

Project Report

Francesco Blangiardi 288265

Sam 281568

Can Karaçomak 287864

Abstract:

1) Instruction

In this project, we will dig deeper into Visual Geo-Localization task and show the details of image retrieval. Respectively in section 2, we will implement two popular methods NetVLAD and GeM as baselines while using ResNet-18 or VGG-16 (pre-trained on ImageNet) backbone, note their accuracy with using triplet loss and the Pitt30k dataset (a smaller subset of the original dataset) with given hyper-parameters. Secondly, we will exhibit the reaction of baselines with different parameters, dataset (St.Lucia) to acknowledge the effects of hyper-parameters and engineering choices. Then in section 3, we will endeavor to contribute Visual Geo-Localization with our personal knowledge.

2) Implementation

Firstly, the NetVLAD network was implemented and tested. Before giving some numbers to show the performance of NetVLAD, we observed that NetVLAD is considerably heavy to train due to the calculation of positives and negatives for each iteration. Even though Pitt30k was used with GPU-based computation, training took more than several hours.

At the beginning of training, we encountered with low recall rates around 27% (recall rate@5) with the given implementation details which are margin = 0.1, learning rate = 0.0001, number of epoch = 5, momentum = 0.9, weight decay = 0.001. Additionally, we observed longer training times than observed in the paper. Then we tried to overcome these issues with 2 different solutions clustering or checkpoints since we assumed the issue is about the centroids.

Clustering, the idea of a solution was inspired by the “get_cluster” function of the given NetVLAD template. The basic idea of this solution is extracting the centroids of clusters with the usage of mini-batches and storing the results in a cache. After then, it computes k-means using the faiss library. Nevertheless, this attempt has increased recall rates just to around 31% and took the same training times.

Checkpoints, the purpose of checkpoints is to keep the current weights in a repository against random initialization of centroids. While training in the Colab, the weights were lost due to deductions. The implementation of checkpoint solved both the recall rate and long training issues. The recall rate increased to 77.4% in the top 5 samples while the training was taking 21 mins for 1 epoch.

3) Personal Contribution

4) Conclusion