Visual Geo-Localization

Matteo Gambino
s287572

Michele Pierro
s287846

Fabio Grillo
s287873

Abstract

In order to predict the location of a query image by retrieving annotated photographs with similar descriptors needs an efficient and reliable generation of those descriptors. In order to accomplish that objective, is fundamental that the network focuses on portion of the various images that contains useful information and at the same time ignore not informative areas like the ones containing elements like cars or pedestrians. For that reason attention layers are fundamental in the proposed network. In addition to that we are comparing state of the art techniques for the visual geo-localization task like GeM [1], NetVLAD [2] and CRN [3]. The code used is publicly available here

1. Introduction

For what concerns geo-localization and its purpose the best results have been achieved through deep learning algorithms that, thanks to the analysis of pixels of a given photograph, could retrieve very accurate information using convolutional neural networks (CNN). Of course image retrieval is not an easy task, it aims at extracting the most interesting features for each image from a large database; then for a query image it has to extract useful features from that too and then find a correspondence between the features of the query photograph and the ones inside the database.

An other relevant aspect on which it is important to keep attention on is the place recognition problem that is very challenging. In fact we must figure out that there are lots of conditions under which a picture can be taken: for example there may be different illuminations for a given place; so it is important to understand what is the appropriate representation of a place that is rich enough to distinguish similarly looking places. The system built over visual place recognition must respect different constraints in terms of time spent and accuracy for retrieving GPS coordinates of a certain photograph. Thus, before starting, consider that each image should be taken from different points of view, with different lighting conditions, with dynamic objects and in different long-term conditions.

Moreover visual geo-localization consists of finding region of interest in photographs: task-relevant information is not generally uniformly distributed throughout an image, in fact focusing on ”interesting” areas can often allow to reach better performances. Elements such pedestrians or trees, vehicles and road signs can introduce misleading cues into the process; because of this, Contextual Reweighing Network (CRN) [3] is a valid option in order to illustrate that the contextual information contains rich high-level information that includes structural cues like different different perspectives of buildings and architectural styles.

2. Related works

Solving visual geo-localization issues leads to the resolution of other sub-problems.

Image retrieval consists of different steps such as features extraction, feature aggregation and similarity research. The methods on which each step has been approached are different: for feature extraction the most important ones are the scale invariant feature transform (SIFT) [4], SURF [5] which tries to keep the computation faster; then RootSIFT [6] which finds better descriptor than SIFT and speed-up computational time.

Feature aggregation comes from natural language processing trying to count the occurrences of many words in a vector (”bag of words”); after that it has become a cluster problem where to assign each item of a given set to a peculiar set which represents components with the same feature: the Vector of Locally Aggregated Descriptors (VLAD) [7] is a way to accomplish to this task, in which is computed the distance between the feature and the center if its cluster for a more specific representation. Weights are also assigned to each feature in order that high-value weights represent more discriminative features. Furthermore NetVLAD [2] has been proposed as a layer plugged in CNNs.

For what concerns similarity research we can assume that the euclidean distance is the oldest method to find the most similar vector of features to a given one and it is practicable until dimensions don’t grow up. In this case it is better to use PCA technique to afford dimensionality reduction.

Visual place recognition could also be seen from another domain using 3D based methods to deal with it. 3D datasets are used to find the location of 2D query image thanks to the use of particular instrumentation that allows to reconstruct images from 2D to 3D but causing a loss in terms of computation time and storage needs [8]. For this purpose those methods are usually used with 2D-based methods that try to filter a limited number of candidates from the dataset and then processing 3D methods.

3. Methods

Like [1], [2], [3] we have casted the problem of place recognition as the task of image retrieval. We have implemented 3 different networks all based on the ResNet-18 [9] backbone without the fully connected layers and the last convolution layer. On the top of this backbone we have inserted 5 different heads (only one at a time can be selected), inspired by the works of [1], [2], [3], in order to generate the image descriptors.

3.1. Base Head

This is the simplest head we have used and it's necessary in order to have a baseline to compare the other results. After the last convolution layer of ResNet-18 we have normalized the feature map and used average pooling in order to generate the descriptors. This simple head tries to extract from the query the spatial information by comparing the average value of the features in a given area and represent the traditional way to extract those descriptors.

3.2. GeM head

Following the work of [1], we have used a Generalized Mean approach in order to extract better descriptors for the query image. The generalized mean we are using is defined as:

$$f_k = (\frac{1}{X_k}\sum_{x \in X_k} x^{p_k})^{\frac{1}{p_k}} \tag{1}$$

where X_k represent one of the normalized features map and p_k is the pooling parameter. This pooling parameter is expressing how much is localized the zone of the image the network is focusing on. The p_k parameter, although it can be learned and inserted into back propagation, it has been fixed and a single value is used for each activation map as suggested by [1]. We have inserted a fully connected layer that takes as input the pooled features in order to whiten the image descriptors since it has been shown by [1] that this approach is providing better results than using other strategies like PCA.

3.3. NetVLAD head

Inspired by the work presented in [2], we have implemented also a NetVLAD head that solves in an elegant way the problem of computing Vectors of Locally Aggregated Descriptors (VLAD), as described in [7], in CNN. This network, in order to compute those VLAD descriptors, is using two different parts. The first is called soft-assignment branch that is replacing the hard assignment of a descriptor to a single cluster with a soft assignment of the descriptor to every cluster. This is performed using a soft-max operation on top of the output of a 1x1 convolution layer. This operation produces the probabilities s_k that a given descriptor x_i belongs to a cluster k by:

$$s_k(x_i) = \frac{e^{W_k^T x_i + b_k}}{\sum_{k'} e^{W_{k'}^T x_i + b_{k'}}} \tag{2}$$

The second part, denominated VLAD core, is effectively computing the VLAD representation of the image given an image descriptor x, the cluster centers c and the computed soft assignments s following the equation:

$$V(j,k) = \sum_{i=1}^{N} s_k(x_i)(x_i(j) - c_k(j)) \tag{3}$$

where $x_i(j)$ and $c_k(j)$ represent respectively the j-th dimension of the i-th descriptor and the k-th cluster center. The obtained descriptors are then intra-normalized (using a column-wise L-2 norm), flattened into a vector and then a final L-2 norm operation is applied. For this network is fundamental to initialize the cluster centers in order to obtain good performances. This initialization is preformed in a preliminary step using a small subset of the training data available and consists in computing the features representations, using the pre-trained ResNet-18 backbone, and then extracting the descriptors by randomly selecting some of the locations of the obtained features. Then, in order to compute the cluster's centroids, k-means is used over the descriptors. The weights in the convolution layer for soft-assignment are initialized, to reproduce the results that would have been obtained with VLAD described in [7], using:

$$W = \alpha(\frac{c}{||c||_2} \cdot d) \tag{4}$$

where c and d are respectively the computed clusters centers and descriptors, α is instead selected to be large in order to better mimic the traditional VLAD.

3.4. CRN head

Seen the results provided from the previous implemented heads and the success of the attention layers to make a network focus on relevant only parts of an image, we have decided to add a context-aware re-weighting layer to the NetVLAD head, following the approach proposed by [3]. This is perfectly integrated in the NetVLAD architecture and it has the duty to produce a map that rescales the weights produced by the soft-assignment step of the NetVLAD layer. This layer is composed by an initial average pooling sub-layer that has the duty to reduce the dimensionality of the feature maps produced by the backbone.

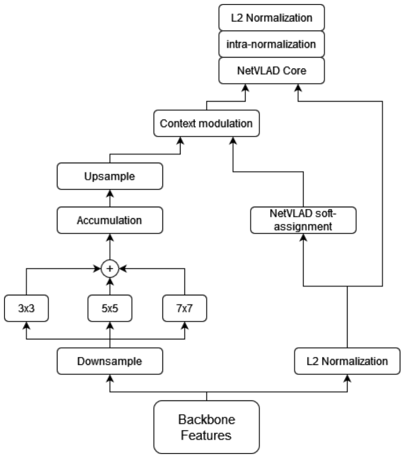


Figure 1. The architecture of the CRN head.

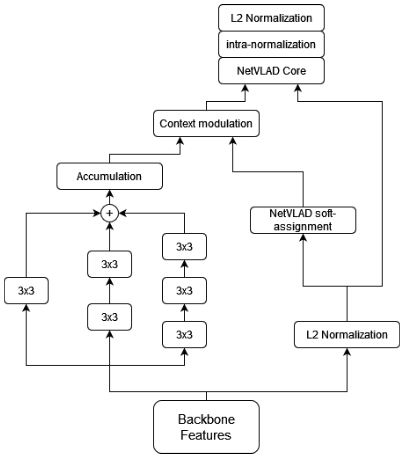


Figure 2. The architecture of the CRN2 head.

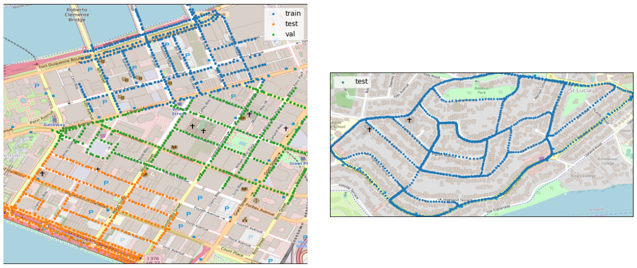
Differently from what specified in [3], it is not reducing the features maps to a fixed size but it is simply reducing by a half the dimensions of the features. In order to capture features at different spatial resolutions, 3 convolution filters (with kernel sizes respectively of 3, 5, 7) are applied to the pooled features. The output of those filter is concatenated and an additional 1x1 convolution filter is used in order to accumulate the features produced. The resulting mask is then upsampled, in order to restore the original features map dimensionality, by using a bilinear interpolator. This results into a mask that is used to re-weight the scores produced by the soft-assignment, specified into the NetVLAD description, as showed in figure 1. This last operation is performed into the context modulation layer. This layer is performing the element-wise product of the mask and the soft-assignment scores. The output of the context modulation layer is then used in the standard NetVLAD core instead of the soft-assignment scores. So the final VLAD descriptors are produced by following:

$$V(k) = \sum_{l \in L} m_l a_l^k (x_l - c_k) \tag{5}$$

where for each spatial location l of the feature map the residual of the locality x_l from the cluster center c_k is multiplied by both the soft-assignment scores a_l^k and the re-weighting mask m_l. Also this head requires the initialization of centroids as the NetVLAD head and the initialization adopted is the same described in the section 3.3.

3.5. CRN2 head

In order to reduce the number of parameters to be learned by the CRN head, we have implemented a second version of this head, called CRN2, that exploits the ideas of concatenating multiple 3x3 filters in order to obtain the same receptive field of a bigger filter but using less parameters as suggested in [10]. In particular we have replaced the 5x5 and the 7x7 filters showed in figure 1 with respectively 2 and 3 stacked 3x3 filters as showed in figure 2. In addition to that, we have used dilated convolution in order to remove the pooling and upsampling layers by generating the mask directly at the desired resolution inspired by the work proposed in [11]. This, although requires more computation, will produce more accurate masks that may help the network to better focus on the relevant parts of the images. Also this head requires the initialization of centroids as the CRN and the NetVLAD heads and the initialization adopted is the same described in the section 3.3.



(a) Pitts30k

(b) St. Lucia

Figure 3. The location of the images contained in both the pitts30k 3a dataset and the St. Lucia 3b dataset.

4. Experiments

In this section we evaluate the results obtained by the implemented heads compared to the base head.

Datasets The experiments have been run on the pitts30k dataset [2] that is containing 3 different predefined splits, respectively for training, validation and testing, composed by $10k$ images each. Those are images taken in the city of Pittsburgh from the Google street view images. In addition, also the St. Lucia dataset [12] has been used only for testing the models trained on the pitts30k dataset. The location of the images contained in both datasets can be seen in figure 3. Those two datasets have a significant domain shift since the images of Pittsburgh are mostly in a urban scenario with scenes mostly composed by skyscrapers while the images on the St. Lucia dataset are taken in a more natural scenario where the trees are much more present than buildings. We are using those dataset in order to assess how well the proposed models generalize on different scenarios.

Mining procedure In order to train our models we need to generate triplets in the form $\{I_q, I^+, I^-\}$: for each query image I_q we are looking for positive examples I^+ and negative ones I^-. In order to do so, for each query image we retrieve the images into the database that are in a range specified by the positive distance threshold parameter. We use 2 possibly different values for this threshold at train and test time. Among all images in this range, we select as best positive the one that has descriptors more similar to the query image and we use this image as I^+. All the images not in the positive distance threshold range are considered as negative samples. We select by default the 10 images, from the negative set, that have the most similar descriptors as the query image. We call those images hard negatives and we are using those instead of randomly picking from the negative set in order to make the task for the network more challenging and for obtaining a more robust model. Since the mining procedure described depends on the descriptors that vary over the training procedure, we are periodically recalculating the triplets within the epochs.

Loss function We have set the problem of visual geo-localization in approximating the location of a query image by retrieving the nearest database images in the descriptors space. For that reason, the objective of our training procedure is to make geographically close images have a similar representation in descriptor space and instead make as far as possible the representation of geographically far images. This is possible by exploiting contrastive learning and the loss function that we are using is a standard triplet loss defined by

$$L(q, p, n) = \sum_{i}^{N} max(||q - p||_2 - ||q - n_i||_2 + m, 0) \quad (6)$$

where p, q and n_i represent the descriptors extracted by the query image, the positive image and the negative ones (by default we extract 10 negatives examples for each query image as described in Mining procedure and the number of negatives samples is denoted by N). The parameter m is instead specifying a margin between the positive and negative descriptor representation of the images. In fact, if a negative sample has a distance in the descriptors space higher than the margin m the resulting loss will be 0 and, instead, if the distance between the positive and negative descriptors is lower than the margin the loss will be proportional to the margin violation.

	Descs.	R@1	R@5	R@10	R@20
Base	256	60.1	80.6	87.4	91.7
GeM	256	71.6	87.0	91.0	94.0
NetVLAD	16384	79.1	89.3	92.3	94.4
CRN	16384	81.7	**90.7**	**93.4**	**95.3**
CRN2	16384	**81.8**	**90.7**	93.2	95.2

Table 1. Results on the pitts30k test set obtained with the various heads compared with the base head. The number of generated descriptors is also shown in the column Descs.

Metric adopted Those results have been obtained by evaluating the models by using a standard evaluation procedure for place recognition. A given query image is said to be correctly localized if at least one of the N retrieved images is placed at a distance lower or equal to test positive distance threshold from the ground truth position. This distance is set, if not differently specified, to 25 meters. After that we are calculating the percentage of correctly classified images for different values of N (indicated with $R@N$).

4.1. Comparison among the proposed heads

The results of comparison between the various proposed heads are reported in table 1. Those results have been obtained on the pitts30k test set. As it's possible to notice the heads that are giving best results are the CRN and CRN2. That shows that adding attention is essential for improving the quality of generated descriptors. It's important to notice also how the results are influenced by the number of produced descriptors. In fact both the NetVLAD and the CRNs heads are generating much more descriptors with respect to the GeM and the base heads and this seems correlated to higher recalls. Since the CRNs and the NetVLAD heads are outperforming the other ones, we will focus more on those 3 during the rest of this section.

The table 2 is reporting the results obtained on the St. Lucia dataset. As it's possible to notice, those results are much lower than the ones obtained on the pitts30k dataset and this is probably due to the significant domain shift between the datasets. However it's important to notice how also in this case attention is providing best results especially for lower recall values while simpler methods like GeM are providing the best performances at higher recall values.

	Descs.	$R@1$	$R@5$	$R@10$	$R@20$
Base	256	26.8	45.4	55.1	63.9
GeM	256	41.7	62.2	**70.6**	**81.4**
NetVLAD	16384	44.8	59.4	66.7	75.3
CRN	16384	**48.7**	**63.3**	69.7	75.9
CRN2	16384	47.6	62.6	69.2	75.3

Table 2. Results on the St. Lucia test set obtained with the various heads compared with the base head. The number of generated descriptors is also shown in the column Descs.

Network	$R@5$	Params. (k)	Time (ms)
Base	80.6	0.0	15.76
GeM	87.0	65.8	15.81
NetVLAD	89.3	16.3	20.64
CRN	90.7	545.9	21.51
CRN2	90.7	262.0	28.47

Table 3. The running time for computing the descriptors of a single image for the various networks computed on a GPU NVIDIA Tesla K80. The $R@5$ are computed on the pitts30k test set. The number of parameters are not including the common backbone and, if required by the network, are not considering the cluster centroids.

4.2. CRN and CRN2 models

As it's possible to notice from table 1, the performances of the CRN and CRN2 networks are very close one to each other as expected from the definition of the two networks. It's important to notice that the CRN2 network is using less parameters with respect to the CRN network. In fact, the CRN2 network is using only 245K parameters compared to the 529k parameters used by the CRN network for the generation of the mask. As shown in the table 3, the time required to extract descriptors for a single image is higher for the CRN2 network with an increase of 32.3% of the required execution time. This is due to the removal of the downsampling layer present in the original CRN implementation that implies the analysis of the image at full resolution and this has to be taken in consideration during the deploy phase where the descriptors for the query images have to be computed online. We have also noticed that an increase of the time required to produce descriptors is correlated to better recalls values with the exception of the CRN2 head which is providing the same results as the CRN head and also this has to be taken in consideration while deploying an application using those networks.

4.3. Visual results

In this section we are going to show some visual results obtained with the various heads. In the figure 4 we are reporting the results obtained with the NetVLAD, CRN and CRN2 heads on a query image taken from the pitts30k dataset. How it's possible to notice from the second column of the figure 4 that shows the soft-assignment scores (summed over the 64 clusters used) heatmap on the query image, the context-aware re-weighting layers used are fundamental in order to make the network focus only on the relevant parts of the image. It's important to notice that the CRN2 network is focusing at smaller portions of the query image and this can increase the difficulty in finding really discriminative zones of the query while the CRN model is focusing on wider areas that include the most relevant parts of the image.

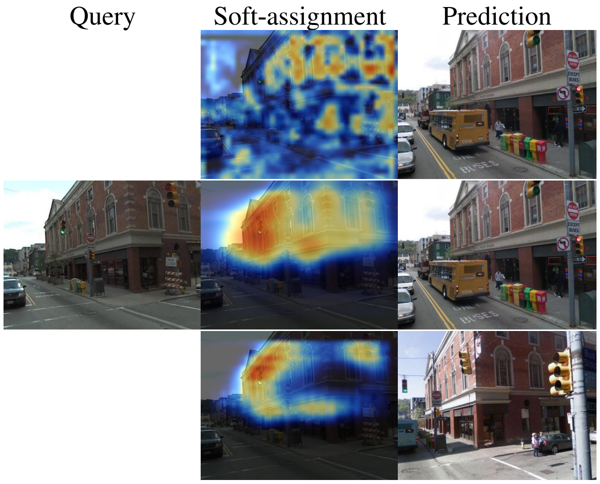


Figure 4. Results on a image from the pitts30k dataset. The first row is showing results obtained with NetVLAD, the second with the CRN network, the third one with CRN2 network all on the same query image shown in the first column. The soft-assignment column is computed by summing over the 64 cluster soft-assignment scores.

5. Ablation study

In this section we discuss the effect of changing one by one some parameters of the NetVLAD network, we especially focused on trying different learning rates and optimizers, modifying the distance at which positives are taken.
We also changed the input images of the dataset by implementing some data augmentation techniques and by changing the resolution of the images.

5.1. Comparison between different learning rates and optimizers

As first ablation study we tried different learning rates, from the table 4 we can see that the best results in calculating the percentage of correctly localized images are obtained with a learning rate of 10^{-5}, with this learning rate the network is superior to the other configuration of learning rate in each case. We also noticed that by decreasing the learning rate we increment the number of epochs needed to end the training. As optimizer algorithm we decided to test the Stochastic Gradient Descent (SGD) and the Adam optimizer. The results of the application of SGD, displayed in the table 5, are showing an increment of the recall int the Pitts30k dataset and a decrement in the St. Lucia dataset compared to ones obtained with the Adam optimizer. We decided to use the Adam optimizer in the rest of our study since it seems to provide more stable results and a lower training time.

	$R@1$	$R@5$	$R@10$	$R@20$
lr = 10^{-3}	78.6	89.4	92.5	94.7
lr = 10^{-4}	79.1	89.3	92.3	94.4
lr = 10^{-5}	**82.3**	**92.7**	**95.0**	**97.0**

Table 4. Results obtained with the NetVLAD head on the pitts30k test set with different learning rates

	$R@1$	$R@5$	$R@10$	$R@20$
(Adam) Pitts30k	79.1	89.3	92.3	94.4
(SGD) Pitts30k	**79.5**	**90.4**	**93.0**	**95.0**
(Adam) St. Lucia	**43.0**	**58.5**	**67.4**	**74.7**
(SGD) St. Lucia	42.1	56.8	63.7	71.4

Table 5. Results on the pitts30k and St. Lucia test sets obtained with the NetVLAD+Adam head compared with the NetVLAD+SDG head.

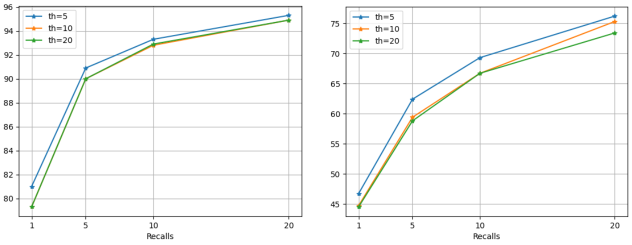
5.2. Comparison between different positive distance threshold

Initially the distance at which positive are taken at train time was set at 10 meters, we tried to change the parameter $train\_positives\_dist\_threshold$ with different values. The graph contained in figure 5 shows that, during training, the positive distance threshold set at 5 meter outperforms all the other thresholds in every recall in both the Pitts30k and St Lucia datasets.

By setting a larger train positive distance threshold we have worse performances because, for the network, the task of localization is easier with larger thresholds since the position is more approximate and the network is producing less discriminative descriptors while training with a more challenging task has provided good results at test time.

We also tried different test positive distance threshold and in this case there is a big difference between the different distances, greater distances perform in a better way than smaller ones as expected since we are reducing the desired precision of the network by accepting as good also images far from the query. In the Pitts30k dataset we can see that the recall on one image is 65.1 with the positive threshold distance set at 10 meters while the recall with the distance set at 50 meters is 82.9, there is a margin of 17.8%

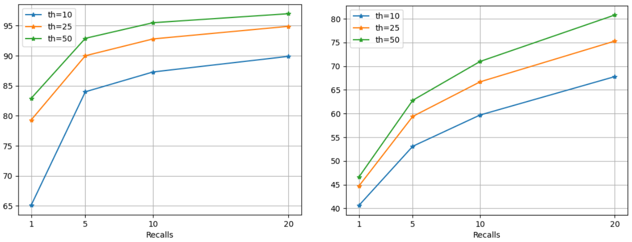
The same trend is maintained in all the different recalls on both the Pitts30k and St. Lucia datasets, but while in the Pitts30k the margin between the distance threshold set at 10 meters and the one set at 50 meters stabilizes, in the St Lucia Datasets keeps incrementing, initially it's 6% at R@1 and becomes 13% at R@20.



(a) Pitts30k

(b) St. Lucia

Figure 5. Graph showing the recalls obtained with different train positive distance threshold on both the pitts30k 5a dataset and the St. Lucia 5b dataset.



(a) Pitts30k

(b) St. Lucia

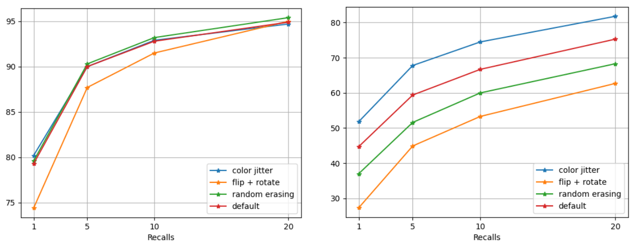
Figure 6. Graph showing the recalls obtained with different test positive distance threshold on both the pitts30k 6a dataset and the St. Lucia 6b dataset.

5.3. Comparison between different data augmentation techniques

In order to see how the results change we tried some data augmentation techniques, we decided to use 3 of them: Color Jitter, Horizontal Flip + Rotate and Random Erasing. With Color Jitter we randomized the contrast and brightness of the image, with Horizontal Flip + Rotate we flipped the image over the vertical axis and we applied an additional rotation to the image and with the last transformation we have applied rectangular black patches on the image in order to mimic occlusions.

For the Pitts30k dataset we can see in figure 7a that all the data augmentation techniques have a similar performance compared to the default configuration of the dataset; Only the flip and rotate technique gets sightly worse results in the first recalls.

For the St. Lucia dataset we can see in figure 7b that all the data augmentation techniques have very different perfor-



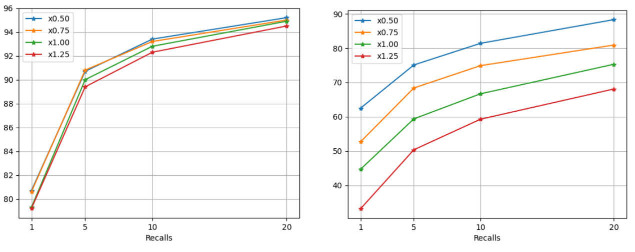
(a) Pitts30k (b) St. Lucia

Figure 7. Graph showing the recalls obtained with different augmentation techniques on both the pitts30k 7a dataset and the St. Lucia 7b dataset.

mance, in this case only the Color Jitter transformation gets better results than the default configurations, while Random Erasing and Flip + Rotate get sightly worse results in all the recalls.
The Color Jitter technique is the only technique that gets better results because it's the only one that does not change the structure of the images.

5.4. Comparison between different images sizes

The last ablation study has been the variation of the images sizes by scaling their dimensions by a scaling factor. We decided to try 4 scaling factors (including the standard one 1.00), the value selected are: 0.50, 0.75, 1.00, and 1.25. The results of the scaling factor are shown in figure 8, for both the Pitts30k and St. Lucia dataset it's clear that a reduction of the image size leads to an increment in the recall's value; This increment is way more visible in the St. Lucia dataset (figure 8b). This is probably due to the fact that the network can ignore more easily small details not relevant to the task since they are less present in the small resolution pictures and easily focus on the relevant parts of the image like the shapes of the buildings.



(a) Pitts30k (b) St. Lucia

Figure 8. Graph showing the recalls obtained with different input image size on both the pitts30k 8a dataset and the St. Lucia 8b dataset.

6. Conclusions

We tried two different convolutional neural network on two different datasets; after seeing that the GeM network performs worse than NetVLAD we decided to continue performing some different configurations on the second network. In order to increase the performances of the NetVLAD one we took inspirations from [3] adding an attention layer which was able to understand which features of the photograph were relevant in order to better recognize the locations in the photograph.
Furthermore we fine-tuned the performances of the network by performing some ablation studies over the parameters of the network and different augmentation techniques: from resizing the whole image to rotating and flipping it. Not all changes we made led to improvements of the network performances, but each of them has been proved useful to understand how the applied changes influence the task of visual geo-localization.

References

[1] F. Radenovic, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *TPAMI*, 2018. 1, 2

[2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *TPAMI*, 2018. 1, 2, 3

[3] H. Jin Kim, E. Dunn, and J.-M. Frahm, "Learned contextual feature reweighting for image geo-localization," *CVPR*, 2017. 1, 2, 3, 7

[4] D. Lowe, "Distinctive image features from scale-invariant key-points," *International Journal of Computer Vision*, no. 60, p. 91, 2004. 1

[5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 110, pp. 346–359, 2008. 1

[6] R. Arandjelovic and A. Zisserman, "All about vlad," pp. 1578–1585, june 2013. 1

[7] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311. 1, 2

[8] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," 2018. 2

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CVPR*, 2016. 2

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. 3

[11] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017. 3

[12] M. Warren, D. McKinnon, H. He, and B. Upcroft, "Unaided stereo vision based pose estimation," in *Australasian*

Conference on Robotics and Automation, G. Wyeth and B. Upcroft, Eds. Brisbane: Australian Robotics and Automation Association, 2010. [Online]. Available: http://eprints.qut.edu.au/39881/ 4