Potential improvements

Due to a severe lack of time, I didn't have time to finish many things. The first thing that could have been done is to use more images (channels), since satellite imagery has multi-channel images. This could have improved the image matching, as the satellite images were taken at different times of the year, cloudy, with shadows, so images from different channels could have yielded more key points and matches. From this, it would be possible to somehow average or collect all the matches and display the matches from all channels on the main (TCI) image

I'd also like to test other algorithms and models that I didn't have time to do. There are really a lot of approaches and modifications in deep learning, so I still need to find a suitable algorithm or model specifically for satellite images. Some of another models I found were: XFeat, R2D2, D2-Net.

I also didn't train the model for a specific task - matching satellite images - but used already trained models. Therefore, I think it would be good to work on the data a little bit, to make a dataset to train the model for finding image matches, all on a specific task. This could improve the results quite a bit.

Another problem that hindered me a bit was the lack of memory when loading satellite images into keypoints detection and matching. It would have required better data processing, since satellite images take up a lot of memory, it is difficult to work with them without good processing. In the assignment, I reduced the image size, but I think if I had time, I could have come up with a better processing.

I found LightGlue model and it was interesting for me how it will handle with satellite images (because with normal and difficult ones it handle well). LightGlue is designed to handle challenging image matching conditions like occlusions, viewpoint changes, and geometric transformations which are common in satellite image matching. Also LightGlue offers pre-trained models which can be trained on specific dataset after. However, LightGlue shows not good results as I was expecting.

LoFTR uses transformers for local feature matching. It is robust to geometric distortions and satellite images often affected by geometric distortions like rotation, scale changes and perspective variations. Also satellite images are typically high-resolution, which can lead to challenges in finding correspondences at different scales or in cluttered environments. LoFTR ability to match features efficiently at multiple scales and its attention mechanism can deal with this problem. And LoFTR indeed showed much better results in satellite image matching than SIFT+LightGlue