Visual Geo-Localization

TA: Gabriele Berton, Gabriele Trivigno ()

OVERVIEW

Visual geo-localization (VG) is the task of coarsely finding the geographical position where a given photograph was taken. This task is commonly addressed as an image retrieval problem: given an unseen image to be localized (query), it is matched against a database of geo-tagged images that represent the known world. The N-top retrieved images, with their geo-tag (typically GPS coordinates), provide the hypothesis for the query's geographical location.

The retrieval is performed as a k Nearest Neighbour search in a learned embedding space that well represents the visual similarity of places. Namely, each image is passed through a network composed of a feature extraction backbone and a head that aggregates or pools the features to create a global descriptor of the image. By using a contrastive learning approach. The model learns to extract descriptors that are discriminative of locations. The similarity search is then implemented as a pairwise comparison among descriptors, e.g. using a cosine similarity.

In this project, you will become familiar with the task of visual geo-localization and with how a visual geo-localization system works. You will start by implementing a baseline and by analyzing the variations that occur by modifying the value of the parameters specific to this framework. With the knowledge gained in these first steps, you will finally propose and implement a variation of the baseline. You will have the freedom to propose your own variation, either to improve the system or to address some problems that you have identified when working with the baseline. However, you will also find some references below as examples and as sources of inspiration.

GOALS

1. Get acquainted with the field of Visual Geo-Localization; understand similarities and differences with Image/Landmark Retrieval
2. Implement two of the most popular methods in the literature, GeM [2] and NetVLAD [1], and train them on the Pitts30k dataset [1].

3. Propose your own variation to improve the system or to overcome some of the challenges encountered in the previous step.

STEPS

1. **Study the literature and get familiar with the task**
   As a preliminary step to get familiar with the task, start by reading about the most popular methods in literature, NetVLAD [1] and GeM [2], which serve as the basis for most of the recent architectures. You can also refer to this survey [3] for a broader overview of the task and its evolution through the years.
2. **Implement NetVLAD[1] and GeM [2] as Baselines**
   Once you are familiar with the theory of visual geo-localization, you will need to implement two methods as baselines. You will use a backbone pre-trained on ImageNet (we suggest a ResNet-18 or VGG-16) and implement as head both the NetVLAD layer [1] and the GeM pooling [2].
   You will need to train both baselines using a triplet loss as shown in [1] and using the Pitts30k dataset [1]. Before getting started with the code, make sure to understand how the dataset is built, how dense it is, which labels are available, plot some diagram with the distribution of the labels and visualize some images.
   To get started, you will be provided with a repository that will give you access to the dataset as well as to a skeleton of the project (including the mining procedure that is required to train the model, as explained in [1], and some details about the choice of hyperparameters). This repository will become available in the next few days.

   As additional resources we suggest to check these repositories:
   - https://github.com/filipradenovic/cnnimageretrieval-pytorch
   - https://github.com/Nanne/pytorch-NetVlad

   At test time, the query image is deemed correctly localized if at least one of the top N retrieved database images is within d = 25 meters from the ground truth position of the query. To assess these baselines, you will have to report the values of recall@N (for N=1,5,10), that is the percentage of correctly recognized queries (Recall) using the N top retrieved images.

3. Ablation Study

The aim of the ablation studies is to understand and verify the effect of different hyperparameters or engineering choices.

1) Try different learning rates and optimizers
2) Test the model trained on Pitts30k also on a different dataset, such as St Lucia
3) Change the parameter which defines the distance at which positives are taken at train time, which in the code is called "train_positives_dist_threshold" and is currently set at 10 meters. How do you expect it to change the behaviour of the model? Did the experiments confirm your prediction?
4) Change the parameter which defines the distance at which positives are taken at test time, which in the code is called "val_positives_dist_threshold" and is currently set at 25 meters. Note that you don't have to train a model again, you can (and should) use the same model for this experiment. How do you expect the results to change? Did the experiments confirm your prediction?
5) Try different data augmentation techniques of your choice, and see how they change the results. Also test models trained with data augmentation on the St Lucia dataset.
6) Try to change the size of the images, both at train and test time. Most datasets use images with resolution 480x640, do you think using bigger/smaller images would improve the results? Validate your hypothesis with some experiments.

4. Add personal contribution

It is now time for you to put in practice what you learned in the previous steps and propose an additional study or an improvement of one of the baselines. The improvement does not need to bring higher results of recall@N, but it may also refer to practical aspects of the system or address some other limitation of the system. Feel free to take inspiration from the examples of variations presented below as well as from the listed referencences.

5. Deliverables

To conclude the project you will need to:
- Deliver PyTorch scripts for all the required steps.

- Write a complete pdf report. The report should contain a brief introduction, a related works section, a methodological section for describing the algorithms you are going to use, an experimental section with all the results and discussions. End the report with a brief conclusion.

EXAMPLES OF VARIATIONS

a. Try with different loss functions

All the recent state-of-the-art architectures for visual geo-localization are based on contrastive learning. The most common loss used is the triplet loss from [1], but there are many possible variations of this loss. For example, you can look at the variations proposed in [4,5,6] and try to implement them.

b. Change the backbone/aggregation architecture

Although the basic architecture of the networks used in visual geo-localization are rather simple (CNN backbone + aggregation head), you can explore different convolutional backbones, or truncate them at different layers. To go one step further, you can move to a backbone based on Transformer networks for images [7,8], or directly use the CLS token [9] without further aggregation.

c. Add attention

A picture of a place, for example in a busy street of a city, is usually crowded with elements that are not informative about the location (e.g. cars, pedestrians). Several architectures [10,11] have demonstrated that attention modules can be used to focus the model on the most informative parts of the scene. You can experiment with inserting these modules into your architecture or also take inspiration from the vaster literature on attention blocks [12,13].

d. Modify the mining procedure

Mining, that is the process of selecting the positive and negative examples used during training [1], greatly affects how the training is performed. You can try to improve the selection of images, or make the mining in general more efficient by taking a smaller pool of samples [14].

QUESTIONS YOU SHOULD BE ABLE TO ANSWER IN THE END

- What is contrastive learning and how it relates to image retrieval ?
- What is the meaning of the ‘mining’ procedure and what are its downsides?
- What is the relationship of NetVLAD and GeM methods with the recall-scalability trade-off?

LITERATURE

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla and J. Sivic, *NetVLAD: CNN architecture for weakly supervised place recognition*, TPAMI 2018

[2] F. Radenovic, G. Tolias, and O. Chum, *Fine-tuning CNN Image Retrieval with No Human Annotation*, TPAMI 2018.

[3] C. Masone and B. Caputo, *A survey on deep visual place recognition*, IEEE Access 2021

[4] L. Liu, H. Li, and Y. Dai, *Stochastic Attraction-Repulsion Embedding for Large Scale Image Localization*, ICCV 2019.

[5] Y. Xia, Y. Xu, S. Li, R. Wang, J. Du, D. Cremers and U. Stilla, *SOE-Net: A Self-Attention and Orientation Encoding Network for Point Cloud Based Place Recognition*, CVPR 2021

[6] T. Ng, V. Balntas, Y. Tian and K. Mikolajczyk, *SOLAR: Second-Order Loss and Attention for Image Retrieval*, ECCV 2020

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, ICLR 2021

[8] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li and H. Shi, *Escaping the Big Data Paradigm with Compact Transformers*, preprint 2021

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of deep bidirectional transformers for language understanding*, In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human LanguageTechnologies

[10] H. J. Kim, E. Dunn, and J.-M. Frahm, *Learned contextual feature reweighting for image geo-localization*, CVPR 2017

[11] G. Berton, V. Paolicelli, C. Masone and B. Caputo, *Adaptive-Attentive Geolocalization From Few Queries: A Hybrid Approach*, WACV 2021

[12] S. Woo, J. Park, J.-Y. Lee and I. S. Kweon, *CBAM: Convolutional block attention module*, ECCV 2018

[13] X. Wang, R. Girshick, A. Gupta and K. He, *Non-local neural networks*, CVPR 2018.

[14] F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang and J. Civera, *Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition*, CVPR 2020