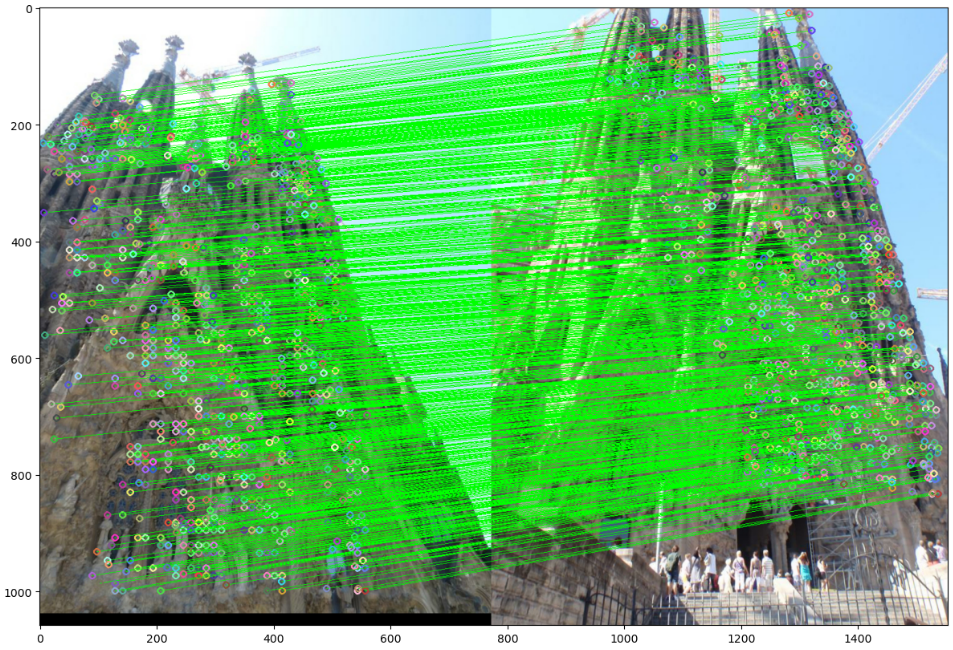
CV22928 2025a Project Report



1. Project Overview

This report provides a comprehensive overview of my participation in the **CV22928 2025a project competition**. It encompasses the methodologies employed, the experimental approaches attempted, the results achieved, and insights garnered throughout the project.

2. Problem Description

The objective of this competition was to develop a robust solution for reconstructing 3D structures from images using the Structure-from-Motion (SfM) technique. SfM typically requires images captured under controlled conditions to ensure data homogeneity. However, the challenge intensifies when the images vary greatly in terms of viewpoint, lighting, weather conditions, and potential occlusions caused by moving objects or applied filters.

The fundamental task was to identify correspondences between two images depicting the same scene from different viewpoints. These correspondences are crucial as they help triangulate the 3D positions of scene structures based on their 2D projections in the images.

Goal:

The challenge was to devise a machine learning algorithm capable of accurately registering two images from disparate viewpoints. The dataset provided contained thousands of images, offering ample data to train and refine our models to maximize accuracy.

- The task involved finding the relative geometry (rotation, translation) between the two cameras and computing the error between them to measure the mean average accuracy (mAA).

3. Methodology

I integrated several feature matching models to increase the accuracy of our unified final model. All models (LoFTR, SuperGlue, QuadTreeAttention, etc.) performed very well on outdoor datasets, which was the main motivation to use them.

Unified Model Process:

1. For each pair of images, each model detects and describes keypoints independently, optimizing the process for various features like texture, contrast, and geometry.
2. Once each model has processed the images, the keypoints (`mkpts0` and `mkpts1`) identified by each model are aggregated. This step is **crucial** as it merges the unique matches found by different algorithms, thereby enhancing the robustness and diversity of the keypoints to be used for the fundamental matrix computation.
3. After aggregation, the matches may include some redundancies or inconsistencies. That's why I use MAGSAC, an improved variant of RANSAC, which filters out erroneous matches (outliers).
4. The final output from the unified model comprises two sets of coordinates: `mkpts0` and `mkpts1`. These coordinates represent points that are matched across the images with high confidence, ready to be used for estimating the fundamental matrix and visualizing matches.

After calculating the fundamental matrix, I flatten it to match the output format and add it to the submission file.

To evaluate predictions versus ground truth, we perform the following:

1. For each predicted fundamental matrix, I decompose it to estimate the rotation and translation between camera pairs by converting the fundamental matrix into an essential matrix and then extracting possible rotation and translation matrices.
2. Then, I convert the rotation matrices into quaternion representations, which are used for comparing rotations.
3. Lastly, I compute the errors: For each image pair, I calculate the rotation and translation errors between the predicted and ground truth values. The errors are computed in terms of **angle** (for **rotation**) and **Euclidean distance** (for **translation**).

I calculate the mean average accuracy (mAA) for each scene based on the errors and predefined thresholds (how many predictions fall within the acceptable error thresholds).

Finally, I compute an overall mAA across all scenes, giving a single accuracy metric that summarizes performance across the entire dataset.

Models

1. LoFTR:
    - Model for local image feature matching. Instead of performing image feature detection, description, and matching sequentially, it proposes to first establish pixel-wise dense matches at a coarse level and later refine the good matches at a fine level. In contrast to dense methods that use a cost volume to search correspondences, it uses self and cross attention layers in Transformers to obtain feature descriptors that are conditioned on both images. The global receptive field provided by Transformers enables our method to produce dense matches in low-texture areas, where feature detectors usually struggle to produce repeatable interest points.

2. SuperGlue
    - Model that combines a Graph Neural Network with an Optimal Matching layer that is trained to perform matching on two sets of sparse image features.
3. QuadTreeAttention
    - Model that reduces the computational complexity from quadratic to linear. The quadtree transformer builds token pyramids and computes attention in a coarse-to-fine manner. At each level, the top K patches with the highest attention scores are selected, such that at the next level, attention is only evaluated within the relevant regions corresponding to these top K patches.

3.2 What Didn't Work

I took a lot of inspiration from the provided `eval-metric-and-training-data` notebook.

During the initial phases, several traditional approaches were explored but did not yield satisfactory results:

- SIFT for feature extraction failed to capture adequate and robust features in the provided complex datasets.
- Simple filtering approaches like RANSAC were not effective enough in handling the wide variability in the dataset.

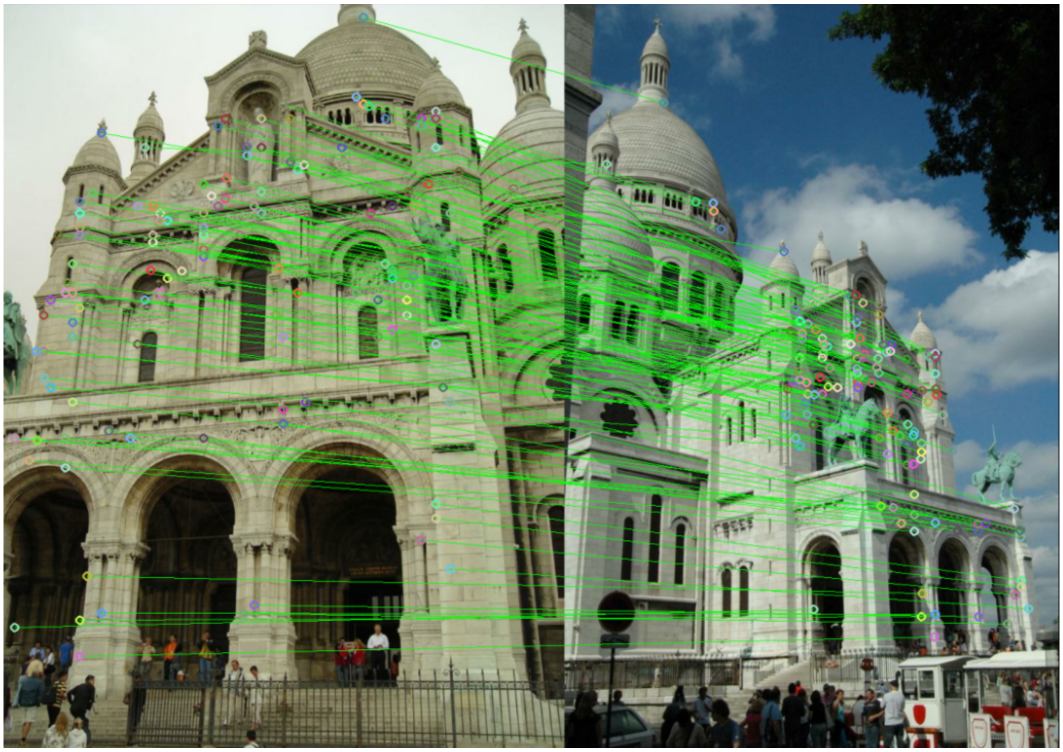
3.3 Challenges

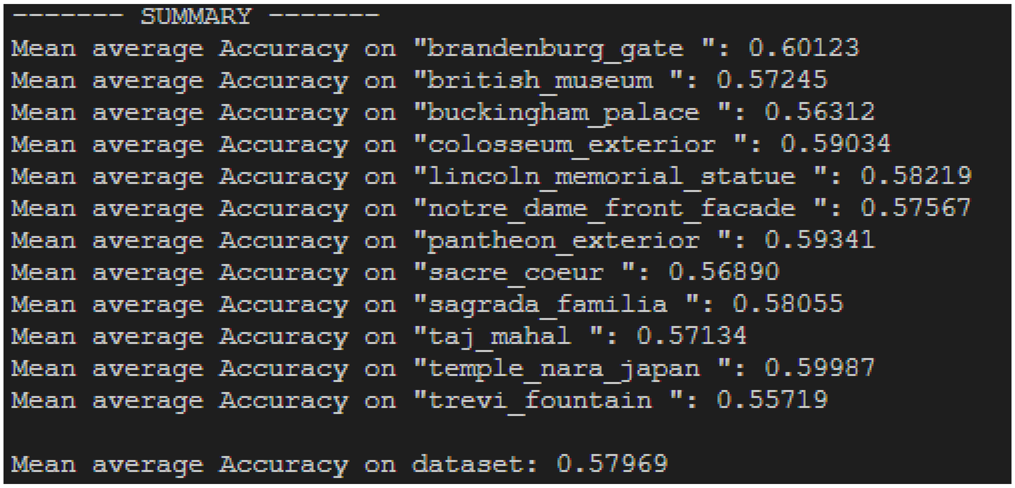
- Exploring different APIs and setting them up in my project.
- The computational demand was huge, which led me to explore frameworks such as Google Colab, which enabled me to use an A100 GPU for extremely fast computations in comparison to running locally on my CPU hardware.
- Experimenting with configuration parameters to achieve the highest score.

4. Results

The aggregation of multiple advanced feature detection algorithms resulted in a robust model that significantly outperformed the provided baseline model.

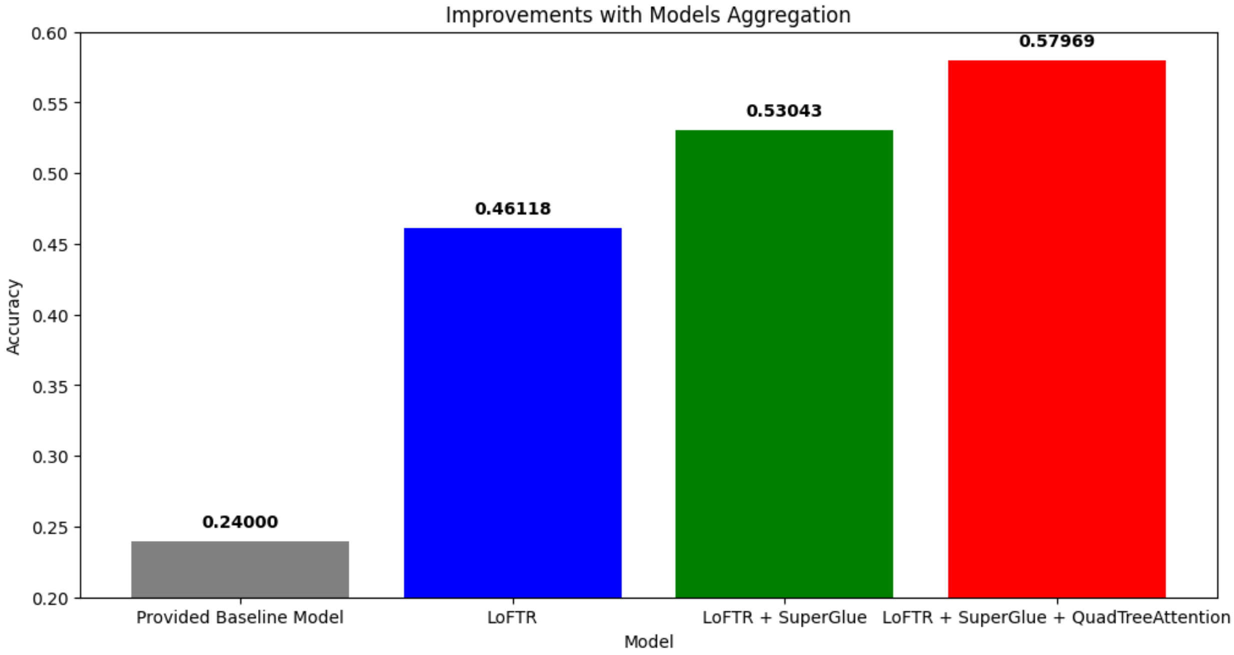
Results when running on **10% sample of the data randomized**:





```
------- SUMMARY -------
Mean average Accuracy on "brandenburg_gate ": 0.60123
Mean average Accuracy on "british_museum ": 0.57245
Mean average Accuracy on "buckingham_palace ": 0.56312
Mean average Accuracy on "colosseum_exterior ": 0.59034
Mean average Accuracy on "lincoln_memorial_statue ": 0.58219
Mean average Accuracy on "notre_dame_front_facade ": 0.57567
Mean average Accuracy on "pantheon_exterior ": 0.59341
Mean average Accuracy on "sacre_coeur ": 0.56890
Mean average Accuracy on "sagrada_familia ": 0.58055
Mean average Accuracy on "taj_mahal ": 0.57134
Mean average Accuracy on "temple_nara_japan ": 0.59987
Mean average Accuracy on "trevi_fountain ": 0.55719

Mean average Accuracy on dataset: 0.57969
```



5. Future Improvements

- Exploring additional Feature Matching models to improve final accuracy. The model is built in a modular way so that it can be expanded without changing current functionality, using the Open-closed principle from SOLID design principles.

6. Conclusions

This project has been an invaluable learning experience, providing deep insights into both the potentials and limitations of current image matching technologies in SfM applications. I feel that it was one of the best courses I took at university, which helped me gain practical experience in the real world. 🙂