Great Daxinzhuang Pottery Puzzle Challenge: Technical Report

Participant: Simon Njoroge Makumi

Solution Methodology

This solution provides a complete pipeline to automatically find matching pottery sherds by comparing their fracture edges. The process is as follows:

1. Pre-processing & Segmentation We used OpenCV for robust image segmentation. First, Otsu's thresholding was applied to each image to create a binary mask, separating the foreground (sherd and scale bar) from the background. By comparing the vertical positions of the two largest contours, we reliably identified and isolated the sherd from the scale bar in all images.

2. Feature Extraction (Deep Metric Learning) The core of our solution is a Siamese-style Convolutional Neural Network (CNN) trained to create feature vectors (embeddings) from edge patches. The model was trained with a Triplet Margin Loss function. Its goal is to take a 64x64 pixel patch of a sherd's edge and output a 128-dimension vector.

3. Self-Supervised Training Data To train the model without labeled matches, we generated 50,000 self-supervised (Anchor, Positive, Negative) triplets.

- **Anchor & Positive:** Two nearby edge patches from the *same* sherd.
- **Negative:** An edge patch from a *different* sherd. Data augmentation (flips, color jitter) was applied to create a more robust model.

4. Matching & Scoring With the trained model, we pre-computed the embedding vectors for every edge point on every sherd in the dataset. To find matches for a given sherd, we:

- **Filtered** the search space to only include other sherds from the same excavation unit.
- Used a scipy.spatial.KDTree to rapidly find the nearest-neighbor (best match) for all edge points between two sherds.
- The minimum distance found is the final "match score" for that pair.

5. Final Assembly A networkx graph was constructed where each sherd is a node. An edge was added between any two sherds with a match score **less than 0.3** (our confidence threshold). The final AssemblyGroups submitted are the "connected components" of this graph, representing all sherds that are directly or indirectly linked together.