow rapid evaluation of candidate placements during beam search.

### Beam Search Assembly Strategy

The puzzle is assembled sequentially in row-major order. At each step, the algorithm considers placing a new piece in the next grid position. The added cost is computed based on compatibility with the left neighbor, the top neighbor, or both, depending on the position within the grid.

For each partial configuration, multiple candidate extensions are generated by trying unused puzzle pieces. The candidates are sorted by cumulative cost, and only the top configurations determined by the beam width—are retained for the next iteration. This process continues until the full  $4 \times 4$  grid is constructed. The configuration with the lowest total cost is selected as the final solution.

### Reconstruction and Visualization

After determining the optimal grid configuration, the solved puzzle image is reconstructed by concatenating the corresponding puzzle pieces according to their positions in the grid. The reconstructed images are saved for visual inspection, allowing qualitative assessment of assembly quality and boundary alignment.

### Ground Truth Mapping and Accuracy Evaluation

When a ground truth image is available, each scrambled piece is matched to its true counterpart using pixel-wise difference comparison. This mapping enables evaluation of the reconstructed grid in terms of true piece identities rather than scrambled indices.

Performance is measured using a **neighbor-based accuracy metric**, which computes the fraction of correctly matched adjacent piece pairs (both horizontal and vertical) relative to the total number of internal boundaries. This metric focuses on local correctness and provides a meaningful evaluation of partial assembly quality even when the entire puzzle is not perfectly reconstructed.

### Results and Discussion

The beam search approach significantly reduces computational complexity compared to exhaustive search while maintaining reasonable assembly accuracy. The inclusion of gradient information in the border cost improves discrimination between visually similar but structurally incompatible edges.

However, the approach remains sensitive to accumulated local errors, as early incorrect placements may influence later decisions. The fixed beam width limits search diversity and may

exclude globally optimal configurations in challenging cases. Despite these limitations, the  $4 \times 4$  experiment demonstrates that guided search strategies combined with classical edge-matching metrics can effectively scale puzzle assembly beyond trivial sizes.



### Summary

The  $4 \times 4$  puzzle assembly experiment demonstrates the effectiveness of beam search as a scalable alternative to exhaustive permutation search. By combining color-based and gradient-based edge compatibility measures, the system achieves a balance between accuracy and computational feasibility. These results motivate further evaluation on larger puzzle sizes, which is addressed in the subsequent  $8 \times 8$  experiment.

## 4. $8 \times 8$ Puzzle Assembly Using Beam Search

### Objective

The objective of the  $8 \times 8$  experiment is to evaluate the robustness and scalability of the proposed puzzle assembly approach on a large-scale puzzle configuration. At this size, the problem becomes significantly more challenging due to the increased number of pieces, higher ambiguity between edges, and the exponential growth of the search space. This experiment tests whether classical edge-matching techniques combined with guided search can still produce meaningful assembly results under realistic constraints.

## Methodology

Each scrambled  $8 \times 8$  puzzle image is divided into sixty-four equal-sized pieces. As in previous experiments, all pieces are converted from the BGR color space to the LAB color space to improve robustness to illumination variations and color inconsistencies.

Exhaustive permutation search is completely infeasible for this puzzle size due to the enormous number of possible configurations ( $64!$ ). Therefore, the same **beam search strategy** introduced in the  $4 \times 4$  experiment is employed, with a reduced beam width to maintain computational feasibility.

The puzzle grid is constructed incrementally in row-major order. At each step, the algorithm selects candidate pieces for the next grid position based on compatibility with already placed neighbors. Only a fixed number of the most promising partial solutions are retained at each iteration, significantly reducing the search space while preserving reasonable solution quality.

## Border Cost Function

The border cost function used for the  $8 \times 8$  experiment is identical to that used in the  $4 \times 4$  case to ensure consistency across scales. The cost consists of:

1. **Color Continuity Term**

The L1 distance between corresponding pixels along adjacent borders, measuring color similarity.

2. **Gradient Consistency Term**

A gradient-based penalty that compares edge gradients with inner gradients to discourage abrupt boundary transitions.

The gradient term is weighted to emphasize structural consistency, which becomes increasingly important as puzzle size grows and color similarity alone becomes less discriminative.

## Precomputation of Adjacency Costs

To improve runtime efficiency, all horizontal and vertical border compatibility costs between puzzle piece pairs are precomputed and stored in matrices. These matrices allow fast lookup of compatibility scores during beam search and significantly reduce redundant computations during puzzle assembly.

## Reconstruction and Visualization

After the beam search completes, the lowest-cost configuration is selected as the final solution. The solved puzzle image is reconstructed by concatenating the selected puzzle pieces according to the final grid arrangement. These reconstructed images are saved and visually inspected to assess overall structural coherence and boundary alignment.

## Ground Truth Mapping and Accuracy Evaluation

When ground truth images are available, scrambled puzzle pieces are matched to their corresponding true pieces using pixel-wise difference comparison. This mapping allows the reconstructed grid to be evaluated in terms of true piece identifiers rather than scrambled indices.

Performance is evaluated using the same **neighbor-based accuracy metric** employed in previous experiments. This metric measures the proportion of correctly matched adjacent piece pairs relative to the total number of internal boundaries. While this metric does not require perfect global reconstruction, it provides a meaningful assessment of local assembly correctness in large-scale puzzles.

## Results and Discussion

The  $8 \times 8$  experiment demonstrates the increasing difficulty of puzzle assembly as puzzle size grows. While the beam search approach successfully maintains computational feasibility, assembly accuracy generally decreases compared to smaller puzzle sizes. This degradation is primarily due to accumulated local errors, increased edge ambiguity, and the limited beam width, which restricts exploration of the full solution space.

Despite these challenges, the results show that the proposed classical computer vision approach can still identify a non-trivial number of correct neighboring relationships. The inclusion of gradient-based edge compatibility improves robustness against visually similar but structurally incompatible edges, although it cannot fully eliminate ambiguities in highly complex scenes.



## Summary

The 8×8 puzzle assembly experiment highlights both the strengths and limitations of classical edge-matching techniques when applied to large-scale problems. Beam search enables scalable assembly, but accuracy is constrained by local decision errors and limited search diversity. These results emphasize the trade-off between computational feasibility and reconstruction quality and provide valuable insight into the challenges of solving large jigsaw puzzles using purely classical computer vision methods.

## 5. Performance Comparison Across Puzzle Sizes

This section compares the performance of the proposed puzzle assembly approach across the three evaluated puzzle sizes: 2×2, 4×4, and 8×8. The comparison highlights the impact of puzzle size on assembly accuracy, computational complexity, and robustness.

For the **2×2 puzzles**, exhaustive permutation search was feasible due to the small number of possible configurations. As a result, the system achieved high neighbor accuracy and consistent reconstruction quality. The edge-matching cost function was sufficient to identify the correct configuration in most cases, making this setup an effective baseline for validating the approach.

In the **4×4 puzzles**, exhaustive search was no longer practical, and a beam search strategy was introduced. While this significantly reduced computational complexity, accuracy became more sensitive to local edge ambiguities and early placement decisions. Nevertheless, the combination of color-based and gradient-based border costs allowed the system to maintain reasonable neighbor accuracy while remaining computationally efficient.

The **8×8 puzzles** posed the greatest challenge due to the large number of pieces and increased similarity between edges. Although beam search enabled feasible execution, neighbor accuracy generally decreased compared to smaller puzzle sizes. Errors accumulated as the grid grew, and the limited beam width restricted exploration of alternative configurations. Despite these challenges, the system was still able to recover a meaningful number of correct neighboring relationships, demonstrating partial success at large scale.

Overall, the results show a clear trade-off between puzzle size, computational feasibility, and assembly accuracy. As puzzle complexity increases, guided search strategies become essential, but accuracy is constrained by local decision-making and limited search diversity.

## 6. Phase Two Discussion

Phase Two demonstrates that classical computer vision techniques can be effectively applied to the problem of jigsaw puzzle assembly without relying on machine learning. The edge-matching strategy based on color continuity and gradient consistency provides a meaningful measure of compatibility between puzzle pieces.

However, the experiments also reveal inherent limitations of purely classical approaches. The reliance on local edge information makes the system vulnerable to accumulated errors, especially in large puzzles. Beam search mitigates computational explosion but introduces a trade-off between search depth and solution quality.

These observations highlight the importance of careful cost function design and search strategy selection when scaling classical image processing solutions to complex real-world problems.

### 7. Phase Two Conclusion

Phase Two successfully achieves its objective of assembling jigsaw puzzles using classical edge-matching techniques and guided search strategies. Starting from an exhaustive solution for small puzzles and extending to beam search for larger configurations, the system demonstrates a clear progression in scalability and complexity handling.

While assembly accuracy decreases as puzzle size increases, the results confirm that meaningful puzzle reconstruction is possible using only classical computer vision methods. The insights gained from Phase Two provide a strong foundation for understanding the challenges of large-scale puzzle assembly and motivate potential improvements in future work.

### Final Conclusion

This project presented a complete classical computer vision pipeline for the automatic analysis and assembly of jigsaw puzzle images. The system was designed and evaluated across two phases, beginning with robust image preprocessing and contour extraction, and culminating in puzzle assembly through edge matching and guided search strategies.

In Phase One, a reliable preprocessing pipeline was developed to enhance image quality, reduce noise, segment puzzle pieces, and extract meaningful contours and rotation-invariant shape descriptors. The produced artifacts were consistent and well-structured, providing a strong foundation for subsequent assembly tasks.

Phase Two demonstrated the feasibility of assembling puzzles of increasing complexity using only classical image processing techniques. Exhaustive search achieved accurate reconstruction for small puzzles, while beam search enabled scalable assembly for larger configurations. Experimental results across  $2 \times 2$ ,  $4 \times 4$ , and  $8 \times 8$  puzzles highlighted the trade-off between computational feasibility and assembly accuracy, particularly as puzzle size increased.

While the system successfully identified meaningful neighboring relationships without relying on machine learning, the experiments also revealed inherent limitations of purely classical approaches, including sensitivity to local errors and accumulated mismatches in large-scale puzzles. Nevertheless, the project illustrates that carefully designed edge-matching metrics and search strategies can effectively address complex visual assembly problems.

## Milestone 2: Preprocessing & Artifact Preparation

Overall, this work reinforces core computer vision concepts and provides practical insight into the challenges of solving real-world problems using classical techniques. The project demonstrates both the strengths and limitations of traditional image processing methods and lays the groundwork for potential future improvements.