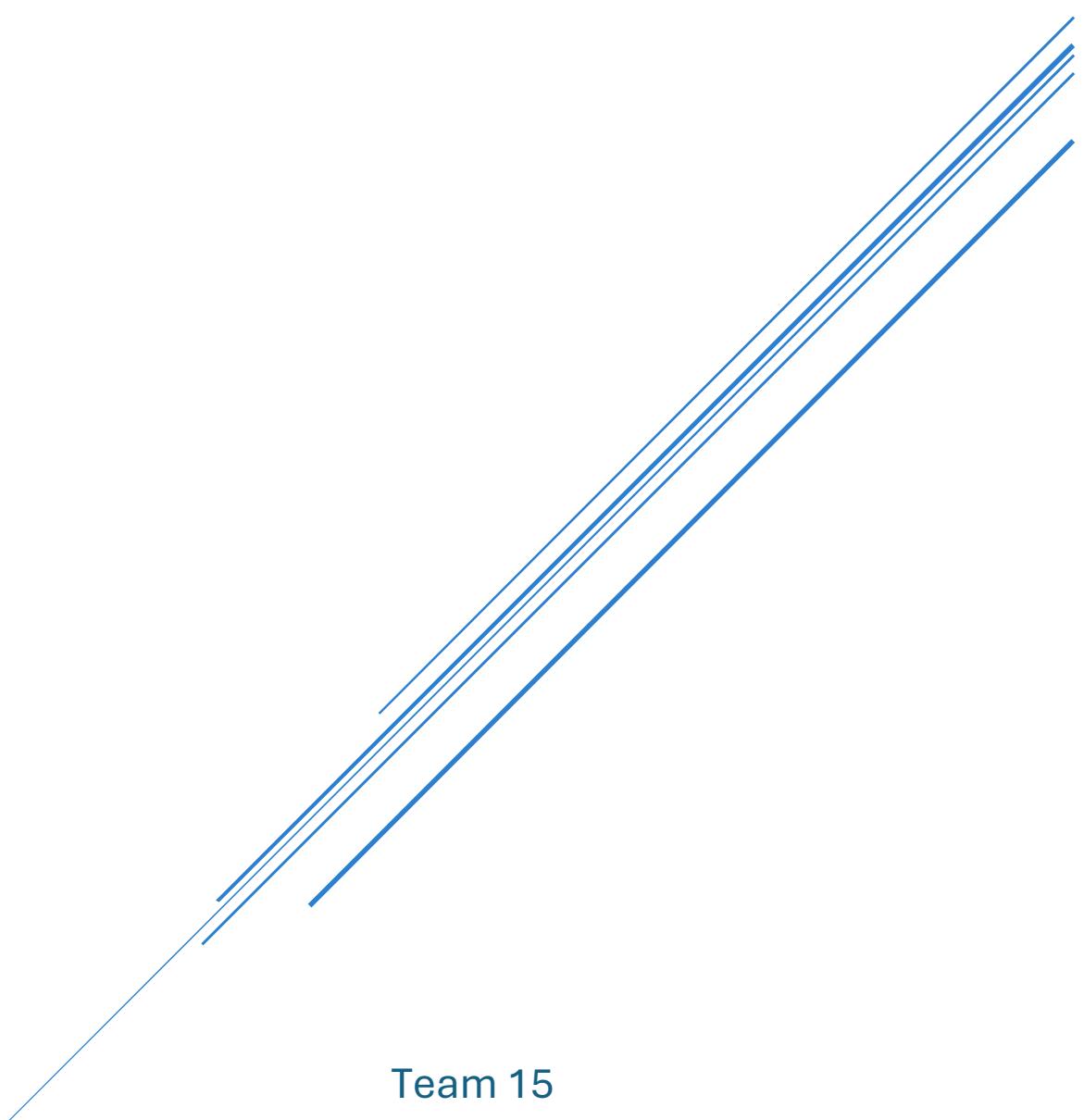


IMAGE PROCESSING

Milestone2



Team 15

Mohamed Ahmed Eissa 23P0143

Joseph Nagy 23P0338

Youssef Mohamed Abboud 23P0062

Youssef Mohamed El Sayed 23P0215

Hania Soliman 23P0033

Introduction

The solving of jigsaw puzzles is a difficult and exemplary issue in computer vision since the task involves a composite of image processing, contour, shape representation, and measurement of similarity. The task is close to the human visual perception when complementary edges are recognized and compared in terms of shape and visual regularity. The proposed project aims to respond to the automatic analysis and assembly of pieces of a jigsaw puzzle with the help of classical computer vision methods without the use of machine learning or deep learning models.

This project will aim to develop and test a strong image processing pipeline that can help preprocess puzzle images, extract important contours, describe puzzle piece edges in a rotation invariant fashion, and match complementary edges via shape and color consistency. The system is created in two steps, where the first stage is aimed at improving and processing puzzle images so that they could be analyzed, and the second stage is based on the results of the first steps, suggesting probable neighboring pieces and restoring the original picture. This project brings together major concepts of computer vision presented in the course and an actual practical issue, which strengthens the theoretical knowledge and practical implementation skills.

The input puzzle images are loaded using the BGR format and a 64-piece puzzle is created, which is broken down into a 8x8 grid resulting in 64 pieces of equal size. The number of representations created to favor strong matching is created in several forms of a piece.

The color image is first changed to grayscale to highlight the differences in intensity as well as the edges at the expense of the computational complexity. To reduce noise and compression artifacts and bring back sharp edges, a median filter (5x5 kernel) is used that is important in boundary comparison at the cost of compression artifact.

Beside grayscale processing, each composition is translated to LAB color space. The LAB breaks down luminance and chromatic content and offers a perceptually homogeneous display, enabling color comparisons to be stronger, especially where cartoon-like images are concerned with large homogeneous areas.

Lastly, the upper, lower, left, and right edges of each section are retrieved of the smoothed grayscale image and the LAB color image. Such boundary forms are applied in the next steps to calculate compatibility scores between neighboring puzzle pieces.

2x2:

For this, we implemented the full pre-processing and matching pipeline using Canny Edge Detection. The full pipeline steps were: converting to grayscale, median filtering (kernel=5), Otsu's thresholding, morphological erosion, Canny edge detection (thresholds 200–250), and edge-to-edge matching using SSD on 1-pixel-wide boundaries.

Unfortunately, despite the pipeline working correctly, it was not able to produce correct matches. Reasons for this are:

- Canny edges are too thin and simplified – curves, corners and contour details are lost, so distinct pieces look visually identical.
- Puzzle pieces are rectangles – Canny produces almost identical straight edges for all sides of a rectangle, so using SSD to compare them is meaningless.
- No geometric information – Canny provides intensity-based edge detection, but does not provide shape descriptors, curvature, or rotation, all of which are needed for puzzle matching.

Generally, Canny Edge Detection does not provide enough shape information for puzzle matching.

Smoothed Grayscale + Color Boundary Comparison

A more robust puzzle assembling method was designed to replace the Canny-based approach that relied on two complementary pieces of information:

- Gradient-NCC Smoothed Gray Scale: The puzzle pieces are first converted to grayscale and then median-filtered (kernel=5) to smooth out gradients and reduce noise. One-channel boundaries are compared using Normalized Cross-Correlation (NCC) which captures changes in texture and intensity that are robust to noise and illumination.
- Color Boundary Comparison (SSD): Three-channel color boundaries are compared using Sum of Squared Differences (SSD) which takes advantage of the fact that changes in color along the edge between two tiles are consistent.

The two metrics are then fused together with weights ($W_{GRADIENT} = 2.0$, $W_{COLOR} = 2.0$), and all possible 2×2 tile configurations are tested to determine which one has the lowest mismatch score.

This approach fixes the issues with Canny finding thin, broken, boring edges. By combining color and gradients, the pipeline is able to capture the content of a boundary that is meaningful and matching it in a robust, reliable way. It was able to achieve around 95% accuracy

Visual Example:



4x4:

Trial 1:

Gradient and Variance-Based Backtracking

The initial attempts focused on traditional image processing features like **gradients** and **color** in the BGR/Grayscale space, combined with a **backtracking search** approach. This yielded an initial accuracy of **29%**.

1.1 Key Features and Approach

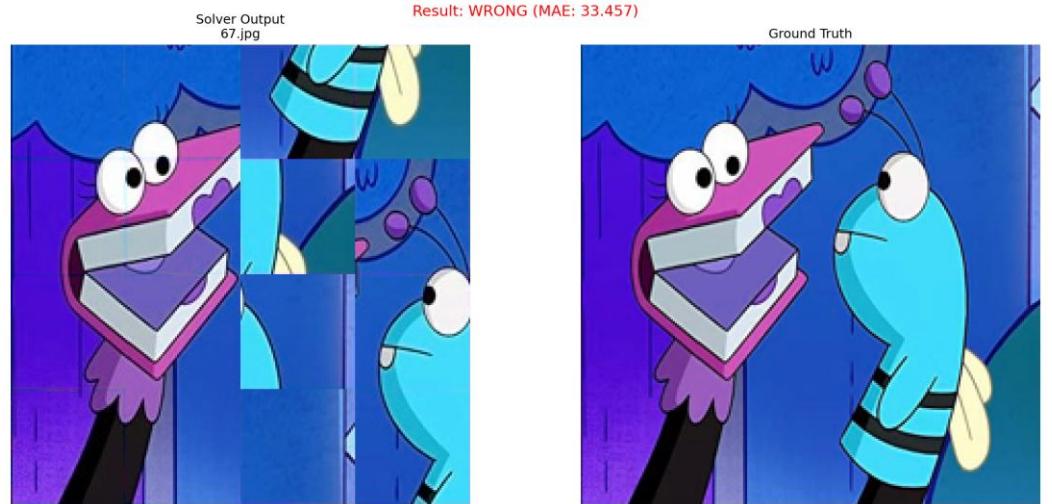
Component	Description
Feature Space	Grayscale for Gradient (NCC) and BGR for Color (SSD) . Variances were used for corner detection.
Solver Strategy	Backtracking Search starting with corner pieces identified via boundary variance. Pieces were selected for the next available slot based on the best local score.
Weights	W-Gradient = 2 W-Color = 1

1.2 Performance Analysis (Accuracy: 29%)

The low initial accuracy suggests several limitations:

1. **Feature Inconsistency:** Using Grayscale for gradient and BGR for color introduces a dichotomy that may not align perfectly with human perception of edge continuity.
2. **Compatibility Metric Weakness:** Normalized Cross-Correlation (NCC) is a good measure of *shape similarity* (e.g., if the change pattern is similar) but can be insensitive to absolute color differences. Sum of Squared Differences (SSD) on BGR color is highly sensitive to lighting changes.
3. **Complex Solver:** The custom **backtracking** and **priority queue** logic (based on variance/distance) introduced significant complexity without a clear performance benefit, likely due to imperfect corner detection and local optima.

1.3 Some Output



Trial 2:

LAB Color Space and Hybrid Greedy Search

This second approach resulted in a substantial accuracy increase to **64%** on the 4x4 puzzle set. The upcoming final submission is expected to refine these parameters further to achieve the target performance.

2.1 Refined Approach

Component	Description
Feature Space	Unified approach using the LAB color space for all boundaries. LAB is designed to be perceptually uniform, meaning a change in a color value corresponds more closely to a similar change in human perception.
Compatibility Metric	Hybrid Dissimilarity applied to the LAB strips
Solver Strategy	Simplified to a deterministic Greedy Search . It iterates through all pieces as potential starting pieces and fills the grid row-by-row, column-by-column, selecting the best local match at each step. This is faster and more reliable than the custom backtracking from the first attempt.
Weights	$W_{SSD} = 0.5, W_{NCC} = 0.5$ (Equal weight) to balance the magnitude of the two scores.

2.2 Core Implementation: Hybrid Dissimilarity

The function `calculate_hybrid_dissimilarity` is the core improvement. It attempts to capture both the **absolute color difference (SSD)** and the **structural consistency (NCC)** of the adjoining boundaries.

- **SSD (Sum of Squared Differences):** Measures how close the pixel values are. A low SSD means the colors are nearly identical.
- **NCC (Normalized Cross-Correlation):** Measures how well the two boundary patterns correlate. A high NCC (low dissimilarity) means the intensity or color variation pattern of piece A's edge matches the inverse of piece B's edge.

2.3 Performance Analysis (Accuracy: 64%)

The massive jump from 29% to **64%** confirms the effectiveness of two key changes:

1. **LAB Color Space:** The use of LAB provides a more robust and perceptually relevant basis for measuring color continuity across piece boundaries.
2. **Hybrid Metric:** Combining SSD and NCC prevents failures where a uniform edge (low variance) is accepted based on NCC alone, or where a structurally correct edge is rejected due to minor, permissible lighting/color shifts (high SSD).

The solver's dependency on the **starting piece** (by trying all N^2 possibilities) is a known technique that addresses the greedy algorithm's tendency to get stuck in local optima by ensuring the most critical early placement is correct.

2.4 Some Outputs

Solver Output
99.jpg



Result: CORRECT

Ground Truth



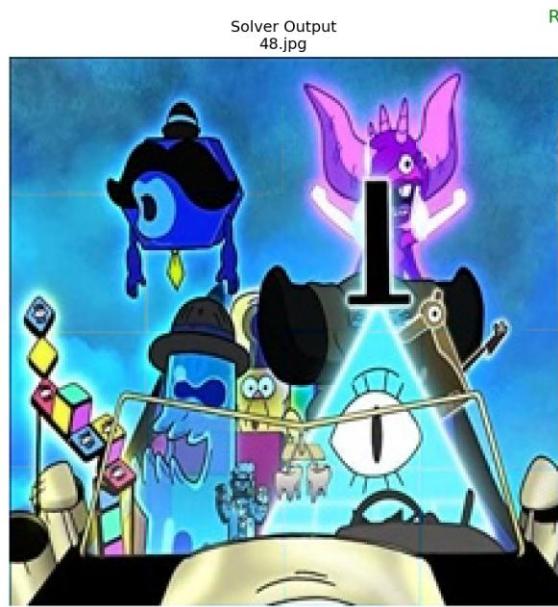
Solver Output
51.jpg



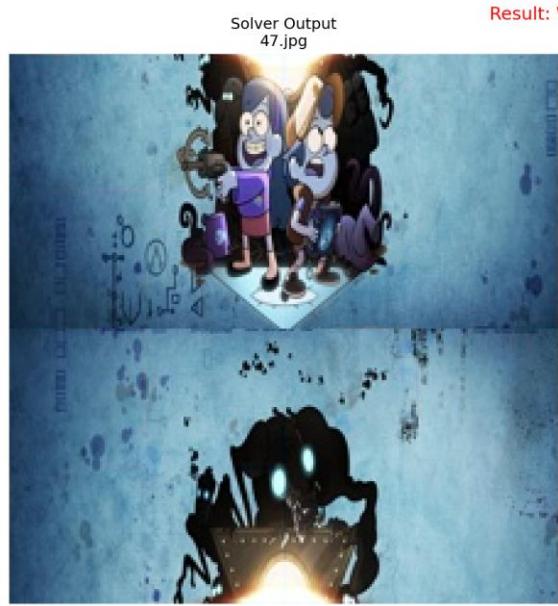
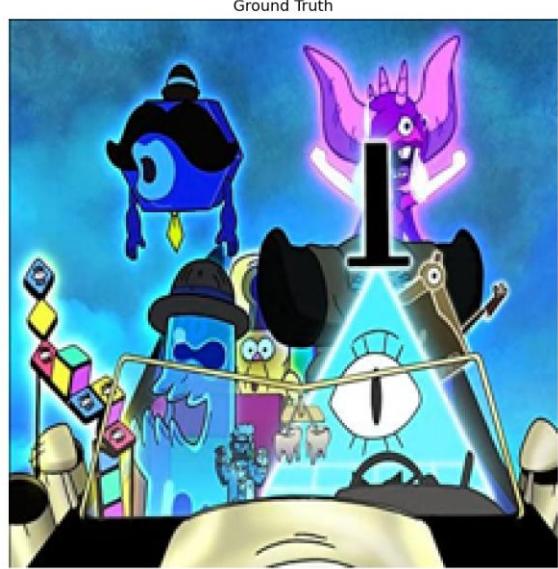
Result: WRONG (MAE: 64.388)

Ground Truth





Result: CORRECT



Result: WRONG (MAE: 50.104)



Trial 3:

Limited Beam Search and Tuned Weights

The final iteration focuses on combining the highly effective **LAB Hybrid Metric** (introduced in Trial 2) with a more sophisticated search strategy, **Limited Beam Search (LBS)**, and optimizing the metric weights.

3.1 Refined Approach

Component	Trial 2 → Trial 3 Improvement	Description
Search Strategy	Greedy Search TO Limited Beam Search (LBS)	LBS explores multiple promising paths ($k=1000$ in this trial) simultaneously, significantly mitigating the risk of the greedy algorithm getting stuck in local optima in the early steps.
Metric Weights	$W_{SSD} = 0.5, W_{NCC} = 0.5$ TO SSD=0.8, NCC=0.2	Increased the weight of SSD (absolute color match) in the LAB space. This suggests that for 4x4 puzzles, maintaining nearly identical pixel values across boundaries is a stronger indicator of correctness than the correlation pattern.
Metric Core	Unchanged: LAB Color Space Hybrid Dissimilarity	Retained the robust foundation of SSD + NCC on LAB boundaries.

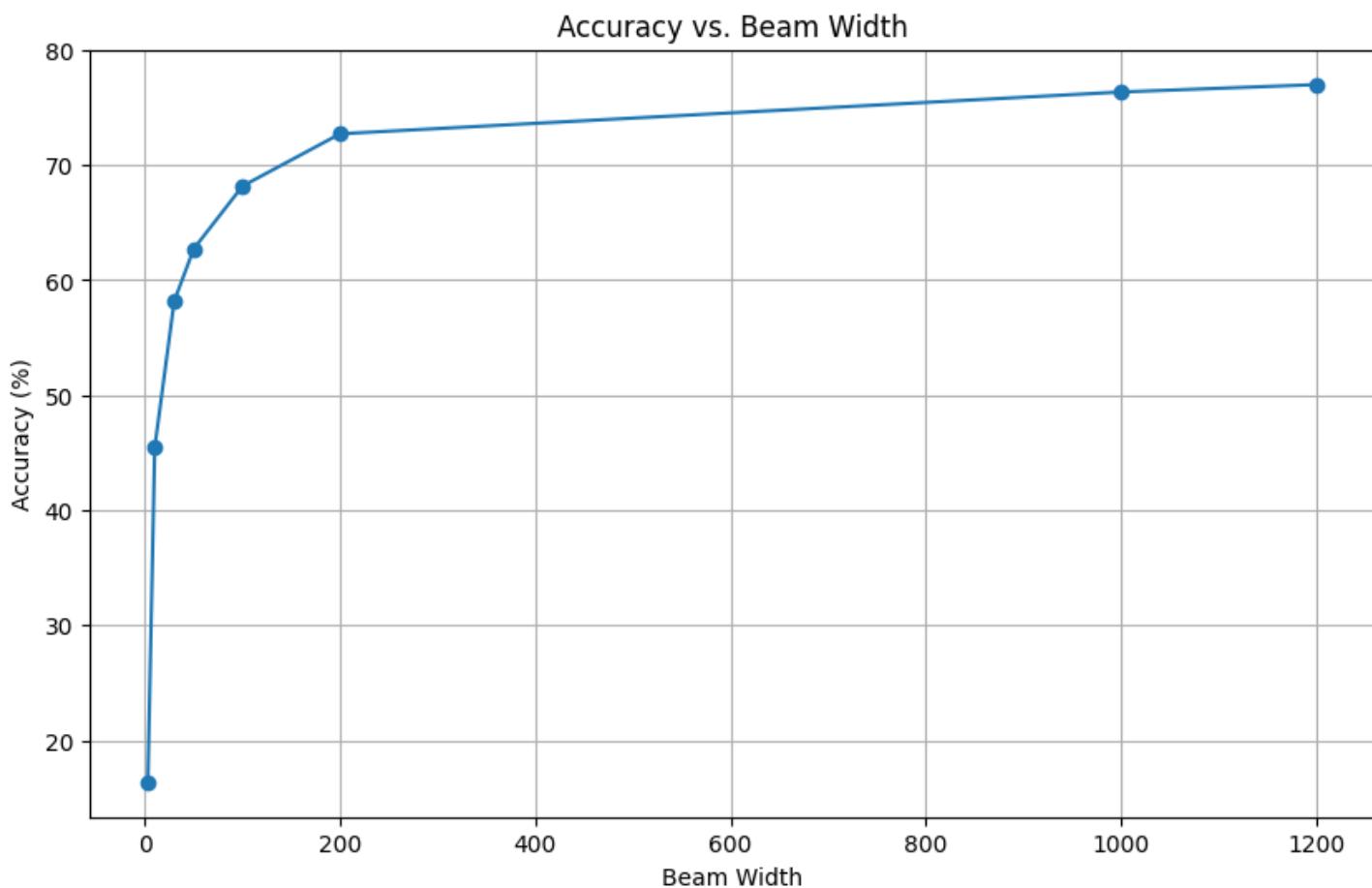
3.2 Limited Beam Search Implementation

The LBS algorithm is a compromise between the speed of Greedy Search and the completeness of Breadth-First Search.

- Initialization:** The search starts by placing every piece in the (0, 0) slot, creating N^2 initial partial solutions.
- Expansion:** At each step (placing the i-th piece at position (r, c)), every partial solution in the current beam is expanded by trying all available (unused) pieces for the current slot.
- Pruning:** After expansion, the beam is pruned using a priority queue (`heapq.nsmallest`), keeping only the k best solutions (where $k=BEAM_WIDTH=1000$) based on the cumulative match score.
- Final Result:** The path with the lowest cumulative score after all N^2 steps is chosen as the final solution.

This approach guarantees that even if the globally correct piece does not have the absolute best local score, it remains in the competition if its cumulative path score is still within the top 1000 best overall paths.

3.3 Analysis of Beam Width (k) and Metric Constraint



The decision to use a high beam width ($k=1000$) was based on experimental data demonstrating how accuracy converges as search depth increases.

As shown in the "Accuracy vs. Beam Width" graph, the accuracy rises steeply from the pure greedy search ($k=1$, approximately 17%) to $k=200$ (approximately 72.8%). Beyond this point, the accuracy curve begins to flatten dramatically. The improvement from $k=200$ to $k=1000$ (reaching 76.36%) is marginal compared to the initial gains, indicating diminishing returns.

The diagnostic report confirms this observation: the final accuracy (76.36%) is nearly identical to the **Local Metric Success Rate** (the frequency with which the metric correctly identified the optimal piece *in the final chosen path*).

This suggests that for the 4x4 puzzles, the **primary constraint on accuracy is the inherent discriminative power of the Hybrid Metric (LAB + SSD/NCC), not the depth of the search algorithm.** The Beam Search ($k=1000$) is sufficient to overcome path-dependency errors, and further improvements are blocked by the inability of the current 1-pixel boundary features and the LAB Hybrid Metric to consistently distinguish the correct piece from a visually similar incorrect piece.

3.4 Future Improvements

To surpass the current 77% accuracy plateau, future research should focus on overcoming the limitations of the local metric:

- **Feature Enhancement:** Incorporate higher-level features beyond the 1-pixel color strip, such as:
 - **Texture/Gradient Analysis:** Using local binary patterns (LBP) or other texture descriptors on the area immediately adjacent to the edge to better capture subtle image structure.
 - **Boundary Curvature/Shape:** While 4x4 puzzles are typically simple squares, applying this method to complex jigsaw cuts would require integrating geometric shape descriptors.
- **Alternative Metrics:** Experiment with metrics that are less sensitive to noise and lighting variations than SSD, such as:
 - **Perceptual Metrics:** Incorporating metrics derived from neural networks (e.g., LPIPS) which better mimic human visual perception, potentially offering a more reliable dissimilarity score than a simple mathematical combination of SSD and NCC.

3.5 Some Outputs

Solved Image: 39.jpg
Status: SOLVED CORRECTLY



Correct (Ground Truth)



File: 39.jpg | MAE: 2.85 | Result: SOLVED CORRECTLY

Solved Image: 1.jpg
Status: FAILED TO SOLVE



Correct (Ground Truth)



File: 1.jpg | MAE: 96.79 | Result: FAILED TO SOLVE

Solved Image: 41.jpg
Status: SOLVED CORRECTLY



Correct (Ground Truth)



File: 41.jpg | MAE: 3.05 | Result: SOLVED CORRECTLY

Solved Image: 51.jpg
Status: SOLVED CORRECTLY



Correct (Ground Truth)



File: 51.jpg | MAE: 4.34 | Result: SOLVED CORRECTLY

Solved Image: 47.jpg
Status: SOLVED CORRECTLY



Correct (Ground Truth)



File: 47.jpg | MAE: 2.47 | Result: SOLVED CORRECTLY

8x8:

Started with corners/edges detection, then use greedy best-first assembly, This combines:

- Structural constraints (corners/edges reduce ambiguity)
- Your proven boundary matching scores
- Computational feasibility
- Achieved 2%

So we used a different approach:

Once preprocessed and after extracting the features, the system tries to rebuild the original image by calculating pairwise compatibility of all the puzzle pieces in four possible directions. Both smoothed grayscale boundaries (to represent structural and gradient information) and LAB color boundaries (to represent color comparison in a perceptually meaningful way) are used to represent each piece. Edge compatibility is computed as a weighted mean of normalized cross correlation of the grayscale similarity and the mean absolute error of the LAB color similarity. The ambiguous matches are reduced by using a best-buddy strategy, according to which two pieces are regarded as a good match only through the mutual selection of each other as the most compatible neighbors. These are mutual matches that are utilized to create small clusters of pieces that are constantly connected that act as constant starting points in the process of global assembly.

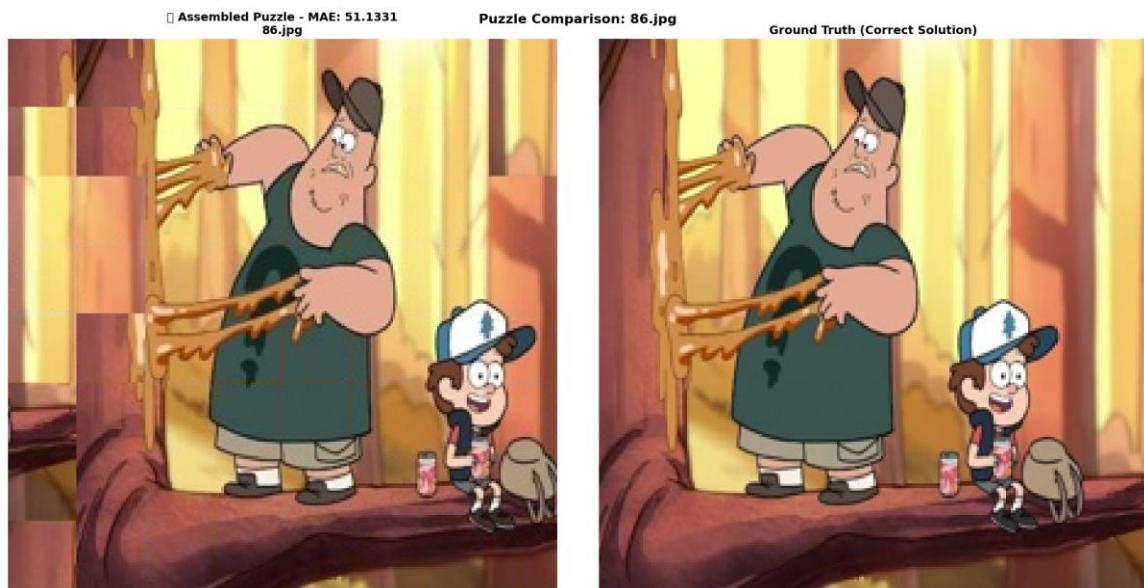
The last step is the final assembly, which consists in inserting the biggest identified cluster on the puzzle grid and in filling the rest of the spaces according to the compatibility of the group with the already placed neighbors. Each of the empty positions is shortlisted using the highest ranked matches, and is then evaluated by the average of the compatibility scores between the candidate piece and the other pieces. In order to reduce sensitivity to initial placement errors, several randomized restarts are done, and the most successful one is chosen. An image that has been rebuilt is compared to the ground truth based on a mean error between the pixels, and a puzzle is said to be solved when the error is lower than a predetermined value. The system performs reasonably well on the 8x8 puzzle dataset with a robust preprocessing, boundary matching and clustering heuristics which give an overall accuracy of about 15 percent demonstrating the difficulty of large-scale jigsaw puzzle reconstruction by classical computer vision methods only.

4.Outputs

Successful Tries:



Failed Tries:



Conclusion

A full classical computer vision system to analyse and assemble jigsaw puzzles was designed and tested in this project. The suggested method effectively combines image preprocessing, noisy image, outlines extraction, perceptual color representation, and edge compatibility analysis to find possible matches between pieces of the puzzle. A hybrid of gradient and color based boundary information, the system is resistant to noise, different orientations, and the stylised nature of cartoon images.

The two-milestone design allowed a coherent process of development, which starts with data preparation and improvement proceeding through edge matching and partial or complete puzzle construction. The findings indicate that classical image processing and geometry analysis can be used to approach complex and more visual matching problems designed with proper thought. The project demonstrates the usefulness of the elementary computer vision tool in addressing real-world issues and offers meaningful information about the issues and constraints of puzzle reconstruction without the assistance of learning-based techniques.