

1. Error Handling and Logging

- Add checks and exception handling for missing files or directories, especially in methods like ``load_from_disk``, ``os.listdir``, and when reading images. This would prevent crashes if expected files or directories are absent.
- Ensure that the model loading (especially when dealing with directories or files like ``./ner_model``) has error handling in place, in case the directory or files are corrupted or absent.
- Handle cases where the images might not be compatible (e.g., not in the expected format, corrupted, etc.).
- Replace ``print`` statements with a more sophisticated logging mechanism
- For longer-running operations like image processing, add progress bars or something like this.

2. Optimization

- Parallelize image loading and processing

3. Improved Visualization

- Add better visualizations for keypoint matches to allow inspecting the images and matches more easily. Also add more numeral data about keypoint matches to image.

4. Modularization and Clean Code

- Some methods, especially in ``ImageMatcher``, could be refactored, for easier usage and understanding.
- Externalizing configurable parameters (e.g., image directory, max image count, model path) into a configuration file so the code is more adaptable and reusable.

5. Cross-Platform Compatibility

- Creating a Docker container for the entire environment, making it easier to deploy the model and codebase in different environments.

6. Model Update

- Add the ability to fine-tune the LoFTR model or replace it with more advanced models.