Project Visual Geo-localization - Report

Advanced Machine Learning

Francesco Blangiardi - s288265
Samuel Oreste Abreu - s281568 Can Karaçomak - s287864

2021-12-23

Abstract:

1) Instruction

In this project, the Visual Geo-Localization task was dug deeper and showed the details of image retrieval. Respectively, in section 2, two popular methods were implemented: NetVLAD and GeM as baselines while using ResNet-18 or VGG-16 (pre-trained on ImageNet) as a 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, it was endeavored to contribute Visual Geo-Localization with personal knowledge.

2) Implementation

Firstly, the NetVLAD network was implemented and tested. Before giving some numbers to show its performance the following is worth noting:

- NetVLAD was observed to be considerably heavy to train due to the calculation of positive and negative inputs 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 following implementation details:

- Margin = 0.1
- Learning rate = 0.0001
- Number of epoch = 5
- Momentum = 0.9

- Weight decay = 0.001

Additionally, longer training times than the original paper were observed. Afterwards, in order to overcome these issues2 different solutions were proposed: clustering and checkpoints, since the issue was assumed to revolve around 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