

Kaggle Image Matching Challenge 2025 – Project Report

Competition Introduction:

The Kaggle Image Matching Challenge 2025 (IMC 2025) is a computer vision competition that focused on reconstructing 3D scenes from 2D images through Structure-from-Motion (SfM). The goal is to estimate camera positions—both rotations and translations—of images taken in various settings like landmarks or urban areas. This process helps to create accurate 3D models while addressing challenges such as illumination variation and occlusion. The evaluation uses mean Average Accuracy (mAA) to measure pose accuracy against ground-truth thresholds.

Method & Implementation:

The implementation was refined from the official competition baseline. For my model architecture, I installed offline dependencies (Kornia, LightGlue, PyCOLMAP) and copied model weights for ALIKED and LightGlue, adhering to competition rules. My pipeline is structured in four main stages: global retrieval, local feature extraction, feature matching, and geometric verification, executed via utility functions and a main loop that processes datasets and outputs poses.

For the global retrieval part, I used DINOv2, a self-supervised vision transformer, to compute image embeddings, shortlist likely matching pairs via L2 distance on normalized embeddings. For the local feature extraction, I relied on ALIKED to detect up to 8192 key points per image, which is designed to be invariant to scale, rotation, and lighting changes. Its working principle is employing a lightweight convolutional neural network architecture with Differentiable Key Point Detection Module and Sparse Deformable Descriptor Head Module. The former Module is to create a score map and refine key point locations, and the later module uses adaptive sampling at key points to generate compact and robust descriptors. For matching features, I employed LightGlue, a transformer-based model, to filter and refine the high-confidence matches. PyCOLMAP (Classical inbuild 3D reconstruction tool)

handled SfM process, applying RANSAC to remove outliers and incrementally estimate camera poses and form 3D point clusters for scene reconstruction. Finally, after multiple trial, I selected key parameters including 8192 for features, a resize dimension of 1536, and a similarity threshold of 0.12 to maintain the balance between accuracy and efficiency while processing on GPU.

Optimization Considered:

For optimization approaches I investigated, one idea was using GIMLightGlue for those precise yet sparser matches. This model operates on advanced geometric invariance within the transformer framework, focusing on high-accuracy correspondences to reduce false positives that could disrupt the SfM reconstruction. Additionally, I tested CLIP embedding for candidate pair filtering, with a cosine similarity threshold of 0.76. Its feature in training on image–text pairs enable it to capture semantic nuances and perform scene segmentation more effectively than DINO, avoiding mismatches in visually similar environments like staircases by distinguishing contextual similarities more effectively. Moreover, I experimented with clustering match points via DBSCAN to filter out spatial noise and focus re-matching around dense regions. Additionally, I attempted incorporate loop-checking to enforce global geometric consistency by detecting closed image loops and removing pairs with loop errors exceeding 30, reducing reconstruction drift and misaligned clusters. I also explored applying orientation check method to predict and fix image rotations applied a 0.9 confidence threshold to reduce false positive by accepting only high-confidence predictions and reduce initial alignment errors. I also tried using the RDD feature detector at two input resolutions (1024 and 1280), each extracting 8192 keypoints, to enhance scale invariance and capture both fine textures and broader structures missed by single-scale ALIKED. This approach aimed to provide richer, more robust inputs for LightGlue matching and COLMAP reconstruction but increased computation and memory usage.

