KNN-Based and Gradient-Based Jigsaw Puzzle Solver

A Comparative Study of Puzzle Assembly Strategies

Farzana S Shaikh Haris Jamal

International Institute of Information Technology, Hyderabad

December 1, 2025

Problem Overview

- Task: Reconstruct an image from shuffled jigsaw pieces.
- 8x8 square puzzle pieces; placement unknown.
- Core challenge: determining piece adjacencies reliably.

Initial Approach: KNN-Based Solver

- Idea: use visual similarity between piece boundaries to estimate how likely two pieces are neighbors.
- Extract rich visual descriptors for each piece:
  - Boundary colour and gradient statistics
  - Local texture via LBP histogram
  - Global patch-level RGB statistics and dominant colours (K-Means)
- Store examples of *true* and *random* piece pairs.
- Build a **Nearest Neighbours index** over these pair features:
  - Not a classifier — no parameters are learned
  - Acts as a fast lookup table of known adjacency examples
  - Given a candidate pair, we find the most similar stored pairs and estimate a compatibility score from their labels
- Assembly is guided by placing pieces that achieve the highest compatibility with already placed neighbours.

Assembly Initialization Strategies

- Assembly was sensitive to starting conditions.
- Explored strategies:
  - Random
  - Center-fixed
  - Boundary-fixed
- Boundary-fixed gave noticeable improvement.

Incorporating Random-Fixed Strategy

- Literature showed success by fixing a few known pieces.
- We experimented with fixing:

$$3,\ 6,\ 9 \text{ random pieces}$$

- Allowed solver to build around stable anchors.
- Result: improvement, but still not satisfactory.

A Better Direction from Literature

- Found a CVPR paper addressing similar puzzle constraints.
- Introduced concepts:
  - Gradient-based boundary prediction
  - Directional dissimilarity
  - Best-buddy adjacency constraints
- Inspired redesign of our solver.

Paikin–Tal Solver: Technical Foundations

- Convert each piece to **LAB colour space** for perceptual accuracy.
- For a direction r, predict expected neighbour boundary:

$$\text{pred}_i(r) = 2 \cdot \text{bound}_i(r) - \text{bound}_i^{\text{inner}}(r)$$

- Compute directional dissimilarity using L1 distance:

$$D(i, j, r) = \|\text{pred}_i(r) - \text{bound}_j(r^{-1})\|_1$$

- Convert dissimilarity into a compatibility score:

$$C(i, j, r) = \max\left(0,\ 1 - \frac{D(i, j, r)}{D(i, j_2, r)}\right)$$

where j_2 is the second-best candidate in direction r.

Low D and high C means piece j is a strong, reliable neighbour of i.

Paikin–Tal Solver: Assembly Steps

- Compute directional compatibilities for all piece pairs.
- Identify **best buddies**: two pieces that select each other as their top match.
- Choose a high-confidence seed piece (one having strong best buddies on multiple sides).
- Iteratively expand the puzzle:
  - attach the neighbour with highest mutual compatibility
  - propagate placements from confident regions outward
- No learning, no randomness: the solver is deterministic and parameter-free.

Builds the puzzle by growing reliable clusters instead of guessing placements.

Quantitative Results

Method	**Direct**	**Neighbour**	**SSIM**
KNN (Best: Boundary-fixed)	0.65	0.55	0.78
Paikin–Tal Solver	**0.81**	**0.92**	**0.90**

Paikin–Tal method exhibits a major performance leap.

Side-by-Side Comparison

Original

KNN

Paikin–Tal

Conclusion

- KNN-based model requires strong constraints to perform well.
- Random-fixed initialization made incremental improvement.
- Paikin–Tal gradient method:
  - deterministic, parameter-free
  - robust adjacency estimation
  - best performance overall

Thank You!