



Semantic-guided de-attention with sharpened triplet marginal loss for visual place recognition



Seung-Min Choi^{a,b,*}, Seung-Ik Lee^{a,c}, Jae-Yeong Lee^{a,c}, In So Kweon^b

^aArtificial Intelligence Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), 218 Gajeong-ro, Yuseong-gu, Daejeon, 34129, Republic of Korea

^bDivision of Future Vehicle, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon, 34141, Republic of Korea

^cUniversity of Science and Technology (UST), 217 Gajeong-ro, Yuseong-gu, Daejeon, 34113, Republic of Korea

ARTICLE INFO

Article history:

Received 23 August 2022

Revised 4 April 2023

Accepted 25 April 2023

Available online 2 May 2023

Keywords:

Visual place recognition

Image retrieval

Triplet marginal loss

Attention

De-attention

Semantic guidance

Semantic segmentation

ABSTRACT

Thanks to Earth-level Street View images from Google Maps, a visual image geo-localization can estimate the coarse location of a query image with a visual place recognition process. However, this can get very challenging when non-static objects change with time, severely degrading image retrieval accuracy. We address the problem of city-scale visual place recognition in complex urban environments crowded with non-static clutters. To this end, we first analyze what clutters degrade similarity matching between the query and database images. Second, we design a self-supervised trainable de-attention module that prevents the network from focusing on non-static objects in an input image. In addition, we propose a novel triplet marginal loss called *sharpened triplet marginal loss* to make feature descriptors more discriminative. Lastly, due to the lack of geo-tagged public datasets with a high density of non-static objects, we propose a clutter augmentation method to evaluate our approach. The experimental results show that our model has notably improved over the existing attention methods in geo-localization tasks on the public benchmark datasets and on their augmented versions with high population and traffic. Our code is available at https://github.com/ccsmm78/deattention_with_stml_for_vpr.

© 2023 The Author(s). Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

1. Introduction

Visual place recognition (VPR) has lately been studied as a critical component for visual localization on a city scale, thanks to the Street View of Google Maps. It retrieves the most similar image from a Street View database for a given query image and takes the geographic pose of the matched image as the location of the query [1]. State-of-the-art deep learning approaches have remarkably succeeded in city-scale visual place recognition [2]. Despite their partial success in some lighting, scale, and viewpoint variations, securing robustness to such changes still remains extremely challenging when the images are highly cluttered with moving objects. In such cases, it would be highly beneficial to eliminate these temporary (or dynamic) cluttered parts of the scene so that the permanent regions that do not change over time can be more focused for better results in visual place recognition tasks. Over the past years, several data-driven attention approaches have been introduced to reduce the influence of dynamic local features. For ex-

ample, CRN [3] presented spatial attention with a two-dimensional mask; SENet [4] applied channel attention with a squeezing and expansion mechanism; BAM [5] and CBAM [6] utilized both the channel and spatial attention simultaneously.

Although these data-driven attention methods help the network focus on relatively important and frequent landmark features by utilizing statistical information from given databases, this statistical information, in practice, can not be easily obtained in many cases due to the insufficient number of datasets, causing the model to fail the retrieve the matched image for a given query, as shown in Fig. 1. A pre-trained model is widely employed to alleviate the insufficient dataset problem. Since it is usually trained as an object classifier, it tends to highlight a broad range of object classes included in labels. For example, a pre-trained model with ImageNet emphasizes objects such as humans, animals, cars, airplanes, plants, etc. This phenomenon remains when conventional attention is applied to the model, as the car object is still highlighted (red) in the bottom right of Fig. 1. We have observed that these dynamic objects are not helpful or harmful to image retrieval applications.

We propose a novel approach (shown in Fig. 2), called *de-attention*, to overcome the drawbacks of existing attention-based

* Corresponding author.

E-mail address: ccsmm@etri.re.kr (S.-M. Choi).

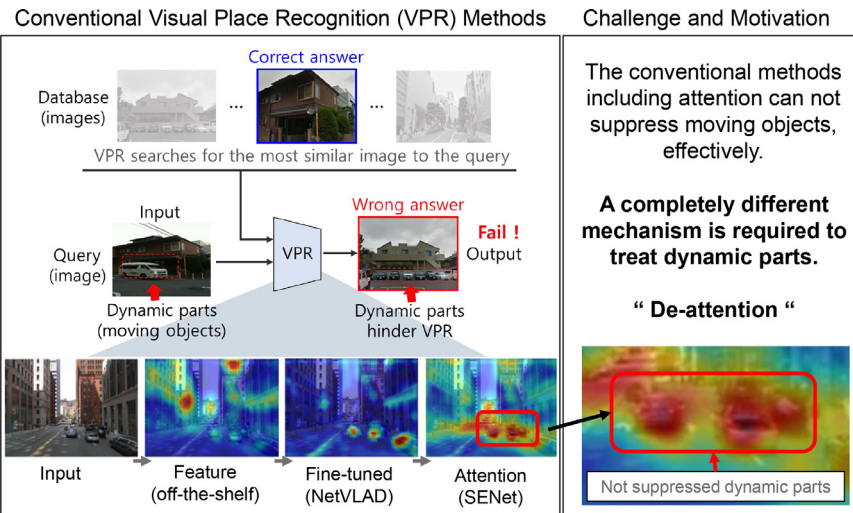


Fig. 1. Motivation: Conventional attention does not effectively suppress the dynamic parts.

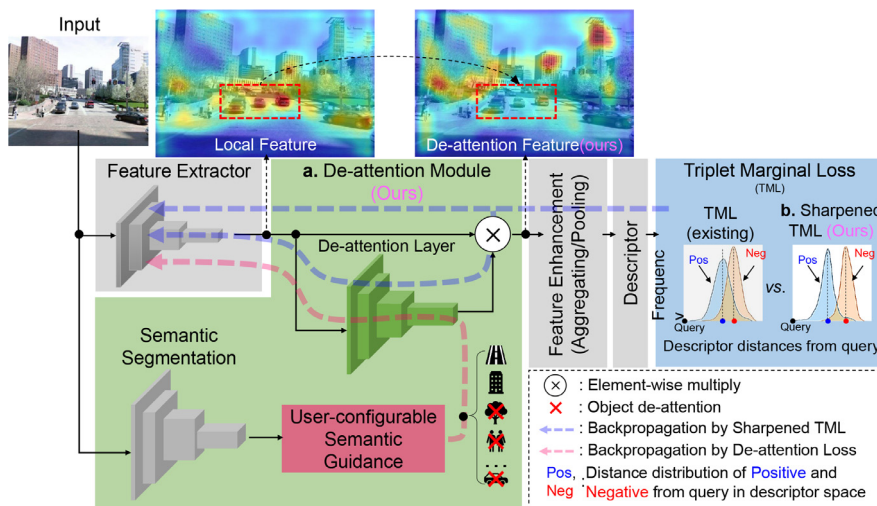


Fig. 2. The proposed de-attention and sharpened triplet marginal loss for visual place recognition First (a.), the de-attention layer is trained by user-configurable semantic guidance specifying what features are unnecessary at the object level. It gives low weights to features on the clutter objects and high importance to features on landmark objects, as shown in a heatmap output of the de-attention layer. Second (b.), with our sTML, the feature distance between the query-positive (blue-colored) and the query-negative (red-colored) increases, and each standard deviation decreases, so the overlapping area is minimized, leading to enhanced recall results.

methods. It excludes dynamic clutters explicitly from the local features, in contrast to the attention on the dynamic clutters of the conventional data-driven attention-based approaches. While the conventional data-driven attention methods are purely dependent on the training data, our de-attention mechanism allows the users to configure what features should be suppressed, for example, the feature of humans or vehicles. Our de-attention module is trained not to focus on a set of user-configured predefined object classes illustrated as semantic guidance in Fig. 2. So that features of static objects such as buildings are more focused with our de-attention (See the heat-map on the top-right in Fig. 2).

In addition, we propose a *sharpened triplet marginal loss (sTML)* for better separation between the query-positive and the query-negative groups by further improving the discriminability of the learned descriptors. Specifically, the mean distance of the two feature groups increases, and the standard deviation of each group decreases, resulting in better (sharper) discrimination as shown in the loss function graph in Fig. 2. Lastly, due to the lack of geo-tagged public datasets of highly crowded non-static objects, we propose a clutter augmentation method for our evaluations.

In summary, we propose a novel feature de-attention scheme to suppress the effects of dynamic clutter objects of images in match-

ing a query with a database, a joint loss function of sTML and de-attention for better representation of the scene, better discrimination between the positive and negative groups in terms of contrast learning, and finally, a data augmentation scheme to get more cluttered scenes for the evaluation of our approach. This literature is structured as follows. Section 2 introduces the latest related research. Section 3 presents the proposed model and the training strategies. In Section 4, implementation details and experimental results are described. In Section 5, our observations and some frequently asked questions are discussed. Finally, we conclude the paper in Section 6.

2. Related works

2.1. Attention methods for visual place recognition

A visual attention mechanism highlights more important parts of the local features extracted from images. CRN [3] introduces a two-dimensional re-weighting mask computed by channel reduction using multi-sized convolutional layers for spatial attention. SENet [4] achieves channel attention with fully connected layer-based channel squeezing and expansion. BAM [5] and sim-

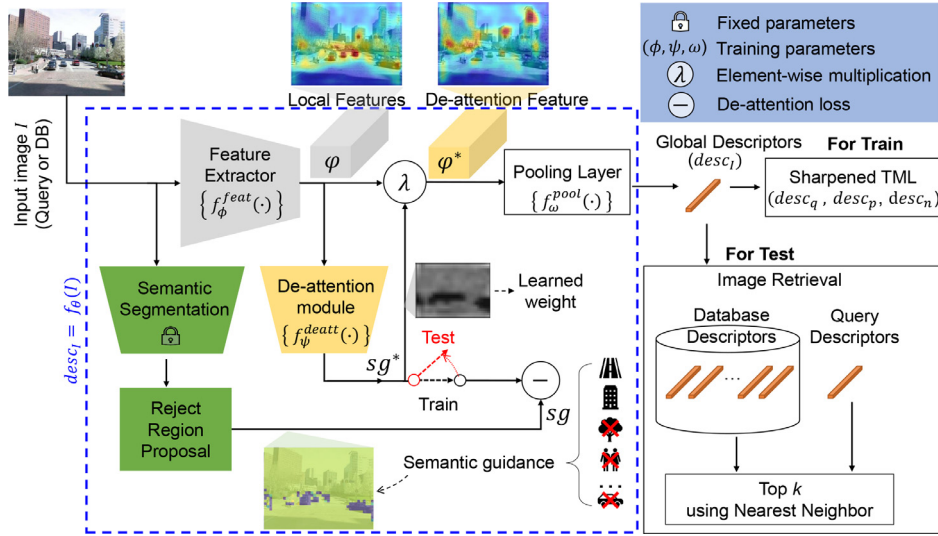


Fig. 3. An Architecture of the proposed approach. Our VPR network is trained to suppress the influence of dynamic parts in the local features obtained from the feature extractor by the de-attention module. It is a self-supervised trainable network because the ground truth semantic guidance is automatically generated from the input image.

ple BAM [7] add a convolutional layer for spatial attention to a SENet-like model to calculate the channel and spatial attention weights simultaneously. CBAM [6] is a compact version that reduces computations and hardware resources with sequential cascading channels and spatial attention models. Attentive weights can also be learned based on the statistical importance of features. PWA [8] utilizes channel-direction variance of local features extracted from all DB images as a re-weighting filter so that the features of frequently appearing objects such as buildings and roads are more emphasized. DeLF [9] and DELG [10] learn attention scores using GPS coordinates and visual features from all database images to select key points of local landmarks.

Although these data-driven attention methods help the network focus on relatively important and frequent landmark features by utilizing statistical information from given databases, most non-static objects with insufficient statistical information may remain under insufficient datasets or extremely crowded environments.

2.2. Marginal loss for metric learning

Distance metric learning (simply metric learning) allows us to construct task-specific distance metrics from annotated data. We can then retrieve DB images in VPR tasks using the learned distance metric. A marginal loss is the most frequently used metric learning and has been successfully adopted. For example, [11] binarizes the descriptors and optimizes their Hamming distance with a contrastive loss. [12] uses a quadruple loss to increase the distance between negative groups for the person re-identification task. [13] proposes an N-pair loss for multi-class classification tasks that uses all other classes' positives for the current query's negatives. [14] uses a triplet marginal loss (TML) to train a Siamese network to separate positive and negative images from a query image by a certain margin in Euclidean space. In particular, since the TML has been successfully utilized in many VPR studies as an efficient loss function for metric learning, we use it as our model training loss.

However, contrary to the TML operation that increases the query-negative distance, we have observed that the query-negative distance decreased as much as the query-positive distance decreased, resulting in reduced discrimination. To avoid this distance collapsing between the query and negative images, we propose a new triplet margin loss function with an additional term to compensate for it.

3. Method

We follow the same standard VPR pipeline [14], in which the deep local features extracted from each image are transformed into more compact and representative descriptors through the pooling process. For the pipeline, three groups of input images are required: a query image q , positive DB images $\{p_i^q\}$, and negative DB images $\{n_j^q\}$ where i and j are DB image indexes. For a given q , $\{p_i^q\}$ must be geographically close and have a similar scene to q , while $\{n_j^q\}$ only needs to be geographically distant from q . Then, the descriptors of query $desc_q$, positive $desc_p$, and negative $desc_n$ are inferred independently by a Siamese network. At training, the shared weights of the Siamese network are trained with the distance relationship of the descriptors $desc_q$, $desc_p$, and $desc_n$. At inference, the similarity between the descriptors of the query and the DB images are measured to retrieve top- k DB images most similar to the query.

A trainable network f_θ depicted in Fig. 3 transforms them into global descriptors, $f_\theta(q)$, $f_\theta(p_i^q)$ and $f_\theta(n_j^q)$ with the training parameter θ of the whole network. Here, θ includes three submodule training parameter groups, ϕ of the feature extractor, ψ of the de-attention module, and ω of the pooling layer. While the pooling layer is optimized with the sharpened TML, the feature extractor and de-attention layer are jointly trained by de-attention loss and sharpened TML. The weight parameters of all modules are updated by learning, except for the semantic segmentation layer marked with a lock. Note that the process to create semantic guidance, sg , is not required for a test time. We can formulate the forwarding process to generate $desc_i$ from a given image I as follows.

$$desc_i = f_\theta(I) = f_\omega^{pool}(\lambda(f_\phi^{feat}(I), f_\psi^{deatt}(f_\phi^{feat}(I)))), \quad (1)$$

where the symbols are defined in Table 1, which will be described in detail in the following sections.

3.1. Deep local feature extractor

Given the image I , the $H \times W \times D$ dimensional deep local feature ϕ_I is created by the local feature extractor $f_\phi^{feat}(I)$, where H , W , and D are the height, the width, and the number of the channel of the local feature, respectively. We employ the activations

Table 1
Symbol Definitions.

Symbols	Definition	Symbols	Definition
$q, desc_q$	A query image and descriptor	$f_{\omega}^{pool}(\cdot)$	The pooling layer function
$p_i^q, desc_{p_i^q}$	i th positive image of q and descriptor	$\lambda(a, b)$	An element-wise multiplication across channels
$n_j^q, desc_{n_j^q}$	j th negative image of q and descriptor	φ_l	The local feature of an image l
$f_{\theta}(\cdot)$	The whole network function	φ_l^*	The attentive local feature of the image l
$f_{\phi}^{feat}(\cdot)$	The local feature extractor function	sg_l	The semantic guidance of the image l
$f_{\psi}^{deatt}(\cdot)$	The de-attention module function	sg_l^*	The estimated semantic guidance of l

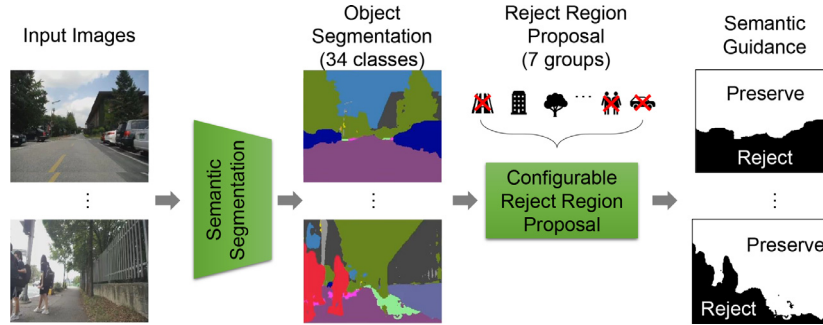


Fig. 4. Semantic Guidance Generation. We classify all pixels of an input image into seven groups in Table 2 and then convert each of them into two values of *preserve* (1) or *reject* (0) to create a ground truth mask called *semantic guidance*.

of the convolutional layer of VGGNet [15] pre-trained with ImageNet [16] as the deep local features φ_l . For better performance, we removed the following ReLU non-linearity [17] in the fifth convolutional layer. We also excluded the max pooling of the last layer to prevent a severe reduction of spatial information.

3.2. De-attention with semantic guidance

In this section, we introduce the de-attention mechanism that reduces the weight of non-landmark objects with semantic guidance. Because these non-landmark features interfere with image retrieval, we want to exclude them explicitly before the pooling process. To this end, we propose the de-attention method that decreases the weight of non-landmark objects with semantic guidance in which weights are determined according to their known motion characteristics.

3.2.1. Semantic guidance

Since landmarks such as buildings and roads are usually fixed to the ground, we use these static properties of objects to determine which objects are to be preserved and which are to be rejected. We first segment the objects from the input image using DeepLabv3plus [18] model trained with the Cityscape dataset [19]. As shown in the object category column of Table 2, there are thirty-four types of object classes in the Cityscape dataset. We again classify these into seven groups of objects such as *road*, *building*, *human*, *vehicle*, *vegetation*, *sky*, and *other objects* according to the known movement characteristics. To generate semantic guidance through the rejection area suggestion function, we set the rejection policy of each object group appropriate for the application. Fig. 4 shows the entire process of obtaining semantic guidance. Input image pixels are classified as one of thirty-four classes by the *semantic segmentation*. After we group them into seven objects again, they have a value of zero (*reject*) or one (*preserve*) according to the user-configurable reject policy in the *reject region proposal*. Note that for the (*optional*), we have to decide its policy to be either (*preserve*) or (*reject*) before training, depending on the dataset or application domain. Based on the preliminary experiments, we select *preserve* for vegetation (See Section 5.2). Lastly, the result of *semantic segmentation* is converted to a two-

dimensional mask, we call it *semantic guidance* (sg), which is used as the ground truth for learning the de-attention layer.

3.2.2. De-attention layer design

Our de-attention module is based on the contextual re-weighting network [3] that has multi-kernel convolution layers for channel reduction and concatenation, as shown in Fig. 5. Convolution operations with kernel sizes of 3, 5, and 7 output 32, 32, and 20 channel features, respectively, and the outputs are concatenated (stacked) into 84 channel features again. Finally, they are converted into a two-dimensional mask by a 1-channel convolutional layer followed by a *sigmoid* function. We can then formulate the above forwarding process of the de-attention as follows. Given local features denoted by φ ,

$$sg^* = f_{\psi}^{deatt}(\varphi). \quad (2)$$

$$\varphi^* = \lambda(\varphi, sg^*), \quad (3)$$

where ψ is the learnable parameter of de-attention layer $f_{\psi}^{deatt}(\cdot)$, and φ^* denotes re-weighted local features, and $\lambda(a, b)$ is a function that performs element-wise multiplication across all channels (D) of the $H \times W \times D$ dimensional a with the $H \times W \times 1$ dimensional b .

3.3. Pooling layer

The pooling layer converts the re-weighted local features into global descriptors. We mainly employ the NetVLAD model used in [14] for the pooling layer which will be introduced in Section 4.1.

3.4. Loss function

The triplet marginal loss (TML) has been successfully utilized in lots of visual place recognition studies as an efficient loss function for metric learning. To explain it, we first define the distance $d_{\theta}(I_1, I_2)$ between two images I_1 and I_2 as Euclidean distance between the corresponding image descriptor $f_{\theta}(I_1)$ and $f_{\theta}(I_2)$ as follow:

$$d_{\theta}(I_1, I_2) = \|f_{\theta}(I_1) - f_{\theta}(I_2)\|_2 \quad (4)$$

Table 2

Look-up table for Semantic Guidance. It is a look-up table that defines reject policy according to the time-varying characteristics of objects known to humans.

No.	Object Group	Object Category	Characteristic	Reject Policy
1	Road	Road, sidewalk, guard rail	Static	Preserve
2	Building	Building, wall, fence	Static	Preserve
3	Human	Person, rider	Time-varying	Reject
4	Vehicle	Car, Bus, truck, caravan, trailer, motorcycle, bicycle	Time-varying	Reject
5	Vegetation	Tree, plant	Season-varying	Optional
6	Sky	Sky	Time-varying	Optional
7	Other objects	Bridge, tunnel, pole, terrain traffic sign, rail track, etc.	Static	Preserve

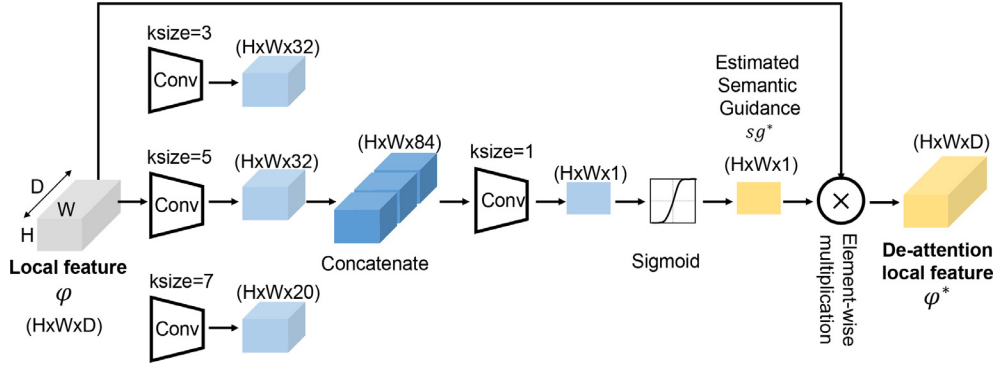


Fig. 5. De-attention Layer. The modified contextual re-weighting network [3] is used as a de-attention layer in which the downsampling ratio and the number of channel reductions are modified to keep spatial information.

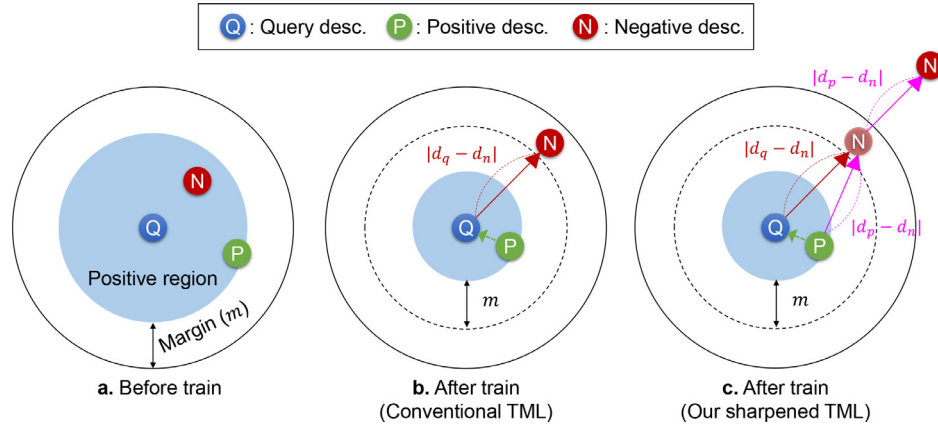


Fig. 6. Graphic description of the proposed sharpened triplet marginal loss. In the descriptor space with the query as the origin, it is desirable that positives get closer to the origin and negatives get further away by learning. With the triplet marginal loss (b), positives get close to the query, and negatives move away from positives by m . And with the proposed sharpened triplet marginal loss (c), the negatives are farther away from the query as the negatives and positives push each other again.

Then the TML in Eq. (5) indicates that the distance $d_\theta(q, n_j^q)$ between a query q and a negative n_j^q should be larger than the distance $d_\theta(q, p_i^q)$ between a query q and a positive p_i^q by a certain margin m in global descriptor space.

$$L_\theta^{TML} = \sum_{i,j} \max((d_\theta(q, p_i^q) + m - d_\theta(q, n_j^q)), 0), \quad (5)$$

where m is a user-defined margin. By minimizing Eq. (5), $d_\theta(q, p_i^q)$ is forced to decrease and $d_\theta(q, n_j^q)$ to increase. However, as $d_\theta(q, p_i^q)$ decreases, $d_\theta(q, n_j^q)$ may also decrease as long as the margin m is not zero, which in turn may cause the degradation of feature discrimination. To avoid this distance collapsing between the query and negative, we add an extra term to Eq. (5) that forces the distance between the positive and the negative to increase (see Fig. 6). We call it *sharpened triplet marginal loss (sTML)* and define

it as follows.

$$L_\theta^{sTML} = \sum_{i,j} \max((d_\theta(q, p_i^q) + m' - d_\theta(q, n_j^q) - d_\theta(p_i^q, n_j^q)), 0), \quad (6)$$

$m' > m,$

where m' is the modified margin and m is the original margin as in Eq. (5). m' is set to be larger than m in order to compensate for the additional negative term, $-d_\theta(p_i^q, n_j^q)$. As a result, we expect that the distance between the positive and the negative can be further increased by sTML as illustrated in Fig. 6c.

Next, de-attention loss, L^{deatt} , is required to train the de-attention layer. We use the mean squared error (MSE) function for the de-attention loss as in Eq. (7).

$$L_{(\phi, \psi)}^{deatt} = \frac{1}{n} \sum_{i=1}^n (sg_i - sg_i^*)^2, \quad (7)$$

where sg_i is pixel-wise semantic guidance (the ground truth), sg_i^* is the de-attention layer output of $[0, 1]$, i is a pixel location and n is

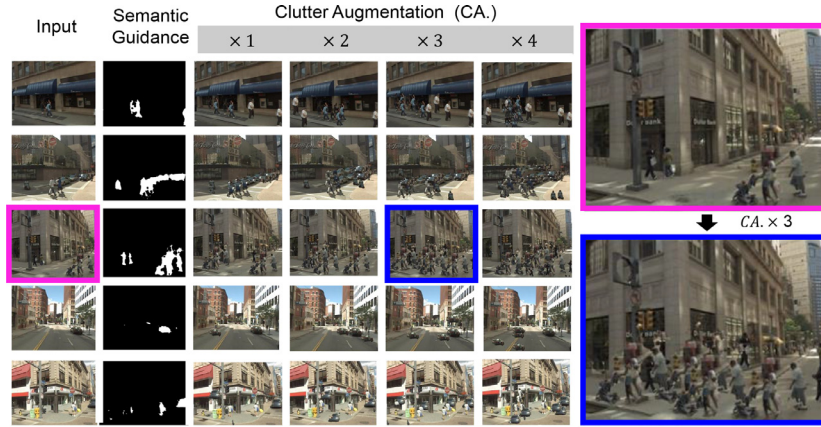


Fig. 7. Clutter Augmentation. Segmented objects in the scene of an existing public dataset are augmented at arbitrary locations with random sizes to create virtual crowded geotagged public datasets.

the size of sg , $H \times W$. For end-to-end learning, the pooling layer is trained by sTML with distance metric, while the feature extractor and de-attention layer are jointly trained by sTML in Eq. (6) and de-attention loss in Eq. (7). Then, the total loss is given by

$$L_{\theta} = L_{\theta}^{TML} + \alpha * L_{\{\phi, \psi\}}^{deatt}, \quad (8)$$

where α is a weight-balancing hyper-parameter for the two loss terms.

4. Experiments

4.1. Implementation

Our VPR system is implemented using pytorch [20] with the original baseline code from [21]. In detail, we employ the convolutional layers of VGG16 and AlexNet pre-trained with ImageNet as deep local feature extractors. And we use Max, Average, GeM [10], and NetVLAD as the pooling layers, where the descriptor dimension in NetVLAD is 32,768, and the dimensions of the others are 512. For the feature clustering required for NetVLAD pooling, we use the k -means algorithm [22]. Given images, their semantic segmentation is generated by MobileNet-based DeepLabv3 plus [18] trained with Cityscape dataset [19]. Lastly, we retrieve the top- k most similar images to the query from the database using the k nearest neighbor (k -NN) method. To implement the k -NN algorithm and the k -means algorithm for the feature clustering, we use Faiss library [23].

4.1.1. Implementation details

VGG16 is cropped at the last convolutional layer (conv5) before the last ReLU. We add descriptor-wise L2-normalization layer after conv5. The number of clusters used in NetVLAD experiments is $K = 64$. We use the margin $m = 0.1$, $m' = 1.5$ for Eq. (6), learning rate = 0.001, which is halved every five epochs, momentum 0.9, weight decay 0.001, batch size of 4 tuples (query, positives, negatives), and train for at most 30 epochs but convergence usually occurs much faster. The network which yields the best recall@1 on the validation set is used for testing. We followed the same mining protocol used in [14] to create the training tuple for a query, where given q in each learning step, $(q, \{p_i^q\}, \{n_j^q\})$ pairs are automatically packed in the dataloader.

4.2. Preparing dataset

4.2.1. Clutter augmentation

To measure the performance of the proposed method, it would be necessary for the test dataset to have enough amount of clut-

ters. However, there are insufficient geotagged public datasets rich in clutter objects (vehicles, pedestrians, vegetation) in complex urban city environments. To overcome this, we augment the existing public geotagged datasets by adding clutters, which we call Clutter Augmentation. Specifically, we first detect clutters in each image and add them to arbitrary locations in the image, where we use the semantic segmentation model used in semantic guidance for clutter detection. Next, we detect the contour lines of the clutters with a contour finding method [24] and crop the clutters along the contour lines as a small image patch shape. And we use clutters of size over 20×20 pixels for natural composition among the clutter patches. Then, they can be re-scaled up to $0.9 \sim 1.1$ of the original size and are pasted into arbitrary locations whose x -coordinates are random values in the image, but y -coordinate is constrained within 30% of its original value. This is because we want to paste the clutters into appropriate contextual regions in the image, for example, a car near the road but not in the air or on the buildings. The number of added clutters ranges from $\times 1 \sim \times 4$ times per clutter. Fig. 7 show some examples of our approach.

4.2.2. Dataset

Geotagged public datasets such as Pittsburgh 30K train/test/val (validation) [14], TokyoTM (TimeMachine) train/test [14], Tokyo24/7 test [25] are used to train and to test for city-scale VPR tasks.

The size of each dataset is described in Fig. 8. For convenience, dataset- O denotes $No.O$ dataset in Fig. 8. Because the same image or images taken on the same day are included in the query and database in TokyoTM (dataset-9, 10), DB images acquired on the same day as the query are removed from DB in TokyoTM for a fair evaluation. Next, in the case of Tokyo24/7 (dataset-14), depth-based scene synthesis [25] is not applied to our experiment. The datasets with clutter augmentation (CA) are notated as *+clutter augmentation* in Fig. 8, where the existing public datasets can become clutter-rich congested datasets, as shown in Fig. 7. And (DB, Q) means that the CA is applied to both DB and query images, and (Q) means that the CA is applied only to the query images.

4.3. Results

4.3.1. Performance

Following [3,14], we evaluate the recall performance of the proposed method. We first search for the top- k images most similar to the given query image in all database images without any geolocation. Next, in UTM coordinates, if any of the retrieved k images are located within 25 meters of the query, recall at k is evaluated as True otherwise, it is evaluated as False. All networks are trained on the Pittsburgh-train dataset (dataset-1 in

No.	Dataset name	The number of images	
		Database(DB)	Query (Q)
1	Pitts30k-train	10,000	7,416
2	+ CA. on (DB, Q)	10,000	7,416
3	Pitts30k-val (normal test)	10,000	7,608
4	+ CA. on (Q)	10,000	7,608
5	+ CA. on (DB, Q)	10,000	7,608
6	Pitts30k-test (normal test)	10,000	6,816
7	+ CA. on (Q)	10,000	6,816
8	+ CA. on (DB, Q)	10,000	6,816
9	Tokyo TM-train	49,104	7,277
10	+ CA. on (DB, Q)	49,104	7,277
11	Tokyo TM-test (normal test)	49,056	7,186
12	+ CA. on (Q)	49,056	7,186
13	+ CA. on (DB, Q)	49,056	7,186
14	Tokyo 24/7-test (normal test)	75,984	315
15	+ CA. on (Q)	75,984	315
16	+ CA. on (DB, Q)	75,984	315

Fig. 8. Dataset size. CA is the abbreviation of *Clutter Augmentation*. We apply CA ($\times 3$) to every dataset. Each dataset consists of original public data and their clutter-augmented version, where CA can be applied only to the query (Q) or can be applied to the Q and a database (DB) simultaneously.

Fig. 8). And evaluations are performed on the public datasets and their clutter-augmented versions. We select one model each from the local extractor and pooling layer for baseline model selection. AlexNet or VGG16 are deep feature extractors, followed by Average, Max, GeM, or NetVLAD as pooling layers. As the recall at 1 of the VGG16+NetVLAD shows the highest value of 0.85 among the local feature and pooling layer combinations, we select the VGG16+NetVLAD combination as the baseline model.

Fig. 9 is a graph comparing the recall results of the existing VPR model and our model, where all values in the graph are from our own experimental results. It shows recalls evaluated with Pittsburgh dataset-3 and Tokyo24/7 dataset-14 with the models trained with the Pittsburgh train dataset-1. For recall at top- k , the x -axis represents k values (1,2,3,4,5,10,15,20,25), and the y -axis represents recall (%). Training proceeds until there is no further improvement in top-1 recall, usually ending before 30 epochs. Because the VGG16 and feature embedding-based NetVLAD pooling layer combination shows better performance than other combinations of local features VGG16 (or AlexNet) and direct pooling layers (Max, Average, or GeM), we select the VGG16+NetVLAD as the baseline. When the existing channel and spatial attention methods such as SENet, BAM, CBAM, or CRN are employed to the baseline, 2 ~ 3% points of recalls are improved. On the other hand, the proposed method improves them about twice for each dataset. And, the performance improves better on the Tokyo24/7 dataset, which is crowded with people and vehicles, compared to the non-crowded Pittsburgh.

Next, the recall results at 1 and 5 for all test datasets are listed in Table 3. Comparing our proposed methods to existing attention methods (SENet, BAM, CBAM, and CRN), we are about 3% points and 8% points ahead of them in dataset-3 and dataset-14, respectively. Improvement is observed from a minimum of 1% to a maximum of 10% for most datasets. In particular, a greater performance improvement is observed in the dataset-4,5,7,8,12,13,15,16 crowded with non-static objects by clutter augmentation, compared to the normal dataset-3,6,11,14. In addition, our model achieves the best in the top 1 recall in most data sets, which has the advantage of reducing post-processing operations such as re-ranking and query expansion [27].

4.3.2. Visualization of de-attention

Fig. 10 shows a visualization of the results of some processing steps in our approach. (b) is the input images. (b) is the semantic guidance, a mask data of 0 and 1. And (c) is the predicted

semantic guidance or de-attention weight. If we train the model with only the de-attention loss, the prediction converges to the ground truth value between (0 ~ 1), but the gradient of sTML is also backpropagated to the de-attention layer, so the model has an appropriate prediction value. Therefore, note that the two groups of probing lines (solid vs. dotted) show similar patterns shown in (c). With a higher balancing hyper-parameter α in Eq. (8), the semantic guidance would be estimated more clearly, but the impact of sTML decreases, leading to performance deterioration. Experimentally, optimal results were obtained when α is 0.001. Because the de-attention layer consists of a small model compared to DeepLabv3plus [18], which is used for creating semantic guidance, blurred de-attention results are predicted compared to semantic guidance. Nevertheless, it often detects non-static objects not detected in semantic guidance for de-attention training ground truth, for example, the excavator pointed by the purple circle. Also, vehicles and people are highly attentive before de-attention as is (e), but they do not get attentive after the de-attention module. Instead, landmarks such as buildings and roads get highly attentive, as shown in (f). Given image I and its local feature $\psi_I \in \mathbb{R}^{C \times H \times W}$, the heatmaps \mathbf{e}, \mathbf{f} are obtained from Eq. (9) similar to [15].

$$\mathbf{F}_{HeatMap}(\psi_I, I) = Superimpose(Normalize(\mathbf{F}_{AvgCh}(\psi_I \times \mathbf{F}_{Avg2d}(\psi_I))), I), \quad (9)$$

where $\mathbf{F}_{Avg2d}(\cdot)$ and $\mathbf{F}_{AvgCh}(\cdot)$ are average the local feature in the spatial direction and channel direction, respectively. And *Normalize* normalizes input between 0 and 1, and *Superimpose*(\cdot, I) overlays (\cdot) on the input image I . Since there is no classification layer in VPR, we use the heatmap method instead of general visualization functions such as class activation map (CAM) [28].

4.3.3. Sharpened triplet marginal loss

We classified the database images into positive and negative groups with trained models to observe the effect of our sharpened triplet marginal loss (sTML). Then we drew a histogram of each group and compared their mean and standard deviation. To distinguish the two groups well, it is desirable that the distance between the centroid of the two groups is far and the standard deviation of each group is small so that the overlapping area is minimized. Fig. 11 is a histogram of the descriptor distance between query and database images. We found that the optimal TML margin m is 0.1 and the optimal sTML margin m' is 1.5 from experiments with dataset-1. To compare the distance histogram with various margins, we set the margin m for triplet marginal loss varies from 0.05 to 0.3. And we set the margin m' for the sharpened triplet marginal loss as 1.5, where the margin between query and negative and the margin between positive and negative is set to be 0.1 and 1.4, respectively.

Fig. 11a is a distance distribution with the off-the-shelf model, and b~e are the distributions of models trained with triplet marginal loss with varying margins. And f results from our sharpened triplet marginal loss. In each graph, the narrower the overlapping area of the two distance distributions (for example, the yellow area of c), the better the descriptor distinction. Fig. 11c is the best result of the model learned with existing TML, where Δm is 0.11, and σ_p and σ_q are 0.09 and 0.08, respectively. And, Fig. 11f is the result of our sharpened triplet marginal loss (sharpened TML), where Δm increases to 0.15, σ_p and σ_q are 0.07 and 0.05, respectively. Hence, the overlapping area of the two groups is the smallest in all graphs in Fig. 11, leading to the best recall result. Note that if only margin m is increased without the term $d_\theta(p_i^q, n_j^q)$ of Eq. (5), the performance improvement is negligible or rather deteriorated. For example, Fig. 11e has the highest Δm , 0.26, but the standard deviations are also high with the large overlapping areas, leading to the low recall results than others. And note that the

Table 3
Benchmark table for various models with public datasets and their clutter augmented version This is a benchmark table comparing the recall results of the existing VPR model and our model, where all values in the Table are from our own experimental results. CA, DB, and Q are abbreviations of clutter augmentation, database, and query, respectively. And BL, DA, and sTML denote the baseline model, the proposed de-attention, and the sharpened triple marginal loss. All models are trained with the Pittsburgh train dataset-1, and they are tested with various datasets-3 ~ 8 and dataset-11 ~ 16. **Bold** font means the best results in every column. Our models achieve the best at the top-1 recall in most datasets.

Test dataset CA type No. in Fig. 8a	Pitts30k-val						Pitts30k-test						TokyoTM						Tokyo24/7					
	Normal (No CA)		+ CA on (Q)		+ CA on (DB,Q)		Normal (No CA)		+ CA on (Q)		+ CA on (DB,Q)		Normal (No CA)		+ CA on (Q)		+ CA on (DB,Q)		Normal (No CA)		+ CA on (Q)		+ CA on (DB,Q)	
	3		4		5		6		7		8		11		12		13		14		15		16	
ModelsRecall	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5
VGG16 [15] (V) + AVGpool	0.69	0.87	0.45	0.68	0.48	0.72	0.65	0.83	0.48	0.67	0.49	0.71	0.69	0.82	0.44	0.59	0.48	0.64	0.23	0.39	0.14	0.26	0.12	0.26
V + MAXpool	0.73	0.89	0.59	0.81	0.56	0.79	0.72	0.87	0.59	0.78	0.57	0.78	0.74	0.85	0.59	0.74	0.55	0.71	0.40	0.59	0.29	0.46	0.28	0.47
V + GEMpool [26]	0.75	0.89	0.57	0.79	0.55	0.78	0.70	0.86	0.53	0.71	0.55	0.76	0.76	0.86	0.55	0.61	0.54	0.70	0.36	0.52	0.27	0.41	0.21	0.38
AlexNet + NetVLAD [14]	0.82	0.93	0.80	0.92	0.78	0.91	0.79	0.90	0.77	0.88	0.75	0.87	0.88	0.94	0.83	0.90	0.81	0.89	0.44	0.57	0.39	0.53	0.36	0.49
V + NetVLAD [14]																								
(BL)	0.85	0.94	0.77	0.91	0.71	0.86	0.81	0.91	0.73	0.86	0.70	0.84	0.91	0.95	0.81	0.89	0.77	0.86	0.57	0.72	0.45	0.63	0.35	0.52
BL + CRN [3]	0.87	0.94	0.85	0.94	0.84	0.93	0.84	0.92	0.82	0.90	0.80	0.89	0.91	0.95	0.87	0.93	0.85	0.91	0.60	0.74	0.55	0.68	0.51	0.64
BL + BAM [5]	0.87	0.94	0.85	0.94	0.83	0.93	0.83	0.92	0.80	0.90	0.78	0.89	0.90	0.94	0.87	0.92	0.85	0.91	0.58	0.69	0.52	0.64	0.49	0.62
BL + CBAM [6]	0.85	0.94	0.81	0.92	0.80	0.92	0.82	0.91	0.78	0.89	0.76	0.88	0.85	0.91	0.77	0.86	0.76	0.86	0.46	0.63	0.37	0.60	0.41	0.56
BL + SENet [4]	0.87	0.95	0.85	0.95	0.83	0.93	0.83	0.92	0.80	0.89	0.78	0.89	0.91	0.95	0.87	0.93	0.85	0.91	0.59	0.75	0.50	0.68	0.53	0.65
BL + DA (ours)	0.88	0.95	0.86	0.95	0.84	0.94	0.83	0.92	0.81	0.90	0.79	0.89	0.91	0.95	0.87	0.93	0.86	0.92	0.65	0.77	0.59	0.72	0.55	0.71
BL + DA + sTML (ours)	0.89	0.96	0.88	0.96	0.86	0.95	0.85	0.92	0.83	0.91	0.80	0.89	0.92	0.95	0.88	0.93	0.85	0.91	0.64	0.76	0.58	0.73	0.51	0.67
BL + DA + sTML + SENet (ours)	0.90	0.96	0.88	0.96	0.86	0.95	0.84	0.92	0.82	0.90	0.81	0.89	0.92	0.95	0.88	0.93	0.86	0.92	0.67	0.77	0.57	0.73	0.58	0.71

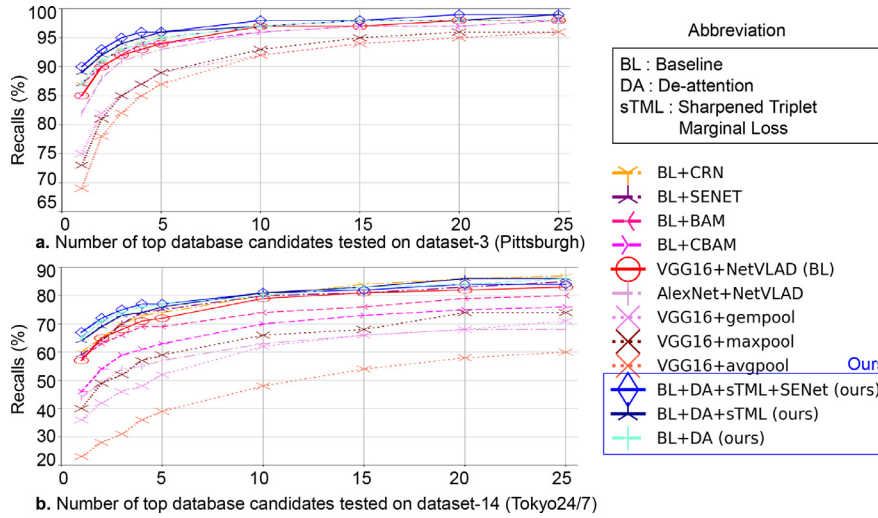


Fig. 9. Recall curve. This graph compares the recall results of the existing VPR model and our model, where all values in the graph are from our own experimental results. All models are trained with the Pittsburgh train dataset-1 and tested with Pittsburgh dataset-3 and Tokyo 24/7 dataset-14. The baseline (red solid line) with the existing channel and spatial attention methods such as SENet, BAM, CBAM, or CRN improve recall by 2 ~ 3% points, while our method achieves an improvement of about twice that for each dataset (blue solid line).

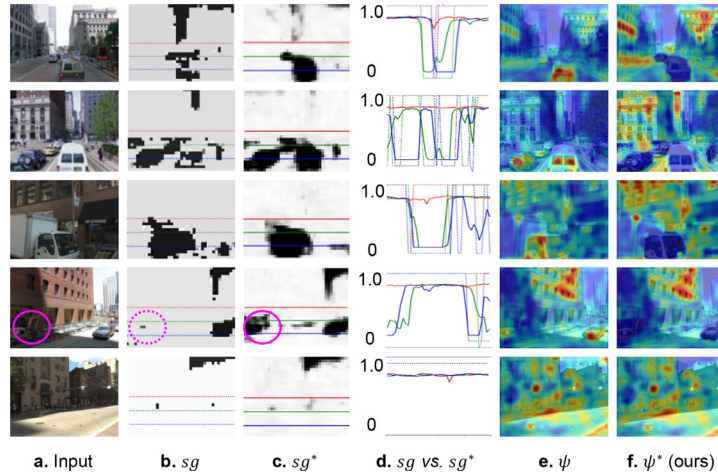


Fig. 10. Visualization of De-attention Result. **b** is semantic guidance for image **a**. **c** is a de-attention weight which is a prediction of the semantic guidance. **d** shows detailed comparisons of **b** and **c** at three horizontal probing lines, where solid lines are for the ground truth, and dotted lines are for predictions. **c** can detect non-static objects not detected in the ground truth (excavators at purple circle). **e** and **f** are local features before and after de-attention, respectively. Landmark features such as buildings and roads are more focused on after de-attention in **f**.

$\angle NQP$ does not change significantly (less than 5°) in any datasets or margins in the experiments. Since this new loss tends to separate the two distance distributions, d_{qp} and d_{pp} , more sharply, as shown in Fig. 11f, we call it sharpened triplet marginal loss (sTML).

4.4. Network size and run-time

Network size is measured by counting the number of tensor parameters. We use an Intel Xeon Silver 4210R CPU (2.40GHz) with one Nvidia RTX3090 GPU to measure the execution time. Table 4

Table 4
Model size and run-time tested for Tokyo24/7 test data.

Models	Size (# of parameters)	Recall@1	Sec./image
Baseline (BL)	14.7M	0.57	0.04
BL+SENet	14.8M	0.59	0.04
BL+CRN	15.8M	0.60	0.05
BL+Deattention (ours)	15.8M	0.65	0.05

is the result of measuring the size and run time of the critical model. The image size used for measurement is $3 \times 480 \times 640$ for the channel, height, and width, respectively.

For baseline (BL) composed of VGG16 CNN layer and NetVLAD pooling, the parameter size is 14.7M. With our de-attention, the recall performance improves by about 8% points, and its size and execution time increase slightly. VGG16 with fully connected layers is included in the table for size comparison with general networks. Regarding the run time, even if it is about 0.05 seconds per image, a real-time operation is not guaranteed when the number of database images to be processed increases. However, when the VPR algorithm is fixed with sufficient performance, real-time operation is possible by building the database descriptor offline.

4.5. Qualitative evaluation

We present an example of the retrieval results for the challenging query of dataset-14 as shown in Fig. 12, using the qualitative evaluation expression method used in [14]. Note that our

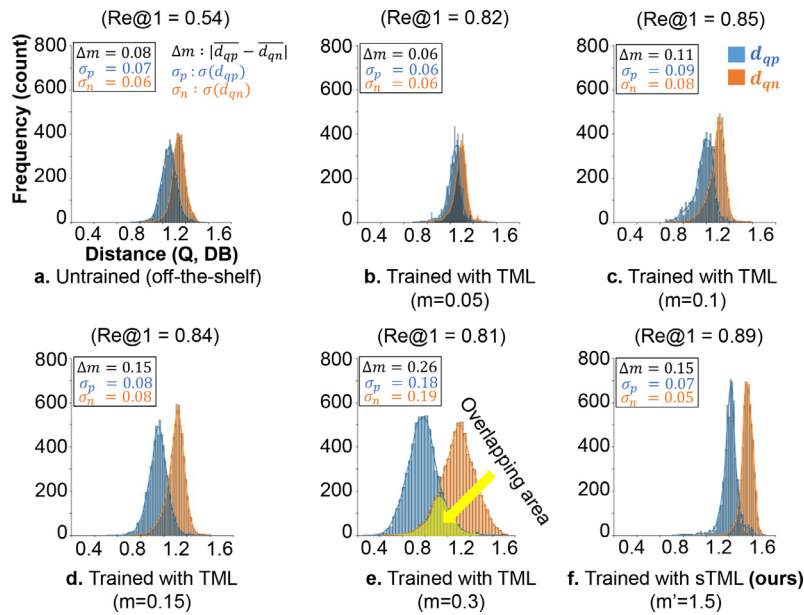


Fig. 11. Histogram of descriptor distances. This is a histogram of the positive and negative group's mean and standard deviation to visualize the distinction between descriptors. The x-axis is the distance bins, and the y-axis is the frequency of each distance. d_{qp} and d_{qn} are the Euclidean distances between query and positive, query and negative, respectively. Δm is an absolute difference between the d_{qp} mean and d_{qn} mean. σ_p and σ_n are the standard deviations of d_{qp} and d_{qn} , respectively.

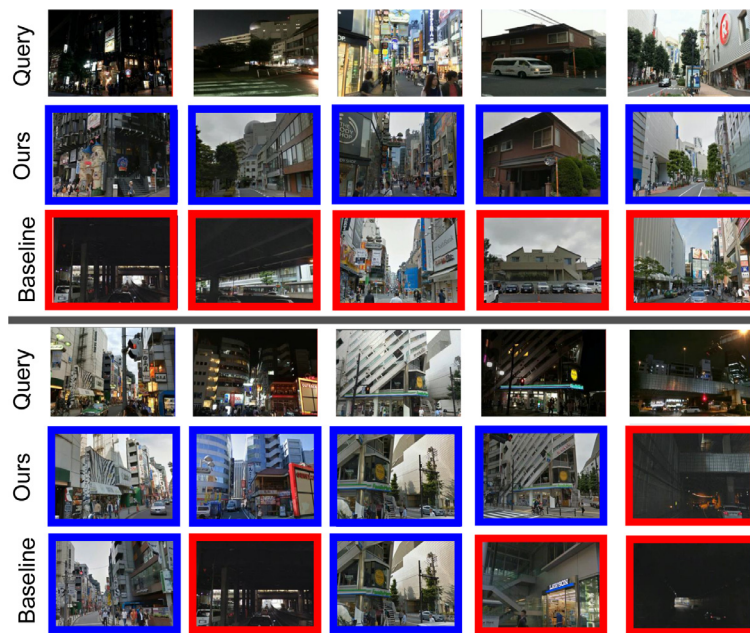


Fig. 12. Retrieval Results for challenging queries Examples of image retrieval results for challenging queries on dataset-14, Tokyo24/7. Each column corresponds to one test case: the query is shown in the first row, the top retrieved image using our best method (BL + DA + sTML + SENet) in the second, and the top-1 retrieved image using the best baseline (VGG16 + NetVLAD: BL) in the last row. The blue and red borders correspond to correct and incorrect retrievals, respectively.

de-attentive descriptor can recognize the same place despite large changes in appearance due to illumination (day/night), viewpoint, and partial occlusion by cars, trees, and people.

In particular, in the case of the fourth column query at the top in Fig. 12, the BL retrieved the parking lot while we ignored the large van and found a house correctly. Also, in the case of the third and fourth columns at the bottom, both BL and ours succeed in the query taken during the daytime, but only ours succeeds in the query taken at night. In the case of the fifth query, which is too dark and lacks landmarks, both failed, but we found

relatively similar structures. Next, Fig. 13 is the retrieval results for various attention and de-attention method with the internal feature visualization. The vertical direction is divided into Baseline (BL), BL+CBAM, BL+SENet, BL+CRN, and ours (BL + de-attention + sTML + SENet) methods, respectively. Then, for each method, the top row is the query and retrieved top-1 database images, and the bottom row is a visualization of their local features. Our method at the bottom reduces the features of dynamic objects such as people and vehicles more than other methods and relatively strengthens landmarks such as buildings.

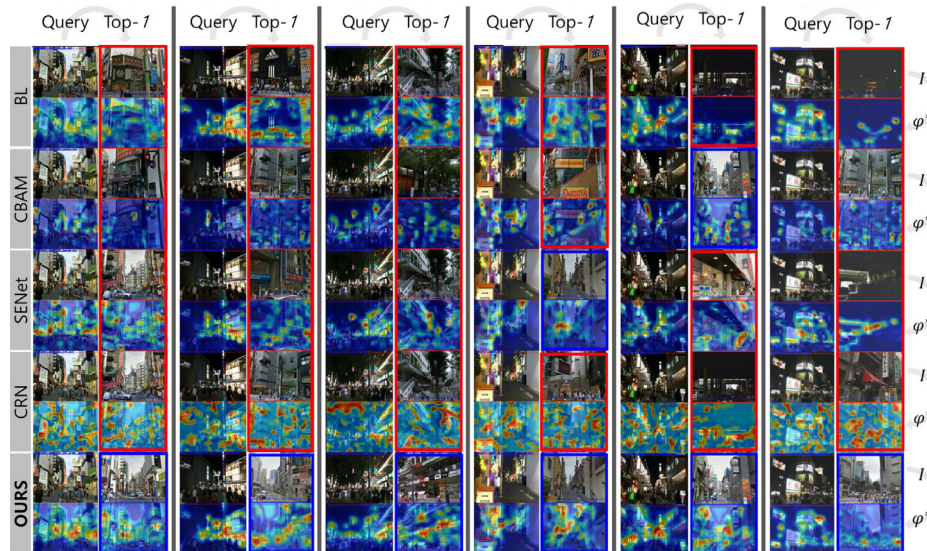


Fig. 13. Retrieval results and visualization for various cases Examples of image retrieval results for challenging queries on dataset-14, Tokyo24/7. Each column corresponds to one test case. From the top to the bottom row, each row corresponds to Baseline (BL), BL+CBAM, BL+SENet, BL+CRN, and ours (BL + de-attention + sTML + SENet), respectively. For each row, the top sub-row is the query and top-1 database image, and the bottom sub-row is the local feature heatmap. The blue and red borders correspond to correct and incorrect retrievals, respectively.

5. Discussion

5.1. Why not employ semantic guidance directly to the input image?

To answer these questions, we performed the following two approaches. First, we added a channel for semantic guidance to the input 3 (RGB) channel. Then, we added a convolutional layer to the front of the feature extractor that receives four channels and outputs three channels to use fixed-size pre-trained weights. Since a new convolutional layer has been added to the front, we must train the model entirely. In our preliminary experiments, if only the added convolutional layers and the last 3 layer of VGG16 were trained, the recall top-1 was 0.83, lower than the baseline's 0.85. Next, the recall performance is similar to the baseline when the entire layer is re-trained with the added extra channel. According to these experiments, delivering semantic guidance to the input channel directly does not improve the performance of the baseline.

The second method is to overwrite the input RGB channel with semantic guidance. To this end, we overwrite the semantic guidance mask over the pixel values in existing input data. The pixel values of the clutter region were replaced with constant numbers (zeros, ones, and random values), or Gaussian blurrings [29]. In our preliminary experiments, all cases of the second method are lower than the baseline without a train. Even with training, the highest recall is 0.86, so the performance improvement over the proposed method is negligible. In addition, it is very disadvantageous in terms of hardware resources that an extra extensive segmentation network for generating semantic guidance is required for all input images, even at test time. To make a long story short, passing semantic guidance directly to the input does not improve performance more than ours.

5.2. Which object should be de-attention?

We have investigated how performance changes when an object is de-attention. A crowded and non-static object-rich Tokyo24/7 (dataset-3) is used for the experiment. Since it has no training data, a network is trained with Pittsburgh (dataset-1). In our preliminary experiments, de-attention (human, vehicle) has the best recall at top-1. All de-attention except vegetation only has bet-

ter recall than baseline. De-attention human, vehicle, vegetation (green solid line) suits applications requiring high season changes or high performance at high recall at If only the recall at top-1 is required without re-ranking [10], de-attention human, vehicle should be selected. We set the human, vehicle combination as the default clutter for other de-attention experiments.

5.3. Limitations of our approaches

With the proposed de-attention method, the VPR network can learn to adjust the weights of local features by class units. However, when the user sets the weight of a hyper-parameter class, it is not guaranteed to be optimal. We alleviated this problem with preliminary experiments in Section 5.2, but consideration for setting may be required depending on the dataset. Next, we use the pre-trained DeepLabv3 segmentation network to generate semantic guidance during training. If the environment of the pre-trained dataset severely differs from the VPR environment, performance can be reduced. For example, if we apply the semantic guidance module based on the Cityscape dataset [19] to the indoor dataset, the result of object segmentation will be inaccurate. We have proposed a clutter augmentation method that extracts an image patch of a non-static object area designated from an input image and copies it to an arbitrary location. This method can produce unnatural images if cluttered objects are inaccurately detected, or the destination location is inappropriate. But our method seems sufficient for generating augmented datasets for performance evaluation. Our proposed de-attention and sTML can be easily added to existing visual place recognition in a plug-and-play type. In this paper, we compared the performance by applying several existing attention and our method to the VGG16-based NetVLAD method. We plan to add our method to the latest state-of-the-art VPR for future works.

6. Conclusion and future works

We have addressed the feature de-attention and ranking loss problems for visual place recognition in a complex urban environment with dynamic non-static clutter objects. We found that the conventional attention mechanisms could not effectively sup-

press the local features of dynamic objects for the visual place recognition tasks. To cope with this problem, we have devised the de-attention mechanism that allows the user to select the types of non-static clutter objects to be ignored. In addition, we have proposed the sharpened triplet marginal loss that forces the two distributions to be more sharply separated by analyzing the distance distributions of positive and negative samples. For evaluation, we have also proposed a new clutter augmentation method to create a crowded version of the public dataset. Because the proposed model significantly improves the state-of-the-art VPR tasks on public benchmark datasets highly crowded with people and cars, it can be useful in city-scale localization. Since the feature influence of the object unit is controllable during the learning process, our method can be used in various applications that interfere with unwanted clutter. As a limitation, we determined which objects would be suppressed by a simple prior experiment, but it may not be optimal. To tackle this, we plan to design a new architecture in which the optimal de-attention weights for each class are automatically learned in future work.

Our contributions are summarized as follows. We have described the details of our proposed de-attention model and sTML function in Section 3. And we have also visited not only the quantitative performance of our model (Fig. 9, Table 3) but also its qualitative performance (Fig. 12) and internal results (Fig. 10) in Section 4.3. In addition, we have conducted a lot of prior experiments to answer questions that readers may have about our de-attention model in Section 5. Lastly, we have shown the advantage of de-attention by visualizing the results of the conventional attention and de-attention methods with heatmaps (Fig. 13).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2019-0-01309, Development of AI Technology for Guidance of a Mobile Robot to its Goal with Uncertain Maps in Indoor/Outdoor Environments)

References

- [1] K. Lee, S. Lee, W.J. Jung, K.T. Kim, Fast and accurate visual place recognition using street-view images, *ETRI Journal* 39 (1) (2017) 97–107.
- [2] X. Zhang, L. Wang, Y. Su, Visual place recognition: A survey from deep learning perspective, *Pattern Recognition* 113 (2021) 107760.
- [3] H.J. Kim, E. Dunn, J.-M. Frahm, Learned contextual feature reweighting for image geo-localization, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 3251–3260.
- [4] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.
- [5] J. Park, S. Woo, J.-Y. Lee, I.S. Kweon, Bam: Bottleneck attention module, in: Proceedings of the British Machine Vision Conference (BMVC), 2018, pp. 1–14.
- [6] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.
- [7] J. Park, S. Woo, J.-Y. Lee, I.S. Kweon, A simple and light-weight attention module for convolutional neural networks, *International journal of computer vision* 128 (4) (2020) 783–798.
- [8] J. Xu, C. Shi, C. Qi, C. Wang, B. Xiao, Unsupervised part-based weighting aggregation of deep convolutional features for image retrieval, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018. <https://ojs.aaai.org/index.php/AAAI/article/view/12231>.
- [9] H. Noh, A. Araujo, J. Sim, T. Weyand, B. Han, Large-scale image retrieval with attentive deep local features, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 3456–3465.
- [10] B. Cao, A. Araujo, J. Sim, Unifying deep local and global features for image search, in: European Conference on Computer Vision, Springer, 2020, pp. 726–743.
- [11] H. Liu, R. Wang, S. Shan, X. Chen, Deep supervised hashing for fast image retrieval, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2064–2072.
- [12] W. Chen, X. Chen, J. Zhang, K. Huang, Beyond triplet loss: a deep quadruplet network for person re-identification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 403–412.
- [13] B. Chen, W. Deng, Deep embedding learning with adaptive large margin n-pair loss for image retrieval and clustering, *Pattern Recognition* 93 (2019) 353–364.
- [14] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, Netvlad: Cnn architecture for weakly supervised place recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5297–5307.
- [15] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
- [16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, *International journal of computer vision* 115 (3) (2015) 211–252.
- [17] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: *Icml*, 2010, pp. 807–814.
- [18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 801–818.
- [19] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3213–3223.
- [20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, *Advances in neural information processing systems* 32 (2019) 8026–8037.
- [21] Nanne, pytorch-netvlad, 2018, (<https://github.com/Nanne/pytorch-NetVlad>). [Online; accessed 4-July-2022].
- [22] J.A. Hartigan, M.A. Wong, Algorithm as 136: A k-means clustering algorithm, *Journal of the royal statistical society, series c (applied statistics)* 28 (1) (1979) 100–108.
- [23] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with gpus, *IEEE Transactions on Big Data* (2019).
- [24] S. Suzuki, et al., Topological structural analysis of digitized binary images by border following, *Computer vision, graphics, and image processing* 30 (1) (1985) 32–46.
- [25] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, T. Pajdla, 24/7 place recognition by view synthesis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1808–1817.
- [26] F. Radenović, G. Toliás, O. Chum, Fine-tuning cnn image retrieval with no human annotation, *IEEE transactions on pattern analysis and machine intelligence* 41 (7) (2018) 1655–1668.
- [27] O. Chum, J. Philbin, J. Sivic, M. Isard, A. Zisserman, Total recall: Automatic query expansion with a generative feature model for object retrieval, in: 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–8.
- [28] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.
- [29] R.A. Hummel, B. Kimia, S.W. Zucker, Deblurring gaussian blur, *Computer Vision, Graphics, and Image Processing* 38 (1) (1987) 66–80.

Seung-Min Choi received his B.S. degree in electronics engineering from Chung-Ang University, Seoul, Rep. of Korea, in 2002, and M.S. degree in electrical and computer engineering from Seoul National University, Rep. of Korea, in 2004, and Ph.D. degree in the division of future vehicle from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 2023, respectively. Since 2004, he has been a principal research scientist with the Artificial Intelligence Research Laboratory, Electronics and Telecommunication Research Institute (ETRI), Daejeon, Republic of Korea. His research interests include artificial intelligence-based visual place recognition, image retrieval, object segmentation, and depth estimation for robot and vehicle applications.

Seung-Ik Lee received his B.S., M.S., and Ph.D. in computer science from Yonsei University, Seoul, Rep. of Korea, in 1995, 1997, and 2001, respectively. He works for the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea. Since 2005, he has been with the Department of Artificial Intelligence, University of Science and Technology, Daejeon, Rep. of Korea, where he is now a professor. His research interests include computer vision, anomaly detection, object detection, deep learning, and reinforcement learning.

Jae-Young Lee received his B.S. in mathematics and M.S., Ph.D. in computer science from Seoul National University, Rep. of Korea, in 1996, 1998, and 2005, respectively. He works for the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea. Since 2007, he has been with the Department of Artificial Intelligence, University of Science and Technology, Daejeon, Rep. of Korea,

where he is now a professor. His research interests include computer vision, visual object tracking, camera calibration, and robot navigation.

In So Kweon received the B.S. and M.S. degrees in mechanical design and production engineering from Seoul National University, South Korea, in 1981 and 1983, respectively, and the Ph.D. degree in robotics from the Robotics Institute, Carnegie

Mellon University, USA, in 1990. He was with the Toshiba R&D Center, Japan, and he is currently a KEPCO chair professor with the Department of Electrical Engineering, KAIST, Korea, since 1992. He served as a program co-chair for ACCV 07' and ICCV 19, and a general chair for ACCV 12. He is on the honorary board of IJCV. He was a member of "Team KAIST," which won the first place in DARPA Robotics Challenge Finals 2015. He is a member of the IEEE and the KROS.