Improvements

Imatch: Satellite Image Matching Project

Currently project availables to use two pretrained baseline models – LoFTR and LightGlue, which give good matching results. Also, LoFTR performs well with images from different seasons. Additionally, there is an experiment framrwork for training SuperGlue descriptor on given images and evaluate the performance with LightGlue.

The problems encountered during creation and ideas for their solution are:

1. Expand and improve data

Current issues are:

- Small dataset – the dataset is small and after splitting contains maximum of 40 images for traning and 10 for validation
- Undiversified image region – there is only two tiles XA and YA from a wide region, which may lead to overfitting the model and making it detect very specific keypoints
- Keypoint noise – keypoints from geojson polygon centroids may not represent optimal features for matching.

For this reasons the creation of large, diverse and clean data with valid keypoints, or with reliable methods of their extraction from polygons can improve model performance. The dataset may be synthetic, but must depict landscape features.

2. Improve preprocessing

The problems here are:

- Augmentations – current image transformations lack the diversity needed to simulate real-world satellite variance
- Preprocessing – standard resizing and grayscale conversion discard potentially critical details

These problems also impact inference with baseline models such as LoFTR and LightGlue. The solution will be to create better image (and keypoint) preprocessing pipeline, with satellite-specific augmentations and processing.

3. Improve training objectives

The main idea was to train/fine-tune the SuperPoint model to detect distinctive keypoints and generate robust descriptors specifically for satellite imagery of the target regions.

During project creation several problems were discovered:

- Self-matching – the idea was to teach the model to produce consistent keypoints and descriptors for the same geographic area under different augmentations. However, this currently leads to overfitting. The model fails to learn generally useful features, instead memorizing the training images and producing keypoints that are not informative.
- Polygon Keypoint – the idea was to use keypoints derived from GeoJSON polygons outlining deforestation as ground truth, under the assumption that these shapes denote meaningful areas. In practice, the model struggles to converge with this data, resulting in poor metrics.

The solution will be to combine this approaches or to determine a better objective for training model for such task on this data. Also it is necessary to determine better loss computation approach and inspect behaviour of the model with the change of hyperparamethers.

Conclusion

The main directions for improvement are creating a larger and more diverse dataset, implementing a task-suited augmentation pipeline, and developing more sophisticated training objectives.

NER: Mountain Entity Recognition Project

Currently project availables train and inference DistilBERT model on sentences to recognize mountain names.

The problems encountered during creation and ideas for their solution are:

1. Expand and improve data

The main problems here are:

- Synthetic data – the sentences are generated by Mistral and can create unrealistic patterns and incorrect entities (for example lake names etc.)
- Dataset size and diversity – dataset may be too small for the task and use limited range of mountain names
- Entity distribution – certain mountain names are over-represented, while others are rare or absent

The solution is to:

- Implement a data validation layer to filter LLM-generated sentences for correctness and naturalness.
- Use and annotate real-world data (for example from Wikipedia)

- Expand the mountain name list significantly and apply sampling (in the project there is a txt file with ~98 most common mountain names used for Mistral parsing handler, in case of using this approach the list must be bigger)

2. Improve model training

The main ideas here are:

- Use bigger model – DistilBERT was selected for its computational efficiency and faster training, making it suitable for resource-constrained environments, but bigger models like BERT-base or RoBERTa can give better performance
- Optimize parameters – hypertune base paramethers for better performance

Conclusion

Better performance can be achieved by transitioning from a synthetic, small-scale dataset to a mixed-data source, and by upgrading the model architecture.

General project

There are several code quality and organization tasks I wanted to get to, but ran out of time for:

- Better Logging - add more detailed logging, especially for the training and validation loops
- Ensure all functions and classes are described propperly and all type hints are set
- Refactor Imatch code by creating base classes for the models and trainers
- Integrate Neptune.ai to track my training experiments