Image Retrieval for Visual Geo-localization

Reza Barati
Department Of Control And Computer Engineering
Politecnico di Torino
reza.barati@studenti.polito.it
S301309

Saeedeh Javadi
Department Of Control And Computer Engineering
Politecnico di Torino
saeedeh.javadi@studenti.polito.it
S301409

Faezeh Saeedian
Department Of Control And Computer Engineering
Politecnico di Torino
faezeh.saeedian@studenti.polito.it
S301308

***Abstract*—The place recognition problem often has been regarded as an instance retrieval task, where by searching through a large geo-tagged database, the locations of the images with the most visual similarity are identified, and these locations are used to estimate the position of the query image. In this work, we address the problem of large-scale visual place recognition. we attempt a few experiments to enhance the "NetVLAD: CNN architecture for weakly supervised place recognition" outcome. These studies are divided into two basic components, the first one is to improve the robustness of the baseline model to the Night domain. To solve this problem, we use data augmentation by using of functional transformers. The second one is some general improvements. By scaling the photos' resolution up and down, we examine the effects of image resizing. We also use various optimizers, such as ASGD and AdamW, and try to determine the optimal values for the learning rate and weight decay, and also try different schedulers like ReduceLrOnPlateau, to test the effects of changes on the performance of the base model. As a post-processing step, we re-rank the final predictions of the baseline model for a given query by using three different approaches. The code is available at https://github.com/saeedehj/Geo-localization.git.**

***Index Terms*—Visual place recognition, Image retrieval, NetVLAD, Image augmentation.**

I. Introduction

Visual Place Recognition (VPR) is one the growing topic these days, and addresses this question "given an image of place, where this picture is taken, and can a human or robot recognize this image belongs to a place it has already seen?"

VPR can commonly be considered as an image retrieval task, that the place of a given image (query) is estimated by using similarity measure through a database which they are collection of geo-tagged images. As shown in figure 1, this approach traditionally implemented in a pipeline include: First, for each image in database, extract local features by using methods such as SIFT [2], SURF [3], RootSIFT [4] etc. However, it is possible to use a global feature descriptor for whole image in VPR, but they are not robust to changing of viewpoint, occlusions and clutter in comparison with local features descriptors. Secondly, the local features are aggregated into a single vector descriptor for the image using methods such as BOW [**?**], VLAD [5] [6] or Fisher vector [7]. Third, a similarity search by using Euclidean distance on the image descriptors is performed to find the best matches between database images and the query image and finally, some post-processing steps is employed to refine the result of previous steps.

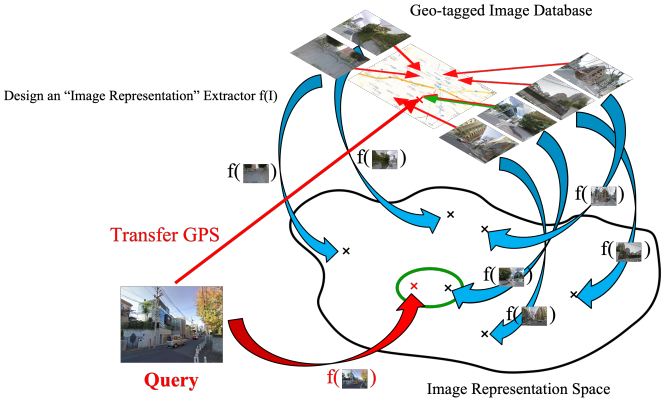

Fig. 1. The overview of Visual Place Recognition.

Recently, convolutional neural networks (CNNs) have played a powerful role as generators of image representations for various place recognition tasks such as object classification, scene recognition or object detection. In last few years, several works have proposed using CNN-based features which include treating activations from certain layers directly as descriptors by concatenating them [8] [9], or by pooling layer [10] [11]. However, those approach used CNNs as black-box descriptor extractors and none of them are not fully trainable in end-to-end manner.

In contrast to previous methods, NetVLAD [1] are proposed to mimic VLAD in a CNN framework and design a trainable generalized VLAD layer which is a differentiable pooling. A novel trainable Generalized-Mean (GeM) [12] pooling layer that generalizes max and average pooling and show that it boosts the performance of image retrieval task. In MultiRes-NetVLAD [13] approach, NetVLAD representation learning are augmented with low-resolution image pyramid encoding which it is leaded to richer place representations and Patch-NetVLAD [14], which combines the advantages of local and global feature descriptor methods by deriving patch-level features from NetVLAD residuals.

While many approaches are proposed to improve the image retrieval task but still there are many challenges can be solved. In this research, we address some issues for image retrieval task to improve the performance of base network NetVLAD to recognizing the place. We propose some ap-

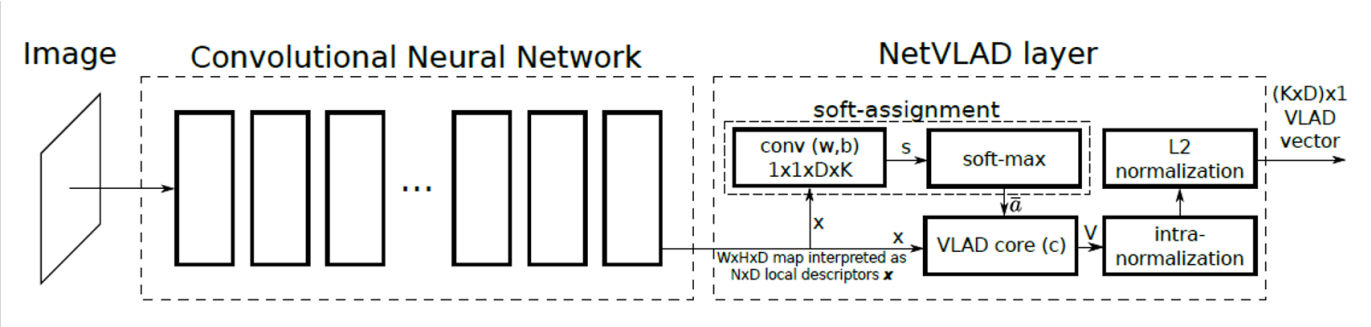

Fig. 2. **CNN architecture with the NetVLAD layer**. The layer can be implemented using standard CNN layers (convolutions, softmax, L2- normalization) and one easy-to-implement aggregation layer to perform aggregation ,joined up in a directed acyclic graph.

proaches to increase robustness to night domain by using of data augmentation transformers model. Also, we implement some methods for post processing stage to refine the top-k candidates retrieved with the kNN.

II. Related Works

Before the research conducted by Arandjelovic, Relja, et al, all relevant learning-based approaches fell into one of the following two categories, learning for an auxiliary task and learning over hand-engineered, superficial descriptors that cannot be fine-tuned to the intended task [1].

A convolutional neural network (CNN) architecture was developed by Arandjelovic, Relja, et al [1], that can be trained end-to-end for the place recognition task, as shown in figure 2. The core component of this architecture, NetVLAD, is easily pluggable into any CNN design and can be trained using back-propagation. Additionally, they created a novel weakly guided ranking loss that enables end-to-end learning. The NetVLAD network architecture uses the Vector-of-Locally-Aggregated-Descriptors (VLAD) approach to generate a condition and viewpoint invariant embedding of an image by aggregating the intermediate feature maps extracted from a pre-trained CNN. Let $f_\theta : I \rightarrow \Re^{H \times W \times D}$ be the base architecture which given an image I, outputs a H × W × D dimensional feature map F. The NetVLAD architecture aggregates these D dimensional features into a K × D dimensional matrix by summing the residuals between each feature $x_i \in \Re^{D}$ and K learned cluster centers which is weighted by soft-assignment. Formally, for N × D dimensional features, let the VLAD aggregation layer $f_{VLAD} : \Re^{N \times D} \rightarrow \Re^{K \times D}$ be given by

$$f_{VLAD}(F)(j,k) = \sum_{i=1}^{N} \bar{a}_k(X_i)(x_i(j) - c_k(j)) \qquad (1)$$

where $x_i(j)$ is the j^{th} element of the i^{th} descriptor, $\overline{a_k}$ is the soft-assignment function and c_k denotes the k^{th} cluster center. After VLAD aggregation, the resultant matrix is then reshaped into a vector and after normalization used as the image presentation.

The NetVLAD, uses a new ranking loss function to optimize VPR performance. In this function, for each test query image q, a weakly supervised ranking loss , L_θ, defines as

$$L_\theta = \sum_j l(\min d_\theta^2(q, p_i^q) + m - d_\theta^2(q, n_j^q)) \qquad (2)$$

For a training tuple (q , p_i^q , n_j^q), which p_i^q are set of potential positives, n_j^q are set of definite negatives and m is a constant parameter which is a margin.

PatchNetVLAD [14], provides a formulation for combining the benefits of both local and global descriptor methods by deriving patch-level features from NetVLAD residuals. To increase the appearance robustness of local descriptors, it employs global descriptor techniques. This paper uses methods that enable further performance improvements through an effective multi-scale fusion of patches, in contrast to earlier key point-based local feature descriptors, to take into account all the visual content within a larger patch of the image. This method is a viewpoint-invariant visual place recognition system that compares locally extracted locally-global descriptors from a set of patches in the feature space of each image to produce a similarity score between two images

Khaliq, A., Milford, M., Garg, S. in 2022 [13] added low-resolution image pyramid to NetVLAD representation learning, and combined features from different resolutions to produce a condensed but more accurate global place descriptor. The resulting image descriptor is adaptable to various feature aggregation algorithms, and in test step, robustness of changing image resolutions improved.

Geometric verification, is based on local features and is the most widely used method for Re-ranking images [6] [15] [16] [17]. The introduction of query expansion techniques for image retrieval [18] [19] [20] was also influenced by text retrieval. The way these methods analyze the neighborhood closest neighbor graph for each test query sets them apart from geometric verification. Diffusion-based approaches [21] [22] aim to learn the structure of the data manifold by propagating similarity over the global affinity network created on a query and all of the database images.

In our work, re-ranking of the set of the closest images in the database to the query is based on the geographic Euclidean distance. Where, after constructing the sorted set of the closest images in the database to the query based on their Euclidean

distance of the descriptors, the members of the initial set are reordered depending on their geographical Euclidean distance.

Several research have shown that multi-scaling affects the model's performance. These approaches vary depending on whether multi-scale information is used during the training process or post-processing.

Learning with Multi-Scale: Some studies [23] [24] have focused on how to enhance VPR performance by adding multi-scale features during the training phase. A trainable end-to-end framework with a deep fusion of multi-layer max-pooled convolutional features is proposed in [23]. This is accomplished by using convolutional kernels of different sized.

Multi-Scale Processing 'After' Training: Many researchers have developed multiscale algorithms that post-process CNN features or use different image resolutions because it is not always simple to use multi-scale information during the training process. To enhance local feature matching, [14] post-processed the final convolutional layer of NetVLAD with different patch sizes. To get a compact representation, [12] combined the final feature maps from several image sizes.

In this work, Each image in the query and database is passed through the model, resized, and given a descriptor. This process is carried out for every image with a variety of sizes, after which the average of the results is computed to identify a single descriptor for a particular image. Additionally, the previous procedure is carried out once for an image's higher resolutions and once for its lower resolutions in order to take into account the effects of higher resolutions (up-scale) and lower resolutions (down-scale) on baseline model accuracy. The following method of multi-scaling involves selecting random patches from an image, calculating their descriptor vectors, then averaging the descriptors over all patches to create a single vector.

Data augmentation is popular for VPR tasks as new data will be generated very easily by applying image transformations such as shifting, scaling, rotation, and etc. Unfortunately, it is not as straightforward to apply in all domains as it is for images.

This approach is practically performed by trial and error, and the types of augmentation performed are related to the time, imagination and experience of the researcher.

Manual augmentation techniques such as rotating, color jittering and adding different kinds of noise like gaussian to the data, are described in depth in [25] which proposed a network that automatically generates augmented data by combining two or more samples from the same class. They provide a list of recommended data augmentation methods. Tran et al. used a Bayesian approach to generate data based on the distribution learned from the training set [26]. DeVries and Taylor used simple transformations in the learned feature space to augment data [27].

To better understand how the proposed architecture of Relja, et al works, we do some experiments in section 3. The NetVLAD layer and the GeM pooling will be used as the head of a ResNet-18 that has already been trained on ImageNet. We test a few extensions in section 4 to enhance the output of the main model. The purpose of the extensions is to increase the robustness to the night domain by applying various techniques to improve the top-k candidates found by the kNN.

TABLE I
SUMMARY OF THE DATASETS

	Train Database/ Query	Test Database/ Query	Queries type
Piits30k	10001/7417	27191/1001	Panorama
sf-xs	0/0	12772/105	phone*
Tokyo night	0/0	12772/315	phone*
Tokyo-xs	0/0	10001/6817	phone*

[a]"phone" denotes that the images were taken using a smartphone.

III. EXPERIMENT

In this section, we describe some key components of our experiment. We first introduce the datasets which we selected for our model, followed by describing evaluation metric that used and finally, we consider the baseline model.

A. Datasets

We make use of four extremely diverse datasets that collectively represent a range of real-world scenarios: various scales, levels of inter-image variability, and various camera types. We employ the Pitts30k [1] dataset for training. Pitts30k is fairly homogeneous, with all photographs using the same camera, resolution, and weather conditions. We additionaly test all models on four additional datasets to determine dataset robustness: Tokyo-night [30], Tokyo-xs [30], sf-xs [31]. Further details on these datasets, are given below.

Pitts30k includes 24k test queries created from Street View but taken at separate times, years apart, and 30k database photos collected from Google Street View and images are cropped from a 360° panorama The division of this dataset into three roughly equal sections for training, validation, and testing was done geographically to ensure that the sets contain distinct images. Each portion of the dataset has about 10k database photos and 8k queries.

Tokyo-xs is a subset of Tokyo 24/7, that presents a relatively large database (from Google Street View) with fewer queries, which are divided into three sets of roughly equal size: day, sunset, and night. These are manually gathered using phones.

Tokyo-night is also a subset of Tokyo-xs that only night images have been selected.

sf-xs, similarly, to Tokyo-xs, sf-xs is a subset of San Francisco extra Large that is made up of a large database gathered by a car-mounted camera and significantly fewer phone-based queries. more detail about 4 databases is shown in table I.

B. Evaluation metric

We follow the conventional place recognition evaluation process. If at least one of the top N database images is located less than d = 25 meters from the query image's ground truth

TABLE II
RECALLS THE BASELINE MODEL

	Pitts30k		sf-xs		Tokyo-xs		Tokyo-night	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Gem	76.1	88.6	13.8	27.1	35.2	50.5	10.5	27.6
NetVlad	85.9	92.7	34	50.1	59.7	73.3	26.7	43.8

position, the image is considered to have been appropriately localized. We use recall1(R@1) and recall5(R@5) to compare models.

C. Baselines assessment

In this part we execute the model that proposed in [33] as base line model. This model uses a ResNet-18 pre-trained on ImageNet and uses as head both the NetVLAD layer and the GeM pooling. We train both baselines using a triplet loss which is proposed by Relja, et al and using the Pitts30k dataset for training. Due to resource constraints, all models in this research were trained over seven epochs so that they could be compared. we test the models on three different sf-xs, Tokyo-xs and Tokyo-night datasets.

NetVLAD The core of the NetVLAD architecture is a generalized VLAD layer that combines mid-level convolutional features extracted from the entire image into a small single vector representation for effective indexing in the same manner as VLAD [5]. Geo-tagged image sets made up of collections of pictures taken from the same places at various times of the year are used to train NetVLAD.

Generalized Mean Pooling (GeM) reduce the dimensionality, perform normalization and finally the final output is the image descriptor. Advantages of GeM pooling are over Max/Average pooling [12]. The comparison between NetVLAD and GeM is shown in table II.

IV. EXTENTION

A. Optimizer and Schedulers

The main function of an optimizer is to decide how much to change the neural network's weights and learning rate in order to minimize losses. ADAM is one of the most popular optimization algorithms todays (Adaptive Moment Estimation). Adam is a stochastic optimization technique that employs the ideas of momentum and gradient descent to minimize the loss function and determine its minimum value. The baseline model use Adam as optimizer.

We compare two optimizers to reach out to the highest accuracy of baseline model. The optimizers are AdamW and Average Stochastic Gradient Descent (ASGD). ASGD, averages the weights that are calculated in every iteration and AdamW optimizer seeks to insert a regularization based on the decay of weights in the optimization process. The regularization consists of adding a penalty to the loss function to produce simpler models that can be used successfully when new data is available [32].

For finding the best parameters for training, learning rate and weight decay are changed for these two optimizers, where, learning rate is controlling the size of the update steps along the gradient, and weight decay is a form of regularization to lower the chance of overfitting.

The results are given in table III. All the results are reported for recall 5. For the pitts30k dataset, the AdamW optimizer shows a minor increase in the model at wd=0.001 and lr=0.000005, and 3 percent enhancement for Tokyo-night in wd=0.05 and lr= 0.000005, and a slight improvement of less than 1 percent for Tokyo-xs in the same hyper parameters and no improvement for the sf-xs dataset.

In ASGD, the best recall in the pitts30k dataset is 93.0 in wd=0.0001 and lr=0.05, and it improves by 5 percent in the same lr and with wd= 0.001 in Tokyo-night. In the Tokyo-xs dataset, the model achieves the best recall of 75.5 with the same hyper parameters of last dataset, and the model improves 2.5 percent in lr=0.05 and wd=0.0001 for the sf-xs dataset.

Also, The recall of the dataset Tokyo-night is improved in comparison to the baseline, by adding the scheduler ReduceLrOnPlateau on AdamW optimizer (R@1 and R@5 are increased 1.9 and 1 percent respectively). Moreover, the R@1 and R@5 of the dataset sf-xs are better by 2.6 and 1.6 percent respectively in compare to the baseline model by adding the same scheduler on ASGD optimizer.

Figures 3 shows the outcomes of modifying the weight decay and learning rate for the ASGD and AdamW optimizers.

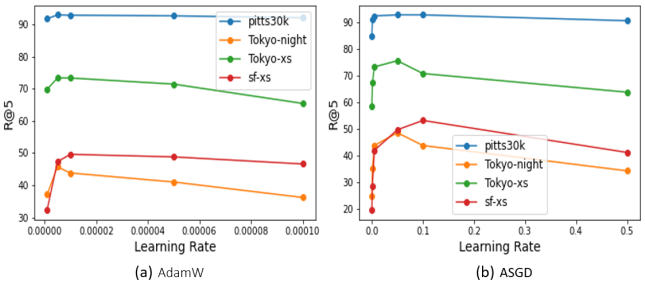

Fig. 3. Recall versus learning rate.

B. Re-ranking

In the image retrieval task, the mainstream approaches is focusing on learning a better feature representation. However, directly tackling the distance or similarity measure between images could also be efficient. To tackle this issue, the idea of re-ranking the top-k retrieved images comes up.

In this section, some strategies have been applied to refine the top-k candidates retrieved with the kNN.

First approach: In this approach, a clustering model applied on top 20 database candidates (on their geographical distance) then, clusters sorted based on the number of elements. It means, if one cluster includes 5 images and one other has 3, cluster with high number of images should be in first place and other clusters fill other positions based on the number of elements.

TABLE III
FINE-TUNING LEARNING RATE AND WEIGHT DECAY FOR ADAMW AND ASGD OPTIMIZERS

optimizer	num.epoch	lr	Weight Decay	Pitts30k R@1	Pitts30k R@5	Tokyo-xs R@1	Tokyo-xs R@5	sf-xs R@1	sf-xs R@5	Tokyo-night R@1	Tokyo-night R@5
Adam	7	0.00001	0	85.9	92.7	59.7	73.3	34	50.1	26.7	43.8
AdamW	7	0.000001	0.001	83.3	91.7	53.0	69.8	19.3	32.3	18.1	37.1
	7	0.000005	0.0001	**85.5**	**92.9**	55.9	73.3	31.5	47.3	26.7	45.7
			0.001	**85.5**	**92.9**	55.9	73.3	31.5	47.3	26.7	45.7
			0.01	85.5	92.7	56.2	73.7	31.8	47.1	24.8	46.7
			0.05	85.3	92.9	**56.8**	**74.0**	31.6	47.1	**26.7**	**46.7**
	7	0.00001	0.001	85.8	92.8	59.0	73.3	**34.1**	**49.6**	25.7	43.8
	5	0.00005	0.001	85.9	92.6	58.4	71.4	33.7	48.8	28.6	41.0
	7	0.0001	0.001	84.5	92.0	51.4	65.4	33.9	46.6	22.9	36.2
ASGD	4	0.00001	0.001	70.9	84.9	44.8	58.4	11.2	19.6	15.2	24.8
	7	0.001	0.001	81.4	91.0	49.2	67.3	16.3	28.5	16.2	35.2
	7	0.005	0.0001	84.5	92.4	56.2	73.7	27.4	41.4	24.8	43.8
	7		0.001	84.4	92.4	56.2	73.3	27.6	42.0	25.7	43.8
	7	0.05	0.0001	**86.1**	**93.0**	60.0	73.7	39.3	52.8	27.6	45.7
	5		0.001	85.4	92.8	**60.6**	**75.6**	35.3	49.7	**28.6**	**48.6**
	4		0.01	84.6	92.3	57.5	71.7	32.8	47.4	34.3	45.7
	5	0.1	0.001	85.6	92.8	59.7	70.8	**38.6**	**53.2**	27.6	43.8
	4	0.5	0.001	81.5	90.6	50.5	63.8	27.0	41.2	25.7	34.3
	4	1	0.001	68.0	83.8	22.2	34.9	6.4	14.7	3.8	11.4

[a]Bold numbers represent the best results for each optimizers.

Second approach: for improving the result of the previous approach, images in each cluster are sorted based on distance of their descriptor to query image (ascending).

Third approach: then, the first image in the k-th group is placed in the k-th position of the list and other elements are placed respectively.

Fourth approach: finally, the first elements are sorted based on distance of their descriptor to query before being at the top of the list.

Clustering methods that used in this approach are DBSCAN and agglomerative and our proposed model. In proposed model, the distance matrix for 20 database images is computed. Euclidean distance of geographic location is used and images that are closer than 5 meters are placed in a same cluster.

The results for three different clustering models and their four different approaches is shown in table IV.

The result of table IV shows that the third and fourth approaches of the clustering method DBSCAN improve the R@5 of pitts30k by 1.1 percent, and rise the R@5 for the sf-xs dataset by 1.2 percent. While, any method do not improve the R@5 tokyo-xs. Moreover, these approaches and the fourth approach of proposed model increase the R@5 of the dataset Tokyo-night 1 percent.

C. Multi-scale Testing

In image retrieval systems, image resolution is crucial since both the query and the database's images are analyzed pixel by pixel [33]. The purpose of this part is to determine the impact of multi-scaling on the baseline model accuracy.

To test this impact, each image in the query and database resize and pass-through model to obtain a descriptor. This operation is done for each image with different sizes and then the average of obtained descriptors are calculated to find a single one for a given image. Moreover, to consider the impact of higher resolutions (up-scale) and lower resolutions (down-scale) on baseline model accuracy, the above process is done one time for higher resolutions of an image and one time for its lower resolutions.

As a next approach of multi-scaling, we take patches from random parts of an image and calculate its descriptor vector

TABLE IV
RE-RANKING POST-PROCCESING METHODS

Recall		Pitts30k R@1	Pitts30k R@5	sf-xs R@1	sf-xs R@5	Tokyo-xs R@1	Tokyo-xs R@5	Tokyo-night R@1	Tokyo-night R@5
Baseline model		85.9	92.7	34	50.1	59.7	73.3	26.7	43.8
Proposed model	First approach	71.2	83.9	29.4	40.3	47.6	64.8	25.7	39
	Second approach	71.2	83.9	29.4	40.3	47.6	64.8	25.7	39
	Third approach	71.2	90.8	29.4	42.8	47.6	67	25.7	41
	Fourth approach	85.9	93.1	34	50.2	59.7	73.3	26.7	**44.8**
DBSCAN	First approach	46.9	68.5	20.2	36.3	36.2	58.4	19	39
	Second approach	85.9	91.1	34	47	59.7	70.8	26.7	41.9
	Third approach	46.9	**93.2**	20.2	**51.2**	36.2	73.3	19	**44.8**
	Fourth approach	85.9	**93.2**	34	**51.2**	59.7	73.3	26.7	**44.8**
Agglomerative	First approach	74.7	77.3	28.6	37.4	51.7	57.5	26.7	36.2
	Second approach	85.9	88.9	34	43.6	59.7	67.6	26.7	36.2
	Third approach	74.7	92.8	28.6	44.4	51.7	71.7	26.7	41
	Fourth approach	85.9	92.8	34	44.4	59.7	71.7	26.7	41

[a]Bold numbers represent the best results for dataset.

and average the descriptors for all patches of an image to reach a single vector.

The results of table V shows that, on the one hand, down-scaling didn't improve the accuracy of both pitts30k and Tokyo-XS databases in R@1 and R@5 in comparison to that of the baseline model, whereas that of Tokyo-XS decreased by a narrow margin (for R@1 and R@5 was lower by 1.3 percent and 0.6 percent respectively). On the other hand, down-scaling improves the accuracy of the sf-xs and Tokyo-night datasets in both R@1 and R@5, whereas the accuracy of Tokyo-night is increased by some 2 percent for R@1 and 1 percent for R@5. Also, sf-xs has a significant rise (R@1 and R@5 are higher by 4.5 percent and 4.1 percent respectively)

As is shown in table V, up-scaling decreases the accuracy of all considered datasets in both R@1 and R@5, except sf-xs where the result of R@1 and R@5 are higher 3.9 percent and 0.4 percent respectively.

The outcome of multi-scaling demonstrates that this method significantly improves the outcome of the Tokyo-night dataset by 8 percent in recall5 and about 6 percent in recall1, while the results on the other datasets do not have any notable improvement.

D. Smart Data Augmentation

Image augmentation, translation, and transformation are one of the most state-of-the-art research topics these days. Many research aims to purpose a method for providing a high-quality transformation of the day-time image to a night-time image.

Data augmentation, a popular technique in deep learning, is commonly used in supervised learning to prevent overfitting and enhance generalization. Data augmentation is the process of randomly applying semantics-preserving transformations to the input data to generate multiple realistic versions of it, thereby effectively multiplying the amount of training data available. It aims at artificially enlarging the training dataset from existing data using various translations, such as rotation, flipping, cropping, adding noises, etc. The simplest example is flipping an image, which preserves its contents while generating a second unique training sample.

Traditionally, resilience to domain shift, especially regarding the Day/Night domains, is critical for VG models. In this experiment, the goal is to improve the recall on the datasets using data augmentation and transformation techniques, especially on Tokyo-Night dataset which contains 105 query images taken using mobile phone cameras. This is one of the most challenging datasets where the queries were taken at night, while the database images were only taken during the day as they originate from Google Street View.

We use torchvision.transforms module to perform some manipulation of the data and implement the data augmentation on the dataset to make it suitable for training. These transformations can be chained together using Compose. Not only there are various transformations in torchvision but also you can build a customized transformation pipeline by using functional transforms which gives fine-grained control over the transformations.

Randomized transformations will randomly apply the same transformation to all the images of a given batch, but they will create different transformations in different calls. While for reproducible transformations across calls, you should use functional transforms. For example, the standard 'torchvision.transforms.ColorJitter' transform, given a value of brightness, uses a uniform sampling inside "[max(0, 1 - brightness), 1 + brightness]", and therefore

TABLE V
MULTI-SCALE TESTING

	Pitts30k		Tokyo-xs		sf-xs		Tokyo-night	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
baseline	85.9	92.7	34	50.1	59.7	73.3	26.7	43.8
sum + upscale	84.5	92.6	**37.9**	**50.5**	59	68.6	25.7	33.3
mean + upscale	84.5	92.6	**37.9**	**50.5**	59	68.6	25.7	33.3
sum + downscale	85.8	92.7	**38.5**	**54.2**	58.4	72.7	**28.6**	**44.8**
mean + downscale	85.8	92.7	**38.5**	**54.2**	58.4	72.7	**28.6**	**44.8**
mean + 4 random Patches(320, 430)	76.3	88.7	28.2	41.9	50.8	67.9	**27.6**	**44.8**
mean + 5 random Patches(240, 320)	68.1	84.6	23.2	34.1	41.6	58.7	21	36.2
mean + 5 random Patches(320, 430)	78.2	90.5	29.4	43.5	54.6	69.8	**33.3**	**51.4**

[a]Bold numbers represent the greatest improvement.

the effect is to obtain only mild changes while using "torchvision.transforms.functional.adjust_brightness" that lets you specify an exact value.

For this task, we use some functional transformation and a combination of them to improve the result on Tokyo-Night dataset, and based on the result in table VI, the best result belongs to adjust_brightness with brightness_factor equal to 0.1 and using adjust_gamma transformation led to improvement on Tokyo-night dataset too.

V. Conclusion

In this paper, we applied some extensions to improve the result of the baseline paper in different aspects of robustness to night domain and general improvements by applying post processing approach on the results of the baseline model.

For enhance the robustness to changes from day-time image to a night-time image, we implement a combination of functional transformers to change brightness and contrast and saturation and so on, and the results show improvements in Tokyo-night in brightness and gamma transformers.

In the baseline model, used Adam as optimizer and other optimizers are selected because they generalize more effectively. We use AdamW and ASGD optimizers with different learning rate and weight decay to reach out to the higher accuracy of baseline model. The results improve about 5 percent in using ASGD and lr=0.05 and 3 percent in AdamW.

The top-k candidates obtained with the kNN are refined using three different strategies in the post-processing stage. We cluster data using DBSCAN, Agglomerative, and a proposed model, and then reranked the results according to the number of subsets in each cluster. The results show…

The next post-processing technique involves computing descriptors for an image at various resolutions, averaging the descriptor vectors for the image to produce a single vector. The results show a 1 percent improvement in recall5 for downscaling an image in the Tokyo-night dataset and indicate this vector is a richer global descriptor.

we combined these strategies to evaluate the model's performance for the Tokyo-night dataset, and the results demonstrate that their combined performance is higher to that of their individual approach. the recalls are reported in table VII.

For the further improvements we can apply technics to enhance robustness to perspective changes and occlusions and domain adaption to be robust to domain changes.

References

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

[2] Tyagi A, Bansal S, Kashyap A. Comparative analysis of feature detection and extraction techniques for vision-based ISLR system. In2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC) 2020 Nov 6 (pp. 515-520). IEEE.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," Comput. Vis. Image Understand., vol. 110, no. 3, pp. 346–359, Jun. 2008.

[4] Arandjelović R, Zisserman A. Three things everyone should know to improve object retrieval. In2012 IEEE conference on computer vision and pattern recognition 2012 Jun 16 (pp. 2911-2918). IEEE.

[5] H. Je gou, M. Douze, C. Schmid, and P. Pe rez. Aggregating local descriptors into a compact image representation. In Proc. CVPR, 2010.

[6] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisser- man. Object retrieval with large vocabularies and fast spatial matching. In Proc. CVPR, 2007.

[7] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2010, pp. 3384–3391.

[8] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In Proc. ECCV, 2014.

[9] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carls- son. CNN features off-the-shelf: An astounding baseline for recognition. CoRR, abs/1403.6382, 2014.

[10] H.Azizpour,A.Razavian,J.Sullivan,A.Maki,andS.Carls- son. Factors of transferability from a generic ConvNet rep- resentation. CoRR, abs/1406.5774, 2014.

[11] A. Babenko and V. Lempitsky. Aggregating local deep fea- tures for image retrieval. In Proc. ICCV, 2015.

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

[13] A. Khaliq, M. Milford and S. Garg, "MultiRes-NetVLAD: Augmenting Place Recognition Training With Low-Resolution Imagery," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 3882-3889, April 2022, doi: 10.1109/LRA.2022.3147257.

[14] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In IEEE Conf. Comput. Vis. Pattern Recog., 2021.

[15] Siméoni O, Avrithis Y, Chum O. Local features and visual words emerge in activations. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019 (pp. 11651-11660).

TABLE VI
SMART DATA AUGMENTATION WITH FUNCTIONAL TRANSFORMERS

Functional Transformers	Pitts30k		sf-xs		Tokyo-xs		Tokyo-night	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Baseline model	85.9	92.7	34	50.1	59.7	73.3	26.7	43.8
adjust_brightness(0.5)	80.1	89.2	26.3	43.9	58.4	75.2	33.3	50.5
adjust_brightness(0.1)	82.5	90.9	30.2	45.3	61.6	**76.2**	35.2	**51.4**
adjust_brightness(0.15)	83.3	91.7	31.8	46.8	61.6	75.9	**38.1**	50.5
adjust_brightness(0.2)	83.4	91.8	32.5	46.9	59.7	74.9	36.2	47.6
adjust_brightness(0.1)+autocontrast()	84.6	92.5	32.4	37.7	56.8	69.2	24.8	38.1
adjust_contrast(0.1)	81.3	90.2	27.2	43	56.2	70.8	31.4	41
adjust_brightness(0.1) +adjust_contrast(0.1)	70.4	84.5	10.5	17.2	46.7	63.2	21.9	35.2
adjust_brightness(0.1) +adjust_contrast(0.1) +adjust_saturation(0.1)	70	84.2	9.7	17.7	45.1	61	20	30.5
adjust_gamma (3,0.1)	78.4	88.8	21.8	34.9	51.1	68.6	22.9	39
adjust_gamma (1.8,0.1)	81.1	90.1	26.7	42.1	59.4	72.7	35.2	46.7
adjust_gamma (1.5,0.1)	81.7	90.6	27.8	43.1	**61.9**	74.6	37.1	50.5
adjust_gamma (1.2,0.1)	82.4	90.2	28.7	45.4	61	**76.2**	35.2	50.5

[a]Bold numbers represent the greatest improvement footnote.

TABLE VII
PERFORMANCE OF MODELS ON COMBINATION OF SOME EXTENSIONS

Recall	Tokyo-night	
	R@1	R@5
Baseline model	26.7	43.8
multi-scale(down) + AdamW optimizer(scheduler)	31.4	46.7
multi-scale(random crop) + AdamW optimizer(scheduler)	30.5	47.6
reranking(proposed model) + AdamW optimizer(scheduler)	28.6	46.7

[16] Hyeonwoo Noh, Andre Araujo, Jack Sim, and Bohyung Han. Image retrieval with deep local features and attention-based keypoints. In Int. Conf. Comput. Vis., 2017.

[17] Cao B, Araujo A, Sim J. Unifying deep local and global features for image search. InEuropean Conference on Computer Vision 2020 Aug 23.

[18] Chum O, Philbin J, Sivic J, Isard M, Zisserman A. Total recall: Automatic query expansion with a generative feature model for object retrieval. In2007 IEEE 11th International Conference on Computer Vision 2007 Oct 14 (pp. 1-8). IEEE.

[19] Chum O, Mikulik A, Perdoch M, Matas J. Total recall II: Query expansion revisited. InCVPR 2011 2011 Jun 20 (pp. 889-896). IEEE.

[20] Tolias G, Jégou H. Visual query expansion with or without geometry: refining local descriptors by feature aggregation. Pattern recognition. 2014 Oct 1;47(10):3466-76.

[21] Donoser M, Bischof H. Diffusion processes for retrieval revisited. InProceedings of the IEEE conference on computer vision and pattern recognition 2013 (pp. 1320-1327).

[22] Zhang S, Yang M, Cour T, Yu K, Metaxas DN. Query specific rank fusion for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2014 Aug 7;37(4):803-15.

[23] Li Z, Zhou A, Wang M, Shen Y. Deep fusion of multi-layers salient CNN features and similarity network for robust visual place recognition. In2019 IEEE International Conference on Robotics and Biomimetics (ROBIO) 2019 Dec 6 (pp. 22-29). IEEE.

[24] Xin Z, Cai Y, Lu T, Xing X, Cai S, Zhang J, Yang Y, Wang Y. Localizing discriminative visual landmarks for place recognition. In2019 International Conference on Robotics

[25] J. Lemley, S. Bazrafkan, and P. Corcoran. Smart augmenta- tion learning an optimal data augmentation strategy. IEEE Access, 5:5858–5869, 2017.

[26] T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid. A bayesian data augmentation approach for learning deep mod- els. In Advances in Neural Information Processing Systems, pages 2794–2803, 2017.

[27] T. DeVries and G. W. Taylor. Dataset augmentation in fea- ture space. arXiv preprint arXiv:1702.05538, 2017.

[28] Perez L, Wang J. The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621. 2017 Dec 13.

[29] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 113–123, 2019.

[30] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. ´ Pajdla. 24/7 place recognition by view synthesis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(2):257–271, 2018.

[31] D. M. Chen, G. Baatz, K. Koser, S. S. Tsai, R. Vedantham, ¨ T. Pylvan¨ ainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, ¨ B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In IEEE Conference on Computer Vision and Pattern Recognition, pages 737–744, 2011.

[32] Llugsi R, El Yacoubi S, Fontaine A, Lupera P. Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito. In2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM) 2021 Oct 12 (pp. 1-6). IEEE.

[33] Marinov M, Kalmukov Y, Valova I. Content-Based Image Retrieval: Impact of image resolution on the search accuracy and results ordering. In2021 International Conference Automatics and Informatics (ICAI) (pp. 72-75). IEEE.

[34] Berton G, Mereu R, Trivigno G, Masone C, Csurka G, Sattler T, Caputo B. Deep visual geo-localization benchmark. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022 (pp. 5396-5407).