Project Visual Geo-Localization Report

Samuel Oreste Abreu - 281568
Francesco Blangiardi - 288265
Can Karacomak - 287864
Politecnico di Torino
Corso Duca degli Abruzzi, 24, 10129 Torino TO

Abstract

Through the application of machine learning techniques such as CNNs, pooling layers and normalization approaches the features of an image can be extracted. This report [1] relays the implementation and discusses the performance of novel methods for the problem of finding a place given a picture containing the aforementioned features. Firstly, the representation methods are explored with all the details of the mining procedure on image retrieval. How CNNs gained significance, what is the role of metric learning, comparison various design choices, and challenges on image-based geo-localization with structure photos. Secondly, the current developments are studied like improvements with attention layers, new loss functions. Finally, outcomes are exhibited on image retrieval benchmarks.

1. Introduction

Visual Geo-localization is a research field that has garnered attention in recent years due to offering a potential solution of finding a place given an image, especially after Convolutional Neural Networks (CNNs) became dominant in image retrieval with their search efficiency, compactness of representation, discriminative power, and usability even under weakly supervised training. These CNNs methods can be categorized into two families: aggregated and pooled representations. NetVLAD (1), and GeM (2) are the most popular methods in literature from these families, and were also expanded in several ways in later works. This project is aimed at studying the performances of the NetVLAD and GeM layers, with a backbone of a pretrained, frozen ResNet-18 in the task of image retrieval with weakly supervised contrastive learning through the creation of aggregated or pooled image descriptors. Moreover, an improvement of the GeM baseline known as SOLAR (9) is also considered and replicated.

2. Related Works

Backbone and Optimizers: The original literature (1) (2) proposed the usage of CNN architectures based on what was relevant at the time (AlexNet, VGG-19 and various ResNet, in particular ResNet101). The same logic can be applied to the usage of optimizers; since then, a more optimal approach has been introduced (4) whose results are of a much higher quality, leaving behind traditional methods (Such as SGD).

Normalization and Dimensionality Reduction: Considering the scope of the task, some proposals on reducing the memory footprint during training were presented using dimensionality reduction and normalization through whitening and L2-normalization (10), at the same time they were found to produce better results.

It is worth mentioning that due to the nature of the task at hand not many works were researched other than the methods cited above, however, during the ablation step a state-of-the-art model (9) stood out for its simplicity and high performance; it achieves this through the usage of an attention layer, a novel approach to computing the loss and the application of normalization techniques to both the dataset and the output features.

3. Deep Architecture & Learning

In this section, visual geo-localization is dug deeper and the details of image retrieval are exhibited. Firstly, the two most popular representation methods from aggregation and pooled families, NetVLAD (1) and GeM (2) are focused as baselines while using ResNet-18 (pre-trained on ImageNet (5) as a backbone. The loss functions used for training can be found in section (3.3), while in section (3.4) the main parameters of the application are described as well as the optimizers that were used during training. Some considerations about the application of data augmentation techniques

[1] Code: `https://github.com/richter43/project_vg`, Data: `https://mega.nz/folder/8QpVXY7Z#32HnYg1PAVJ9aRhL7I8bMg`

are discussed in section (3.5). Finally, in section (3.6) the improvements that were implemented to the GeM baseline are listed , namely the SOLAR (9) descriptor and whitening.

3.1. NetVLAD

NetVLAD (1) is one of the most popular method in the aggregated representation family in Visual Geo-localization.

The core component of NetVLAD is obtained by plugging the soft-assignment into the VLAD (Vector of Locally Aggregated Descriptors) descriptor as shown in the following equation.

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_k^T x_i + b'_k}} (x_i(j) - c_k(j)), \qquad (1)$$

where w_k, b_k and c_k are sets of trainable parameters for each cluster k. Similarly to the original VLAD descriptor, the NetVLAD layer aggregates the first order statistics of residuals $(x_i - c_k)$ in different parts of the descriptor space weighted by the soft-assignment $a_k(x_i)$ of descriptor x_i to cluster k. All parameters of NetVLAD are learnt for the specific task in an end-to-end manner.

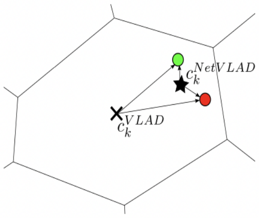

Figure 1. Visual representation of the assignment of a feature in a Voronoi diagram

Fig.(1) demonstrates that the NetVLAD layer can be visualized as a meta-layer which is further decomposed into basic CNN layers connected in a directed acyclic graph. The first term in eq.(1) is a soft-max function $\sigma_k(z) = \frac{exp(z_k)}{\sum^{k'} exp(z_{k'})}$. Therefore, the soft-assigment of the input array of descriptors x_i into K clusters can be seen as a two step process: (i) a convolution with a set of K filters w_k that has spatial support 1×1 and biases b_k, producing the output $s_k(x_i) = w_k^T x_i + b_k$; (ii) the convolution output is then passed through the soft-max function σ_k to obtain the final soft-assignment $a_k(x_i)$ that weights the different terms in the aggregation layer that implements eq. (1). The output after normalization is a (K × D) × 1 descriptor.

3.2. GeM

Generalized mean pooling (GeM) (2) is the current state-of-the-art method of the pooled representation family. This layer implements a parametric generalized mean. Parameters that come from each feature map can be shared, thus reducing the parameters count to one. In short, it is a pooling method that summarizes the convolutional features.

$$f^{(g)} = [f'_1 g ... f_k^g ... f_K^g], f_k^{(g)} = \left(\frac{1}{|X_k|} \sum_{x \in X_k} x^{p_k} \right)^{\frac{1}{p_k}} \qquad (2)$$

Considering X as input, the pooling process produces a vector f as an output. As it can be observed in (2), when $p_k \rightarrow \infty$, it becomes max pooling and for $p_k = 1$ it becomes average pooling. As it can be observed GeM generalizes both max and average pooling and consistently outperforms them. Since the pooling operation is differentiable, the parameter (p_k) can be learned as part of back-propagation. Additionally it is possible to keep just one shared pooling parameter (**p**) instead of **k** different ones, which is the approach taken in this project as it's considered as the best approach in (2)

3.3. Loss functions

The task of weakly supervised contrastive learning implies the comparison of the currently generated descriptor with descriptors generated both from matching and non-matching images, which in this case can be recognized only through their geographical coordinates. All losses used in this project reflect these aspects and rely on the computation (also known as "mining") of a training tuple $(q, |p_i^q|, |n_j^q|)$, where **q** is the current query image, $|p_i^q|$ is the set of all images taken within a certain distance (10 meters by default) from **q** and $|n_j^q|$ is a set of 1000 images taken from viewpoints that are distant more than 25 meters from **q**. Moreover,the distance between two descriptors $d(D_i, D_j)$ is defined as the degree 2 norm of $D_i - D_j$ and we refer to D_q, D_p and D_n as the descriptors generated from q, from the best positive match in $[p_i^q]$ according to descriptor distance, and the 10 hardest negative matches respectively. The Triplet Ranking Loss (or first order loss, used during training of NetVLAD and GeM models) can now be written as

$$L_{TRS} = \frac{1}{|D_n|} \sum_{D_{ni} \in D_n} l(d^2(D_q, D_p) - d^2(D_q, D_{ni}) + m) \qquad (3)$$

where l is the hinge function and m is a margin parameter: hence each term of the summation will have no contribution to the loss if the difference of the two distances squared is less than the margin i.e. the total loss is lower when D_q is very similar to D_p and is very different from each hard negative descriptor D_{n_i}. Additionally, the Second Order

Similarity loss can also be defined as

$$L_{SOS} = \frac{1}{|D_n|} \sqrt{\sum_{D_{ni} \in D_n} (d^2(D_q, D_p) - d^2(D_q, D_{ni}))^2} \tag{4}$$

which combined with L_{TRS} and a balancing parameter λ (10 by default) as

$$L_{SOLAR} = L_{TRS} + \lambda L_{SOS} \tag{5}$$

defines the loss function used during training of the SOLAR models

3.4. Optimizer, Learning Rate and Parameter Choices

3.4.1 Optimizers

According to the reference papers, Stochastic Gradient Descent (SGD) is the choice of NetVLAD (1) while GeM (2) uses SGD with AlexNet (5) and Adam (4) with ResNet (7) and VGG (6). If the significant number of parameters and samples are considered, the Adam optimizer is one of the best choices. Because Adam is computationally efficient, has few memory requirements which is well suited for problems that are large in terms of data and/or parameters. Additionally, the method is appropriate for non-stationary objectives and issues with noisy and/or sparse gradients. Meanwhile, there are two bottlenecks for SGD, the number of parameters and samples. Therefore, SGD and Adam are the optimizers used for exhibiting their effects on the chosen datasets and backbones.

3.4.2 Learning Rates

The learning rate is a parameter that determines the step size of each iteration. A learning rate should let us move towards the minimization of a loss function where the gradient direction is tangent to an isocurve. A big learning rate may cause risk of overshoot while a small rate may slow convergence speed. Implementation details for learning rate can be found in (4.2).

3.4.3 Validation & Training Distance Thresholds

Since the mining procedure and testing relies on weak labelling (Geographic coordinates of the viewpoint where the images are taken), these two parameters are introduced in order to tell images representing different places apart. In detail: the training positive distance threshold (TPDT) is used during mining to extract the set of potential positive images $|p_i^q|$ i.e. the set of images taken in a radius of at most TPDT meters from where the query image was taken; the validation positive distance threshold (VPDT) is instead the distance at which a sample is considered a positive at test time i.e. the result of the query at a given recall value k is considered as a hit if at least one of the k best matches found is actually within VPDT meters from where the query image was taken. Both of these parameters are very important as setting them with inadequate values can both make training less effective and tests less accurate.

3.5. Augmentation

Data augmentation is a technique to increase the amount and diversity of samples by applying various transformations. In this way, while increasing the number of images, there ensues a chance to have various appearance of the same image. If the samples are considered in Pitts30k or another image based geo-localization datasets, some augmentation technique may be beneficial as cropping, brighening, random perspective since image retrieval is a similarity based method. In other words, if the augmentation technique creates new samples rather than variants of the original image such as flipping horizontally (There are no mirrored samples in the test set or in a real-world scenario.) then these techniques may reduce accuracy slightly. Such transformations would be better for a classification task. However, cropping is auspicious to improve accuracy (refref:Sare). Moreover, different conditions are challenging for visual geo-localization like weather conditions, time of day, among others. In such cases data augmentation would be helpful using a few techniques like brightness, grey scale, etc. The results of the application of augmentation techniques can be found in graph 7.

3.6. Personal contributions

3.6.1 Solar

SOLAR is a descriptor that improves the GeM baseline by adding attention and exploiting second order relations between features. This is achieved by passing to the GeM layer a feature map X^{so} that incorporates second-order spatial informations, which is computed by interleaving several **S**econd-**O**rder-**A**ttention blocks (SOA) between the last layers of the CNN backbone. As a first step, each of the SOA block computes three projections of the input map **X** called **q**, **k** and **v** through 1x1 convolution. Then, the second order attention map **z** is computed as

$$z = softmax(aq^T k) \tag{6}$$

where *a* is a scaling factor set to 0.5. Finally, X^{so} map is obtained by

$$X^{so} = X + g(z \times v) \tag{7}$$

where *g* represents another 1x1 convolution. In this way, every element $(x_{i,j}^{so})$ of map X^{so} will be a function of the features from all locations of **X**, and it will be passed as the input of the GeM layer which will compute the solar descriptor as described in (7). Moreover, another contribution taken from the SOLAR paper is the usage of the second-order similarity loss, which is described in section 3.3

3.6.2 Whitening

Whitening consists in decorrelating the different dimensions of a two dimensional matrix by forcing its covariance matrix to be equal to the identity matrix.

$$\begin{aligned} \Sigma &= cov(X) \\ W^T W &= \Sigma^{-1} \\ Y &= WX \end{aligned} \tag{8}$$

Where W is the whitening matrix, Σ is the covariance matrix and Y is the whitened matrix.

This is usually performed on to the entire dataset prior to training, but newer methods (9) apply it to the feature vectors during the normalization step.

The implemented methods in this project are:

- **ZCA whitening through eigendecomposition**:decomposes the covariance matrix in eigenvalues and eigenvectors, however, the decomposition can not be performed if the matrix is singular (Such assumption does not hold in this case).

  $$\begin{aligned} \Sigma &= M \Lambda M^T \\ W_{ZCA} &= M \Lambda^{-1/2} M^T \end{aligned} \tag{9}$$

  Where Λ is a matrix with the eigenvalues in its diagonals and M is the eigenvector matrix.

- **ZCA whitening through singular value decomposition**: decomposes the covariance matrix in its singular values.

  $$\begin{aligned} U, s, V &= svd(\Sigma) \\ W_{ZCA} &= U s^{-1/2} V^T \end{aligned} \tag{10}$$

- **Learnable whitening:** considering the fact that whitening is nothing but a linear transformation, it can also be implemented through means of a learnable linear transformation ((9).

For further information on its usage see (4.2).

4. Experiments

4.1. Datasets and evaluation methodology

The results are reported on two datasets:

Pitts30k: This dataset contains thirty thousand (database) images of the city of Pittsburgh with slight variations in pitch and yaw. The images are divided in equal parts for testing, training and validating purposes, they were not modified and depict an urban setting, characterized mainly by concrete structures, traffic signs and vehicles.

St. Lucia: This is a smaller dataset (Around three thousand images) destined for testing purposes. The images were not modified and depict a small town setting, characterized mainly by trees, vehicles and sporadic wooden buildings.

Both of the datasets contain RGB images of size 640x480px and neither of them contain a radical change in weather or time setting.

Evaluation metrics The image subsets are further divided in two, a database (Which contains the images that will be fetched at comparison time) and a query (Whose location is desired to be known). Once a query image is fed to the model it will return a given amount of ranked images. The evaluation consists of computing after which rank did the model correctly assign the image, usual values for this are 1, 5, 10 and 20. This is defined as ”Recall after N” (R@N). It is worth noting that the standard distance for which an image is considered to be within range is 25 meters.

Training environment All the models were trained using the Colab service offered by Google (Equipped with two Intel Xeon CPUs and one K80 NVIDIA GPU).

4.2. Implementation Details

As baselines, ResNet (7) backbone is used which is extended with NetVLAD(1) and GeM pooling (2) layers. The backbone is cropped at the convolutional layer4, but in the case of SOLAR (9) the backbone was modified by adding a SOA block between layer2 and layer3 and another SOA block before the GeM aggregation. Moreover, in the SOLAR models an optional whitening layer was added after the GeM layer. Since conv5 was removed from the resnet backbone, raw conv4 descriptors (with no normalization) are used for max pooling while an additional descriptor-wise L2-normalization layer is added after conv4 for VLAD. Adam optimizer was used to train the SOLAR models, while training for NetVLAD and GeM was performed with both Adam and SGD optimizers. Training positives distance threshold default value is 10 while validation positives distance threshold is 25. Several Learning rates were considered for NetVLAD (1), GeM, and Solar. Figure 2 shows the validation recall rates extended with NetVLAD and GeM original implementation details in (1, 2).

All models were observed to be considerably heavy to train due to the mining procedure at each iteration. NetVLAD generally required more epochs to train compared to GeM and SOLAR. To avoid overfitting and very long training times, the maximum number of epochs of training was set to 10, while the patience mechanism (no improvement registered for 3 epochs in a row) was set as the main termination mechanism.

In the first attempts, longer training times than the original paper and a considerable increase of recall rate with a learning rate of 10^{-4} were observed. In order to overcome these issues, two different solutions were proposed: clustering and checkpoints.

Clustering: the idea that this could improve the results stemmed from the presence of the ”get_cluster” function in the provided NetVLAD template. The main idea is that, instead of initializing NetVLAD’s centroids at random, they are extracted from a simple pass through the ResNet-18 backbone achieved by sending images in mini-batches and storing the results in a cache. The procedure is concluded with a k-means clustering algorithm with k equal to the desired amount of clusters that will be used in NetVLAD. Moreover, some tests were run by reinitializing the clusters to the initial value and keeping only the conv4 layer of the backbone as contribution of the training phase, and the results were only a few percentiles lower compared to the standard tests.

Checkpoints: the purpose of checkpoints is to avoid losing the progress that has been obtained throughout the training step, thus, continuing from the last epoch from which it was saved. It was implemented in a way that all data (Training state, current numpy seed and the dataset state) and metadata (Epoch number, best R@5 and recalls) is able to be loaded between training sessions. Due to how Colab works the saved state has to be exported before a disconnection occurs, for this reason it is uploaded to a cloud service called ”Mega”. Once a training session is restarted a saved state can be downloaded from the cloud by specifying its path.

4.3. Results & Discussion

4.3.1 NetVLAD and GeM

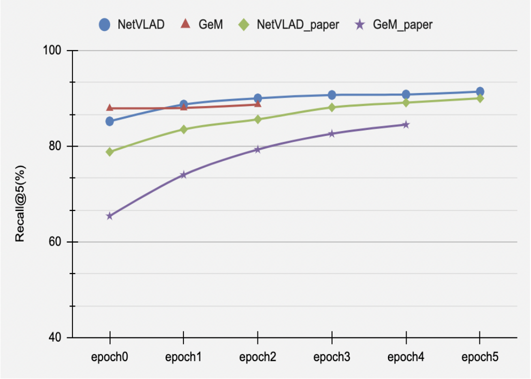

Figure 2. Comparison of recalls and different models through training epochs, including the original paper’s parameters

Comparison with the base models As can be seen in Fig.(2), using the original parameters for training NetVLAD and GeM result in much worse performances.

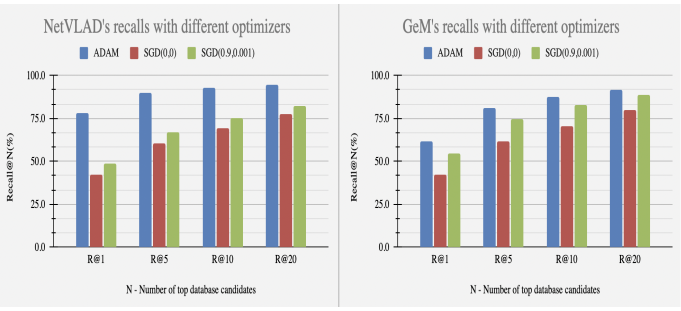

Figure 3. Application of different types of optimizers on both NetVLAD and GeM - Test Pitts30k

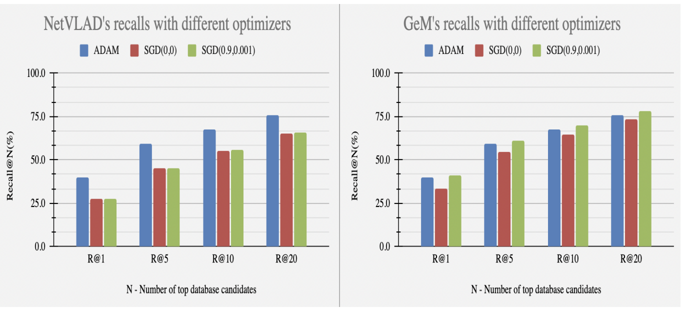

Figure 4. Application of different types of optimizers on both NetVLAD and GeM - Test St.Lucia

Optimizer variation: Considering Fig.(3) and Fig.(4), the Adam optimizer was shown to be considerably better on both types of models on the Pitts30k test. Other observations include: NetVLAD performs better than GeM on Pitts30k (R@5 89.7% vs 81.2%) and St.Lucia (R@5 59.3% vs 39.8%). Astonishingly, GeM’s SGD (Momentum 0.9,Weight decay 0.001) performed better than the Adam version (R@5 61.1% vs 39.8%). (For more information refer to table 1)

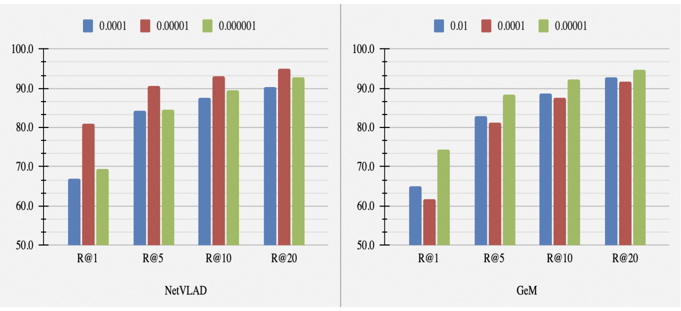

Figure 5. Application of different types of learning rates on both NetVLAD and GeM - Test Pitts30k

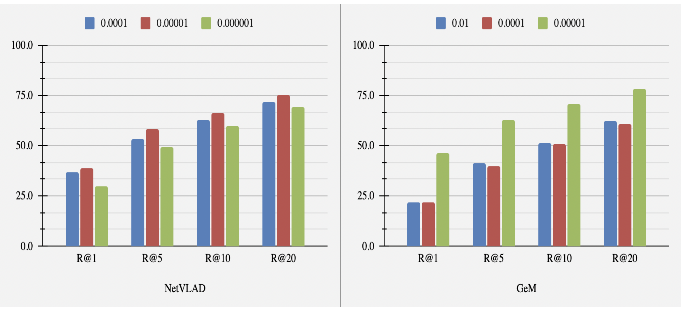

Figure 6. Application of different types of learning rates on both NetVLAD and GeM - Test St.Lucia

Learning rate variation: Referring to Fig.(5) and Fig.(6) Both GeM and NetVLAD were found to train optimally with a learning rate equal to 10^{-5}. In the original papers this value was much lower (10^{-6}). This is a well known trade-off, at the risk of overfitting the model less epochs will be needed to train the model. The original methods had to be trained up to 30 epochs while the models implemented in this report were trained up to 6 epochs without any sign of overfit. Additionally, learning rates 10^{-2}, 10^{-4} were found to overfit quite easily and as a consequence their recall rates at the test set are lower than 10^{-5}. (For more information refer to table 2

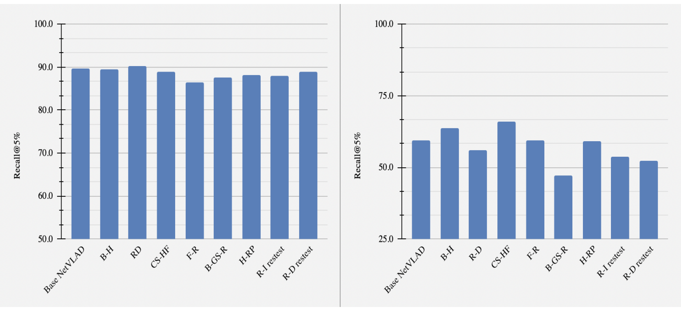

Figure 7. Pitts30k-test and St.Lucia-test results of data augmentation, brightness (B), hue (H), resolution decrease (RD), horizontal flip (HF), contrast saturation (CS), rotate (R), greyscale(GS), random perspective (RP), resolution increase(RI), resolution decrease(RD), random crop (RC).

Data augmentation: For the Pitts30k dataset, recall rates decreased slightly in all augmentations except resolution decrease while ascent is observed in brightness - hue, contrast saturation - horizontal flip. St.Lucia's dataset includes more colourful samples that its Pitts30k counterpart. Because of that, the techniques of brightness, hue, contrast saturation may be beneficial for results on St.Lucia. Most of the augmentation techniques can be ineffective in image retrieval for its trained dataset if the dataset includes the same type of condition. If the data set includes the challenges like different weather conditions or the times of day, the augmentation techniques may be beneficial like brightness, hue, greyscale. For more information refer to table 4.

Train/validation positive distance threshold variation: Some tests were also performed with varying train and validation positive distance threshold. As shown in Fig.(8), increasing or decreasing the train threshold with respect to the default value 10 leads to a small decrement in performances both during training and testing.

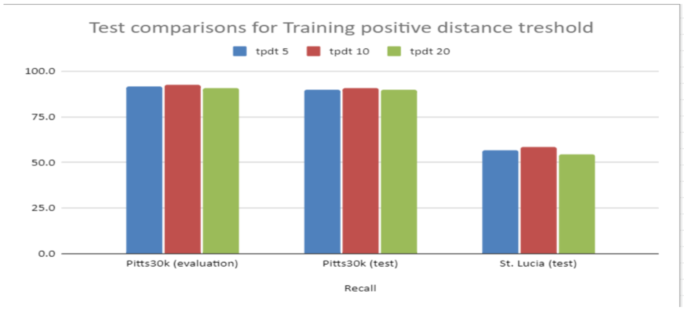

Figure 8. Comparison of R@5 obtained from training, testing on Pitts30k and testing on St. Lucia for NetVLAD models trained with varying train positive distance threshold.

A possible explanation for this result is that by lowering the threshold the mining procedure can collect fewer samples among the soft positive ones, and since Pitts30k is a rather small dataset this may also lead to having a best match sample which is not as structurally similar to the query as possible others which were taken far from the query image (i.e. a picture taken in the same place as the query but facing a different direction); likewise increasing the threshold increases the probability that the best match is actually not depicting the same place as the query. The results for varying validation positive distance threshold are instead shown in Fig.(9). For more information refer to table 3.

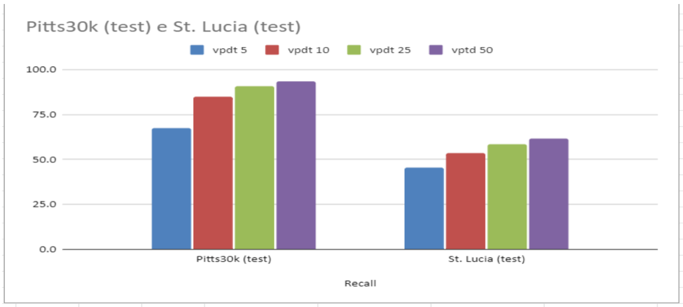

Figure 9. Comparison of R@5 a NetVLAD model with learning rate of 0.00001 and tpdt of 10 and tested on Pitts30k and St. Lucia with varying validation positive distance threshold

As expected increasing the VPDT parameter leads to better results, since it controls how strict the testing procedure is when considering a given match as a hit. An important thing to keep in mind however is that, due to the weakly labelled nature of the data, the probability of considering as a hit a matching is also being increased, that actually depicts a completely different place, so these results should not be blindly trusted.

Image resizing: It was hypothesized that decreasing the size of an image should increase the recall score mainly due to the fact that the features of an image would become so small that it could be exchanged for another feature, however this was not always the case. Whereas some tentative results may have proved this idea (Downsizing to 512x384px, a 80% reduction in size), the improvement is so small that it can not be considered so. Moreover, anything below that size would perform worse than the baseline version, which is counter intuitive to the initial expectations. Refer to table 5. Similarly, increasing the image resolution did not bring to any improvement, expecially in the St. Lucia test experiments, which may be due to the fact that increasing size does not add any additional information but just increases the features to process in an image, which can lead more easily to an overfitting model.

4.3.2 SOLAR

SOLAR models were trained with 3 different learning rates and both with and without whitening, the best performing model was trained without whitening with a learning rate of 0.00001, and trained for a total of 4 epochs. As shown in Fig.(10), the model performed better than GeM in all datasets while keeping the same training time, and was able to outperform both GeM and NetVLAD on St. Lucia, thus proving to be better at generalising features. For more information to table 6.

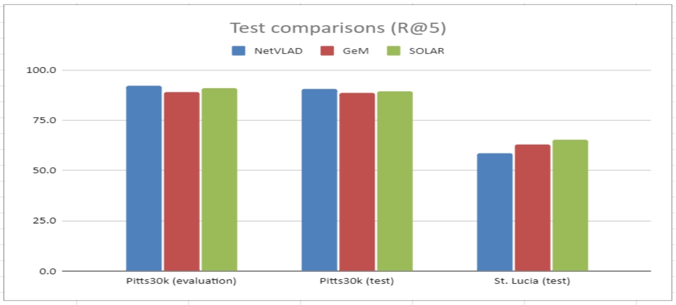

Figure 10. Comparison of performances between the best NetVLAD, GeM and SOLAR models (no augmentation). Among all the models trained, only NetVLAD with CS-HF augmentation was able to outperform SOLAR without whitening

Additionally, SOLAR models containing whitening did not seem to improve the performances with respect to the SOLAR model without whitening, the closest model being the one implementing whitening with a learnable linear transformation as shown in Fig.(11)

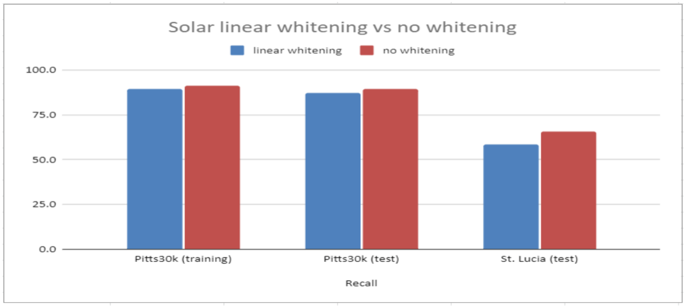

Figure 11. Comparison of performances between Solar models with learnable whitening and without whitening. Adding whitening to the models did not bring to an improvement in the results

5. Conclusions

Visual Geo-localization comprehends a vast area of research, with the state-of-the-art improving at each iteration. In this paper a glance at three of such models was taken, based on an image retrieval approach with contrastive learning and their performances juxtaposed.

NetVLAD layer proved to be the best model for the task of image retrieval for images similar to the training set.The initialization of the centroids is vital to reach good performances, and has has a greater weight than training itself. However the layer needs several epochs to be trained and a strong domain shift with respect to the training set can hinder the test performances noticeably, the only solution being training with several data augmentation techniques. GeM layer represented a good tradeoff between training time and performances, as well as being very easy to implement and being able to generalise better than the NetVLAD layer. SOLAR models instead improve the GeM modles by adding attention and making use of second-order relations through the SOA block and the Second Order Similarity loss function, which proved to be a very good altenative to GeM and NetVLAD both in training time and performances.

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla and J. Sivic, [NetVLAD: CNN architecture for weakly supervised place recognition, CVPR 2016

[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] D. Kingma and J. Ba, Adam: A method for stochastic optimization, ICLR, 2015

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classifi- cation with deep convolutional neural networks, NIPS 2012

[6] K. Simonyan and A. Zisserman, Very deep convolutional net- works for large-scale image recognition, in arXiv:1409.1556, 2014

[7] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, CVPR 2016.

[8] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li, Self-supervising Fine-grained Region Similarities for Large-scale Image Localization, ECCV 2020

[9] Ng, Tony and Balntas, Vassileios and Tian, Yurun and Mikolajczyk, Krystian SOLAR: Second-order loss and attention for image retrieval, ECCV 2020.

[10] Jégou, Hervé ans Chum, Ondrej Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening , ECCV 2012.

6. Appendix

6.1. Tables

Table 1. Baselines recall on test sets with various optimizers.

Model	Dataset	Optimizer	R@1	R@5	R@10	R@20
NetVLAD	Pitts30k	Adam	77.9	89.7	92.7	94.8
		SGD(0,0)	42.3	60.6	69.1	77.4
		SGD(0.9,0.001)	48.5	67.2	75.3	82.4
	St.Lucia	Adam	39.8	59.3	67.3	75.6
		SGD(0,0)	27.7	45.3	55.0	65.0
		SGD(0.9,0.001)	27.5	45.5	55.8	65.6
GeM	Pitts30k	Adam	61.8	81.2	87.5	91.7
		SGD(0,0)	42.4	61.6	70.6	79.7
		SGD(0.9,0.001)	54.3	74.8	82.7	88.6
	St.Lucia	Adam	21.9	39.8	50.8	61.0
		SGD(0,0)	33.2	54.6	64.4	73.4
		SGD(0.9,0.001)	40.8	61.1	69.7	78.3

Table 2. Recalls on test sets with learning rate variations

Model	Dataset	LR	R@1	R@5	R@10	R@200
NetVLAD	Pitts30k	1.00e-4	80.9	90.5	93.4	95.4
		1.00e-6	69.4	84.5	89.4	92.7
	St.Lucia	1.00e-4	37	53.5	63	71.7
		1.00e-6	29.8	49.5	59.9	69.4
GeM	Pitts30k	1.00e-2	65.0	83.0	88.8	92.9
		1.00e-5	74.5	88.5	92.3	94.6
	St.Lucia	1.00e-2	21.8	41.2	51.5	62.1
		1.00e-5	46.1	62.9	71.0	78.2

Table 3. NetVLAD's variations on TPDT and VPDT

Dataset	TPDT	VPDT	R@1	R@5	R@10	R@20
Pitts30k	5	25	77.6	89.2	92.3	94.7
	20	25	77.1	89.0	92.3	94.7
	10	5	32.8	49.0	52.3	54.5
	10	10	63.2	83.1	87.1	89.6
	10	50	81.3	92.6	95.1	96.9
St.Lucia	5	25	37.0	56.1	65.8	74.0
	20	25	33.9	53.5	62.0	69.7
	10	5	21.8	37.8	44.5	51.0
	10	10	35.7	53.3	61.1	67.8
	10	50	41.9	62.9	71.2	80.5

Table 4. Performance of NetVLAD trained on transformed datasets

Dataset	Training transformation	R@1	R@5	R@10	R@20
Pitts30k	Brightness-Hue	79.8	89.9	92.8	95
	Contrast-Saturation-Horizontal flip	75.7	88.9	92.1	94.3
	Horizontal Flip - Rotation	72	86.5	90.8	94.1
	Brightness-Grayscale-Rotation	73	87.6	91.5	94.9
	Hue-RandomPerspective	77.3	89.1	92.7	95
	Resolution increase	75.6	87.9	91.1	94
St.Lucia	Brightness-Hue	46.2	63.8	72.1	78.8
	Contrast-Saturation-Horizontal flip	49.1	66.1	72.9	80.5
	Horizontal Flip - Rotation	30.1	46.1	54.5	63.9
	Brightness-Grayscale-Rotation	30.5	47.3	56.1	65.2
	Hue-RandomPerspective	39.1	59.1	67.8	75
	Resolution increase	31.1	53.8	64	74.1

Table 5. Data resizing and its effects measured in R@5.

NetVLAD	Pitts30k	St.Lucia
Baseline	89.7	59.3
x0.80 unmodified test	90	59.9
x0.80 resized test	89.6	56.8
x0.75 unmodified test	89.3	48.8
x0.75 resized test	86.3	49.4
x1.25 resized test	87.9	53.8

Table 6. Solar and variations on whtening on test datasets

Dataset	Whitening	R@1	R@5	R@10	R@20
Pitts30k	None	77	89.4	92.8	94.9
	Linear	72.2	87.8	92	94.6
	SVD covariance	34.7	53.7	62.4	72.8
St.Lucia	None	46.9	65.4	71.9	79
	Linear	39.4	58.5	67.8	75.8
	SVD covariance	12	30.5	44.1	56.9