Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.058233

ARTICLE

DIEONet: Domain-Invariant Information Extraction and Optimization Network for Visual Place Recognition

Shaoqi Hou[1,2,3,*], **Zebang Qin**[2], **Chenyu Wu**[2], **Guangqiang Yin**[2], **Xinzhong Wang**[1] and **Zhiguo Wang**[2,*]

[1]School of Computer Science and Technology, Xinjiang University, Urumqi, 830046, China
[2]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China
[3]Institute of Public Security, Kash Institute of Electronics and Information Industry, Kashi, 844000, China

*Corresponding Authors: Shaoqi Hou. Email: shaoqi.hou@foxmail.com; Zhiguo Wang. Email: zgwang@uestc.edu.cn

Received: 07 September 2024; Accepted: 11 November 2024; Published: 06 March 2025

ABSTRACT: Visual Place Recognition (VPR) technology aims to use visual information to judge the location of agents, which plays an irreplaceable role in tasks such as loop closure detection and relocation. It is well known that previous VPR algorithms emphasize the extraction and integration of general image features, while ignoring the mining of salient features that play a key role in the discrimination of VPR tasks. To this end, this paper proposes a Domain-invariant Information Extraction and Optimization Network (DIEONet) for VPR. The core of the algorithm is a newly designed Domain-invariant Information Mining Module (DIMM) and a Multi-sample Joint Triplet Loss (MJT Loss). Specifically, DIMM incorporates the interdependence between different spatial regions of the feature map in the cascaded convolutional unit group, which enhances the model's attention to the domain-invariant static object class. MJT Loss introduces the "joint processing of multiple samples" mechanism into the original triplet loss, and adds a new distance constraint term for "positive and negative" samples, so that the model can avoid falling into local optimum during training. We demonstrate the effectiveness of our algorithm by conducting extensive experiments on several authoritative benchmarks. In particular, the proposed method achieves the best performance on the TokyoTM dataset with a Recall@1 metric of 92.89%.

KEYWORDS: Visual place recognition; domain-invariant information mining module; multi-sample joint triplet loss

1 Introduction

Visual Place Recognition (VPR) is an indispensable key technology for smart city construction and national defense security construction. As a downstream task in the field of image retrieval, it plays a key role in loopback detection for robots and navigation and localisation systems for unmanned vehicles [1]. The basic flow of a VPR system is as follows: given a query image, retrieve the most similar image from the existing geo-location reference image database, and use the retrieved image as the predicted geo-location of the query image. However, affected by different viewpoints and different environments (such as lighting changes, seasonal changes, etc.), images at the same location present large style differences, which brings great challenges to the VPR task.

VPR tasks have been studied since the era of handcrafted features. In order to solve the above challenges, Mei et al. [2] used fast detection technology to extract key corners in the image, and then used SIFT (Scale Invariant Feature Transform) descriptors to characterize the features. Churchill et al. [3] achieved accurate

localization by querying repetitive structures in similar images and weighting related bag-of-words phrases. These models essentially use local feature matching, so they are resistant to viewpoint changes, but less robust in the appearance changes. In order to solve the challenges caused by appearance changes, many researchers consider the global image and use global feature descriptors. NetVLAD [4] uses the soft assignment VLAD (Vector of Locally Aggregated Descriptors) algorithm to aggregate the local features extracted by the neural network, and then reduces the dimension of the aggregated features to obtain the global features. NetVLAD has shown good robustness against environmental changes, and its success has established the classic VPR architecture–"backbone network-feature aggregation layer".

At present, one of the core pursuits of VPR model design is how to obtain robust descriptors that adapt to complex environmental changes. On this requirement, CNNs (Convolutional Neural Networks) and ViTs (Vision Transformers) show their respective advantages. CNN can learn different granularity information from different spatial levels when processing images. The self-attention operation in ViTs can aggregate global context information. For example, in recent representative work, Berton et al. [5] built a standard baseline for VPR using several different types of CNN backbones, which allows them to directly determine the impact of CNN architectures with different feature granularities on VPR accuracy by comparing multiple sets of experiments. Unlike Berton's research, Oquab et al. [6] demonstrated the effectiveness of utilising only ViT as a feature extractor, which can learn global dependencies between different patch tokens from an arbitrary collection of images. Based on this, AnyLoc [7] designed a generic VPR processing flow. However, the above two mainstream architectures inevitably have a problem, in that they emphasize the extraction and integration of general features, and the extraction ability of salient features, which plays a key discriminative role in the VPR tasks, is seriously insufficient.

Objects such as buildings can be used as salient feature information of images. In VPR images, building related classes have the advantage of maintaining detailed edge and contour information across seasons, ages, and even day-night lighting changes. Based on this, we design a Domain-invariant Information Mining Module (DIMM) to enhance the feature extraction ability of existing networks for static objects. The design of DIMM is inspired by the idea of dynamic convolution, and consists of a learnable attention map and a weight generation module. The attention map captures the semantic relationship between different spatial regions of the feature, and is constantly updated during the learning process. The weight generation module is a stacked structure, which can adaptively generate feature weights according to the input. In particular, the weights generated by DIMM are fed to the input feature maps in the form of channel by pixel, in order to comprehensively enhance the expression of building feature information from both channel and spatial dimensions.

In addition, the current research on VPR algorithm mainly focuses on representation learning, and few researchers consider more effective metric learning methods from the training level. As an important part of the training strategy, the loss function plays a role in constraining model parameters and promoting model convergence. As with upstream retrieval tasks, Triplet Loss has been widely used in VPR to aggregate similar samples and separate unrelated samples in the feature space, but its disadvantages are also very obvious: Since Triplet Loss only sees a negative sample, the problem of overfitting caused by inappropriate selection of negative samples makes the model have optimization errors. In addition, $d_{q,n}$ (query-negative sample distance) may decrease as $d_{q,p}$ (query-positive sample distance) decreases, causing the distance between the query and negative samples to collapse, leading to the degradation of the model discrimination ability.

To address the above problems, we designed a new Loss function named MJT Loss (Multi-sample Joint Triplet Loss). On the one hand, MJT Loss introduces multiple negative samples on the basis of the triplet loss to expand the sample space in the minimum training unit and increase the stability of training. On the other hand, we also add a distance constraint term for positive and negative samples to push the distance

between positive samples and multiple selected negative samples in the feature space synchronously, so that the descriptors extracted by the model have higher discrimination, thus preventing the degradation of the feature discrimination ability of the model.

The main contributions and innovations of this work are as follows:

(1) A plug-and-play Domain-invariant Information Mining Module (DIMM) is proposed, which uses a cascaded group of convolutional units to implicitly separate salient information from non-salient information in features. The model pays more attention to static objects, and effectively improves the robustness and generalization ability of the VPR model.

(2) A new metric learning method, Multi-sample Joint Triplet Loss (MJT Loss), is designed, which modifies the "single-single" sample processing pattern of the distance constraint term to the "single-many" sample processing pattern. Furthermore, the distance constraint is added to make the optimization path of the model move towards the global optimal direction.

(3) Complete ablation experiments and algorithm comparison experiments are carried out to prove the effectiveness of the proposed algorithm.

2 Related Work

Visual Place Recognition. Early VPR algorithms [1] were based on handcrafted features for retrieval, such as Bag-of-Words [8], Fisher vector [9], and VLAD [10]. With the rapid development of deep learning, the current mainstream approach is to use CNN or ViT framework as feature extractor to extract local features. The NetVLAD [4] algorithm proposed in 2017 is an important transformation from traditional manual features to deep features. The recent MixVPR [11] used ResNet as the backbone network and designed the Feature-Mixer Feature aggregation module, which is at the advanced level in terms of effect and processing speed on multiple datasets. In particular, the emergence of TransVPR [12] provided a new idea for the development of VPR. TransVPR combined CNN and self-attention mechanism to extract global features, and fused tokens generated by Transformer modules of different layers as local features to complete the second stage of retrieval. Although the TransVPR model achieves high accuracy, its computation and complexity far exceed previous work.

Dynamic Convolution. Different from the traditional convolution operation, the main idea of dynamic convolution is to let the network adaptively adjust the weight of the convolution kernel according to the characteristics of the input data. Some researchers adjust the parameters by adaptively adjusting the size of the convolution kernel. Deformable kernels [13] sampled the weights in the space of the convolution kernel to adapt to the effective reception field (ERF) while keeping the receptive field unchanged. Of course, the size of the convolution kernel can be changed during the convolution operation, and the parameters of the convolution kernel can also be dynamically adapted. Dynamic convolutional neural network DY-CNN [14] started from the weight direction of convolution kernels, and performed soft attention on multiple convolution kernels to adaptively generate weight parameters. Our DIMM relies on content-aware dynamic convolution, which focuses on the landmark information of a specific location, and shows good results in visual scene recognition tasks.

Loss Function. Loss function is the most important part of the training strategy, which updates and optimizes parameters through backpropagation [15]. Since the triple loss function was proposed in 2015 [16], it has become the most widely used loss function in retrieval tasks. In order to better represent the semantics, Chen et al. proposed quadruplet loss [17], which introduces an additional negative sample to build another triple with different queries. CosPlace [18] is a work that used visual scene recognition as a downstream classification task, which introduces Large Margin Cosine (LMC) loss function [19] into visual

scene recognition and shows good performance. However, classification tasks require high data sets, and fine-grained recognition cannot be achieved in large visual scene recognition tasks. Based on the above analysis, designing a better form of triple loss function is still an effective way to achieve fast and stable convergence of the model.

3 Methodology

3.1 Overall Framework

As shown in Fig. 1, DIEONet takes NetVLAD [4] as the baseline and consists of three parts: the backbone network VGG-DIMM, the NetVLAD aggregation head, and the Loss function MJT Loss. Among them, VGG-DIMM and MJT Loss are the proposed new schemes, which perform the function of feature mining and optimizing model parameters in the whole architecture, respectively. Specifically, the process of DIEONet is as follows:

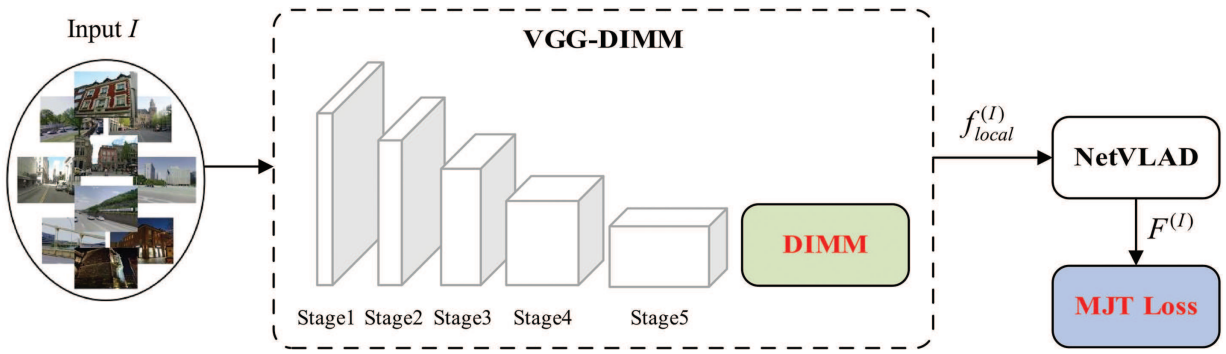

Figure 1: Overall architecture of DIEONet

Step 1: For a given image I, we utilize the backbone network VGG-DIMM to initially extract its local feature descriptor $f_{local}^{(I)} \in R^{H \times W \times C}$, which is expressed as follows:

$$f_{local}^{(I)} = \mathrm{DIMM}\left(\mathrm{VGG}\left(I\right)\right) \tag{1}$$

Step 2: Take the local feature $f_{local}^{(I)}$ extracted by VGG-DIMM as input, and use the vanilla NetVLAD aggregation head [4] to aggregate $f_{local}^{(I)}$ into a one-dimensional global feature descriptor $F^{(I)} \in R^{D}$, where $D = cluster\_num \times C$ and cluster_num is the number of aggregated clusters in NetVLAD, C is the number of channels of $f_{local}^{(I)}$. The expression of $f_{local}^{(I)}$ is as follows:

$$F^{(I)} = \mathrm{NetVLAD}\left(f_{local}^{(I)}\right) \tag{2}$$

Step 3: Firstly, a certain number ratio of query, positive sample and negative sample images are all operated by Step 1–Step 2 to obtain different global feature descriptors. Then, these descriptors are measured using the designed MJT Loss, and the parameters of the whole model are updated through the backpropagation mechanism.

3.2 VGG-DIMM

VGG-DIMM is composed of VGG16 (Visual Geometry Group 16) and DIMM in series. The original intention of our design is that VGG16 extracts the basic features of the input images, and then DIMM completes the mining of the salient information in the basic features, as shown in Fig. 2.

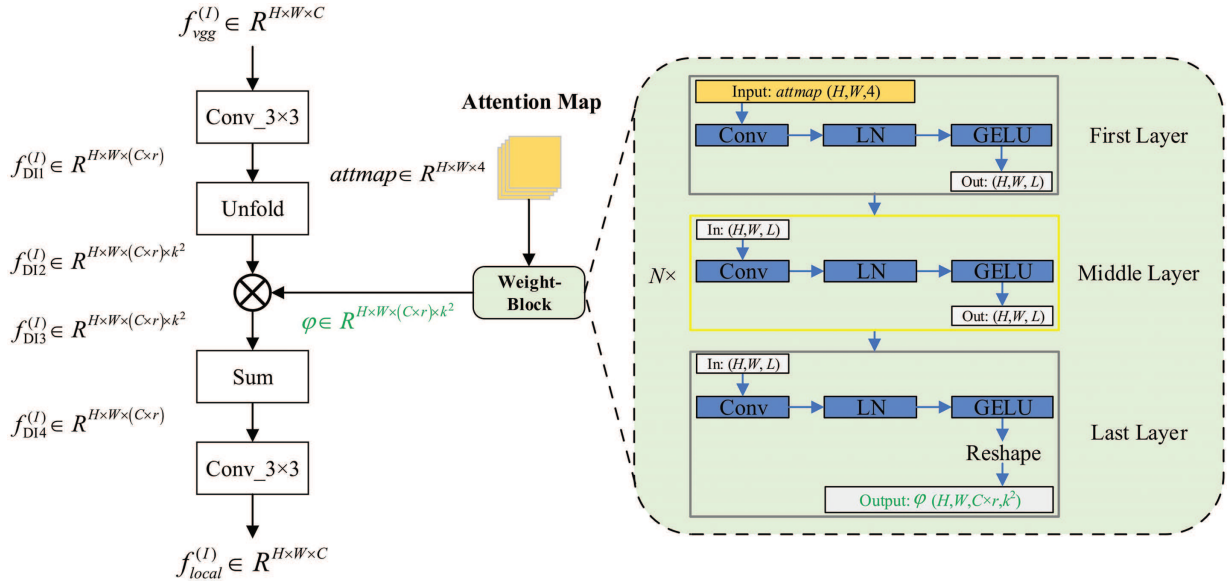

Figure 2: Structure diagram of DIMM

Specifically, as a plug-and-play feature enhancement module, DIMM is processed as follows:

Step 1: For the basic feature map $f_{vgg}^{(I)} \in R^{H\times W\times C}$ extracted by VGG16, DIMM first increases the capacity of feature channel information by a $k = 3 \times 3$ convolution operation with *stride* = 1 to obtain a new feature representation $f_{\mathrm{DI1}}^{(I)} \in R^{H\times W\times(C\times r)}$, where r is the feature expansion rate.

Step 2: In order to improve the capacity and richness of the spatial information of $f_{\mathrm{DI1}}^{(I)}$, a sliding window with $k = 3 \times 3$ and *stride* = 1 is designed to traverse the $f_{\mathrm{DI1}}^{(I)}$ feature map along the width and height directions. In particular, this process does not perform convolution operation, but merely copies the local receptive field region selected by the sliding window and rearranges it according to its original spatial structure to obtain $f_{\mathrm{DI2}}^{(I)} \in R^{H\times W\times(C\times r)\times k^2}$. That is:

$$\begin{cases} f_{\mathrm{DI2}}^{(I)} = \mathrm{Unfold}\left(f_{\mathrm{DI1}}^{(I)}\right) \\ f_{\mathrm{DI2}}^{(I)}\left(i,\ j,\ C\times r,\ \alpha\times k+\beta\right) = f_{\mathrm{DI1}}^{(I)}\left(i+\alpha,\ j+\beta,\ C\times r\right) \end{cases} \tag{3}$$

where i, j represent the pixel index of feature map $f_{\mathrm{DI1}}^{(I)}$ in high and wide dimensions respectively, i.e., $i = 1, 2, \ldots, H$, $j = 1, 2, \ldots, W$; $\alpha \times k + \beta$ is controlled by the size of the sliding window and is the parameter of the local receptive field expansion into columns, representing the index on the last dimension of $f_{\mathrm{DI2}}^{(I)}$, where $\alpha = 0, 1, \ldots, k-1$, $\beta = 1, 2, \ldots, k$.

Step 3: The attention mask φ output by the weight generation module, which has the same shape as $f_{\mathrm{DI2}}^{(I)}$, is fed back into the feature map $f_{\mathrm{DI2}}^{(I)}$ to mine the important domain-invariant information in $f_{\mathrm{DI2}}^{(I)}$. and obtain

the feature expression $f_{\mathrm{DI3}}^{(I)} \in R^{H\times W\times (C\times r)\times k^2}$ with strong robustness and strong generalization ability, that is:

$$f_{\mathrm{DI3}}^{(I)} = \varphi \otimes f_{\mathrm{DI2}}^{(I)} \tag{4}$$

where φ represents the weight descriptor output by the weight generation module, and the symbol $\otimes$ represents the multiplication of corresponding elements.

As shown in Fig. 2, the attention map (a tensor initialized with all ones of shape $H \times W \times 4$) is the input of the weight generation module, which is transformed into the attention mask φ after being processed by a cascaded group of convolutional units. φ can deeply obtain the interrelationship between different spatial regions in $f_{\mathrm{DI2}}^{(I)}$ through subsequent iterative training, and recalibe the saliency information that needs to be paid attention to. The attention mask φ can be defined as follows:

$$\begin{aligned}\varphi &= \mathrm{WeightBlock}\left(attmap\right)\\ &= \mathrm{GELU}\left(\mathrm{LN}\left(\mathrm{Conv}\left(attmap\right)\right)\right)_{\times(N+2)}\end{aligned} \tag{5}$$

For the weight generation module, we borrow the idea of the Transformer's stacked structure that each layer can capture the characteristics of the input data, and with the increase of layers, the model can learn more complex and advanced data representation. It should be noted that the structure of the middle layer (N convolutional units) of the weight generation module is exactly the same, except that the first layer and last layer need to match the channels and dimensions of the external feature map. Specifically, each convolutional unit of the weight generation module is composed of a convolutional layer ($k = 3 \times 3$ and *stride* = 1), layer normalization, and GELU (Gaussian Error Linear Units) activation function. It is well known that layer normalization performs better than batch normalization in training small batches of samples, and layer normalization is more flexible. The activation function GELU has zero centrality, which helps alleviate the vanishing gradient problem and can provide better gradient information.

In general, during the training process, the attention map is iteratively processed by the weight generation module, which can adaptively obtain the weights of different local features in $f_{\mathrm{DI2}}^{(I)}$ during the learning process, and can implicitly capture the semantic relationship between different spatial regions in $f_{\mathrm{DI2}}^{(I)}$, so as to mine the building information in the image.

Step 4: Step 1 and Step 2 respectively enrich the detailed information of the feature map from two dimensions of channel and space. In order to obtain the relevance of the spatial information of the feature maps, the last dimension k^2 of $f_{\mathrm{DI3}}^{(I)}$ is summed to fuse the different local receptive field information to obtain $f_{\mathrm{DI4}}^{(I)} \in R^{H\times W\times (C\times r)}$. In addition, in order to interact the channel information of the feature map, $f_{\mathrm{DI4}}^{(I)}$ further goes through the convolution operation with $k = 3 \times 3$ and *stride* = 1 to obtain the local feature descriptor $f_{local}^{(I)} \in R^{H\times W\times C}$ of the final output of VGG-DIMM.

3.3 MJT Loss

Triplet loss has been proved by many research works [11,12], and it is a reliable and effective loss function in image retrieval tasks including VPR. The core idea of Triplet loss is to try to learn a feature space in which samples from the same class are closer together and samples from different classes are farther apart:

$$L_T\left(q,\ p,\ n\right) = \max\left(0,\ d\left(q,\ p\right) - d\left(q,\ n\right) + m\right) \tag{6}$$

where m is the custom margin in the triplet loss function, which is parameter used to control the hard distance between samples. $d(q, p) = ||q\text{-}p||_2$ denotes the Euclidean distance between q and p; q, p, n denote the global feature descriptors of the queries, positive samples, and negative samples, respectively.

It is well known that the input form of Triplet loss is a triplet (q, p, n), that is, during the training process, for a given query sample, the dataloader will randomly select a negative sample and a positive sample corresponding to it, as shown in Fig. 3a. With the input of different triples one by one, the best optimization path for the model is to learn the convergence information from each triple in order, that is, to let the query sample accumulate the experience of moving away from the last negative sample and continue to move away from the next negative sample. However, when the loss function considers only one negative sample at a time, the model may overfit that particular negative sample, especially if the negative sample is not selected optimally. In addition, in VPR tasks, the number of positive samples is usually relatively small, because the geographical location of the positive samples is required to be the same as that represented by the query sample. The negative sample space is much larger, because any sample that is geographically different from the query sample can be used as a negative sample. Therefore, due to the above class imbalance phenomenon, it is almost impossible for the model to follow the globally optimal path to learn.

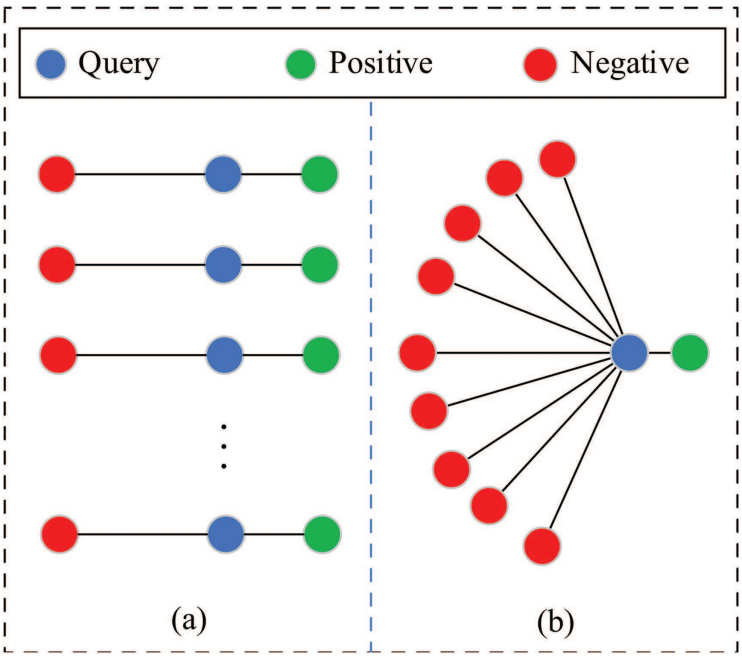

Figure 3: Schematic diagram of individual and joint processing

To solve the above problems, we design a "joint processing" mechanism for negative samples instead of the previous "separate processing" way, in order to avoid the model falling into local optimum during training. As shown in Fig. 3b, we introduce the consideration of multiple negative samples in the input triples. In particular, considering that the problem of high inter-class similarity is prominent in the VPR domain, we constrain the query to all the selected negative samples by maximizing the minimum distance between the classes:

$$L_U\left(q,\ p,\ \{n_i\}\right) = \max\left(0,\ d\left(q,\ p\right) - \min\left(\{d\left(q,\ n_i\right)\}\right) + m\right) \tag{7}$$

where $\{d(q, n_i)\}$ denotes the distance set formed by the query and the selected negative samples, and i denotes the index. Taking the query-negative sample with the minimum distance as the triple optimization goal, on the one hand, the discrimination of the model to all negative samples can be greatly increased, and the

generalization ability of the model can be improved. On the other hand, it can help the model to better learn the features of those less common categories, so as to improve the overall learning effect of the model.

For Eq. (8), we want to achieve the purpose of optimization by reducing $L_U(q, p, \{n_i\})$, that is, using $d(q, p)$-min$(\{d(q, n_i)\}) > m$ to bring the triplet $(q, p, \{n_i\})$ image closer or farther away in the feature space. However, this optimization method still has an obvious disadvantage: in the process of learning, as the distance between the query and the positive sample continues to get closer, the distance between the query and the negative samples may continue to get closer, which will cause the degradation of the feature recognition function of the model. Based on this consideration, we further add a new distance constraint to Eq. (7) to force the distance between positive samples and all negative samples in the feature space to be increased. Similarly, we achieve this by "focusing only on the closest negative sample to the positive sample". The final loss function is defined as follows:

$$L_{MJT} = \max\left(0,\ d\left(q,\ p\right) - \min\left(\left\{d\left(q,\ n_i\right)\right\}\right) + m - \min\left(\left\{d\left(p,\ n_j\right)\right\}\right) + m'\right) \tag{8}$$

where $\{d(q, n_i)\}$ and $\{d(q, n_j)\}$ are the query-negative sample distances sets and positive-negative sample distances sets; n_i and n_j are the feature descriptors of the i-th and j-th negative samples in the joint processing, respectively; m' is the hard distance parameter for positive-negative samples.

Furthermore, in order to express the working principle of the proposed loss function more clearly, we make a graphical description. As shown in Fig. 4, (a) is the feature space before training, where all samples are disordered; (b) is the expression of the loss function L_U in Eq. (8), which sees several more negative samples than L_T in a single learning. Under the constraint of this loss function, the positive samples are close to the query, all negative samples are far away from the query, and any query-negative sample distance is at least m larger than the query-positive sample distance. (c) is a schematic of the working mechanism of MJT Loss, which constrains the distance between positive samples and all negative samples to be at least m' to make all negative samples further away from the query, so as to further improve the discrimination of inter-class samples on the basis of (b).

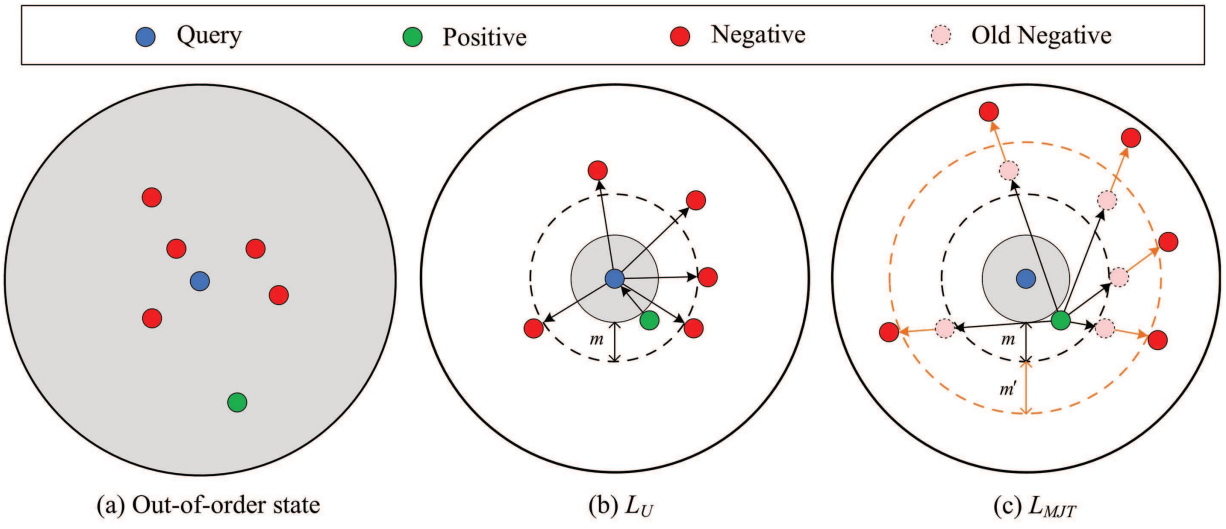

Figure 4: Graphical description of MJT Loss

4 Experiments

4.1 Implementation Details

Model details. For the backbone network VGG-DIMM, its VGG is the part before the last ReLU (Rectified Linear Unit) of the fifth layer of VGG16 obtained by cropping, which is pretrained on the ImageNet dataset. In DIMM, the feature expansion rate is set to $r = 4$, and the number of convolution units in the middle layer of the weight generation module and the convolution dimension of each convolution unit are respectively set to $N = 4$ and $L = 4$. As the aggregation layer, NetVLAD inherits the Settings of the baseline [4], where the number of clusters is 64 and the dimension of the output global feature vector is $64 \times 512 = 32{,}768$. In the optimization process of MJT Loss, we cache 1000 triple at a time, and each triple is used as a direct input, containing 1 query sample, 1 positive sample and 5 negative samples. In particular, we set the two distance parameters $m = 0.1$ and $m' = 1.65$ for MJT Loss.

Experimental setup. In order to ensure the consistency of the experimental results, the experiments in this chapter are carried out under the same software and hardware environment. The computing platform is 20.04.6LTS Linux system, using three Tesla T4 graphics cards (16 GB of video memory), Pytorch version 1.10.1, CUDA (Compute Unified Device Architecture) version 11.4, and Python version 3.6.13. In the training process, the training set is uniformly scaled to 640×480, and the Batch was set to 64. For the training strategy, we use the SGD (Stochastic Gradient Descent) optimizer to update the model parameters, with momentum set to 0.9 and weight decay set to 0.001. In addition, the learning rate adopted a multi-step learning strategy, the initial learning rate was set to 0.001, and the learning rate was reduced to 0.5 times of the original when the iteration reached the 5-th Epoch, and the training was stopped after 10 Epochs.

4.2 Datasets and Evaluation Metrics

Datasets. We use Pittsburgh30k [20] as the training set and evaluate our proposed method on four public benchmarks including Pittsburgh30k-test (hereafter Pitts30k), Pittsburgh250k-test (hereafter Pitts250k), Aachen Day-Night (hereafter Aachen) [21], and TokyoTM [4]. As shown in Table 1, these benchmarks take into account complex environmental variations such as season, weather, lighting, viewpoint, and so on.

Table 1: Introduction to the datasets

Datasets	Query	Database	Scene	Day-Night chage
Pittsburgh30k-test	6816	10,000	City + Suburb	×
Pittsburgh250k-test	8280	83,952	City + Suburb	×
Aachen Day-Night	191	6697	City	√
TokyoTM	7186	49,056	City	×

Evaluation metrics. In all experiments, we use Recall@1 (hereafter R@1) as the evaluation metric, which indicates the fraction of query images that are correctly located the first time. Specifically, the geographical location represented by the query image is taken as the center of the circle, and the reference image within the radius of 25 m is considered to be in the same position as the query image. Therefore, if at least one of the previous predicted images is within the threshold of the true geolocation of the query image, this query is considered to be correctly retrieved. In this paper, in order to be consistent with other advanced algorithms, the threshold value of Aachen dataset is set to 0.2 m, and the default value of other datasets is 25 m.

4.3 Ablations

(1) Module Validation

As shown in Table 2, we successively add DIMM and MJT Loss on baseline (BL) NetVLAD to verify their effectiveness, respectively. Among them, the Loss functions used by "BL" and "BL + DIMM" are Triplet Loss by default, and "BL + DIMM + MJT loss" is our overall model DIEONet.

Table 2: Ablation experiments for the modules

Models	Pitts30k	Pitts250k	Aachen	TokyoTM
	R@1 (%)	R@1 (%)	R@1 (%)	R@1 (%)
NetVLAD (BL)	84.48	84.40	45.55	90.48
BL + DIMM	84.54	84.58	56.02	90.95
BL + MJT Loss	85.50	86.76	52.88	92.76
BL + DIMM + MJT Loss (Our DIEONet)	85.72	87.15	62.30	92.89

Table 2 shows that our designed DIMM and MJT Loss improve the performance of the model to varying degrees on the four test benchmarks. Although the performance of DIMM is weak on Pittsburgh and TokyoTM, the performance of DIMM is significantly improved on Aachen, which is 10.47% higher than the baseline, which fully verizes the ability of DIMM to extract domain invariant information. For MJT Loss, it shows a great improvement over the baseline on all datasets. Specifically, the introduction of MJT Loss, which respectively improves 1.02% and 2.36% on Pitts30k and Pitts250k, and 7.33% and 2.28% on Aachen and TokyoTM, emphasizes the necessity of global optimization. In addition, we found through the last set of experiments that the improvement of the model by these two schemes is not conflicting, and both show a positive accumulative effect. At this point, our DIEONet even achieves 62.30% on Aachen, which is 16.75% higher than the baseline. This phenomenon shows that the two schemes generate a good coupling mechanism inside the model, and especially show strong robustness against the challenge of day-night variation.

Further, in order to demonstrate the effectiveness of the proposed model more intuitively, we show several attentional heatmaps of NetVLAD (BL) and DIEONet, which are randomly selected from the chosen dataset. As shown in Fig. 5, compared with NetVLAD (BL), our DIEONet pays more attention to strongly discriminative static-like objects and regions in the scene images, which fully reflects that DIEONet can effectively construct global associations of environmentally invariant location features.

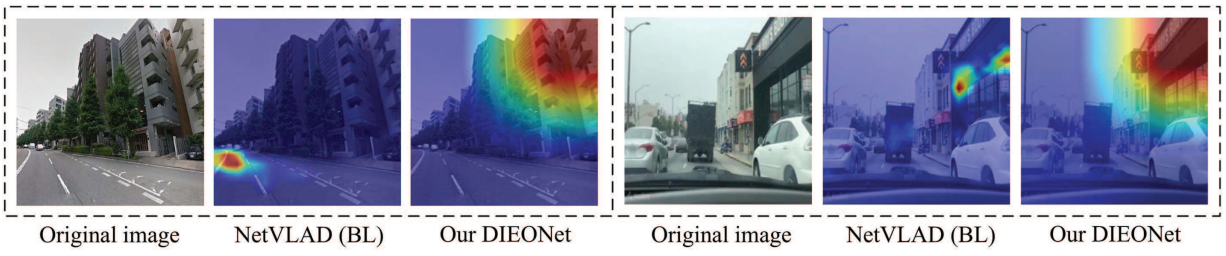

Figure 5: Attention heatmaps of NetVLAD (BL) and our DIEONet

(2) Hyperparameter Validation

In Section 4.1, we mentioned four hyperparameters, which are the number of convolution units N in the middle layer of the weight generation module in DIMM, the convolution dimension L of each convolution unit, the embedding stage position $stage_n$ of DIMM in VGG16, and m' in the Loss function MJT Loss. We follow the control variables approach and verify each hyperparameter one by one based on the best model, by keeping the best default setting for each of the remaining parameters when verifying one hyperparameter. To compare the comprehensive performance of the proposed model on all datasets, we averaged the results on these datasets.

As shown in Table 3, in DIMM, we similarly find that the variation of DIMM hyperparameters mainly affects the performance of the model on the Aachen dataset. Firstly, for the change of hyperparameter N, the recall rate of the model on the Aachen dataset first increases and then decreases, which is just the opposite of the performance on the Pitts30k and Pitts250k datasets. In this regard, we infer that DIMM has strong applicability for day-night scenes where the image style changes significantly. Secondly, when verifying the hyperparameter L, DIMM also shows the opposite monotonicity of the model on the two Pittsburgh datasets and the Aachen dataset. However, unlike N, the ability of DIMM to mine domain-invariant features weakens as L increases; therefore, we chose $L = 4$ as the best parameter. In particular, we find that no matter what values of N and L are taken, the impact on the performance of the model on the TokyoTM dataset is negligible. $stage_n$ represents the stage at which DIMM is embedded into the VGG16 network, and in order to take into account the computational cost, we sequentially choose the deeper positions $stage_n$ = 3, 4, and 5 for verification. Specifically, the exact location of DIMM embedding is before Maxpool in each stage of VGG16. From the last three rows of Table 3, we can see that embedding DIMM in the third stage directly causes the model to collapse on all datasets. With the deeper position of DIMM embedded, the effectiveness of DIMM is gradually exerted. Therefore, we draw the following conclusion: DIMM completes the extraction of domain invariant descriptors in high-level abstract information.

Table 3: Parameter verification experiments for DIMM

Hyper-params		Pitts30k	Pitts250k	Aachen	TokyoTM	Average
		R@1 (%)	R@1 (%)	R@1 (%)	R@1 (%)	R@1 (%)
	1	86.96	88.13	49.21	93.35	79.41
	2	86.24	87.68	52.88	93.36	80.04
N	4	85.72	87.15	62.30	92.89	**82.02**
	8	86.80	87.56	40.84	93.56	77.19
	16	87.56	88.83	41.36	93.68	77.86
	4	85.72	87.15	62.30	92.89	**82.02**
	8	86.06	87.67	53.93	93.46	80.28
L	16	85.99	86.88	53.93	93.06	79.97
	32	87.13	88.49	48.69	93.82	79.53
	64	87.25	88.35	48.49	93.74	79.46
	3	6.37	2.61	10.47	4.98	6.11
$stage_n$	4	84.65	85.18	40.84	89.31	75.00
	5	85.72	87.15	62.30	92.89	**82.02**

In addition, m' in MJT Loss is further verified by us, and we adopt an interval of 0.25. As shown in Table 4, the overall performance of the model on each of the other datasets, except Aachen, slightly decreases as m' increases. When $m' = 1.65$, the trained model has the strongest ability to combat day-night variation. Therefore, we choose $m' = 1.65$ as the best parameter choice for our proposed loss function.

Table 4: Parameter validation experiments for MJT loss

Hyper-param		Pitts30k	Pitts250k	Aachen	TokyoTM	Average
		R@1 (%)	R@1 (%)	R@1 (%)	R@1 (%)	R@1 (%)
m'	0.90	86.14	87.14	59.16	93.47	81.48
	1.15	86.09	87.26	59.16	92.57	81.27
	1.40	85.92	87.13	55.50	92.53	80.27
	1.65	85.72	87.15	62.30	92.89	**82.02**
	1.90	85.89	86.80	42.93	92.85	77.12

4.4 Comparison with State of the Art

In order to verify the overall performance of the proposed algorithm, we compare 9 cutting-edge works in the field of VPR from 2017 to 2024, which are basically implemented based on the classical NetVLAD. In addition, to further ensure a fair comparison, we tried our best to reproduce these algorithms on the same experimental equipment and environment configuration. For the experimental results that were not reported in the original paper and we had insufficient reproduction conditions, we indicated by "/".

As shown in Table 5, overall, the test results of our work on each benchmark data set surpass the vast majority of algorithms, and are close to the state-of-the-art work on each benchmark data set. Specifically, for two Pittsburgh datasets, Pitts30k and Pitts250k, Our DIEONet outperforms the recent works DW-T (2024) [22] and Res2Net-SE-NetVLAD (2023) [23] tested on them by significant margins of 3.08% and 3.05%, respectively. However, for the state-of-the-art works Patch-NetVLAD (2021) [24] and CosPlace (2022) [18] presented on Pitts30k and Pitts250k, our DIEONet still has a gap of 2.98% and 2.55%, respectively. It is worth explaining that Patch-NetVLAD and CosPlace are two-stage re-ranking algorithms, which achieve the purpose of improving accuracy at the cost of sacrificing time cost. In addition, CosPlace introduces a classification algorithm as an aid in the first stage. Therefore, taken together, our end-to-end DIEONet is a more optimal solution, and at the same time contains more "potential".

Table 5: Performance comparison with state of the art algorithms

Algorithms	Pitts30k	Pitts250k	Aachen	TokyoTM
	R@1 (%)	R@1 (%)	R@1 (%)	R@1 (%)
NetVLAD (2017) [4]	84.48	84.40	45.55	90.48
VGG16-GeM (2018) [25]	78.54	76.98	37.70	88.44
HSCNet (2020) [26]	/	/	32.70	/
Patch-NetVLAD (2021) [24]	**88.70**	/	/	/
CosPlace (2022) [18]	88.50	**89.70**	/	/

(Continued)

Table 5 (continued)

Algorithms	Pitts30k	Pitts250k	Aachen	TokyoTM
	R@1 (%)	R@1 (%)	R@1 (%)	R@1 (%)
DeAttention (2023) [27]	85.04	86.07	34.55	91.90
Res2Net-SE-NetVLAD (2023) [23]	/	84.10	/	/
DW-T (2024) [22]	82.64	/	/	/
NocPlace (2024) [28]	/	/	**68.60**	/
DIEONet (Ours)	85.72	87.15	62.30	**92.89**

In particular, on the TokyoTM dataset, our DIEONet achieves the best performance so far, with R@1 outperforming the recent DeAttention (2023) [27] algorithm by 0.99%. As two feature mining schemes, DeAttention emphasizes the use of spatial-like attention mechanism to eliminate dynamic information in scene images, while our DIEONet focuses on designing cascaded interaction layers to directly recalipay static objects in scene images. It turns out that our scheme is more efficient, since direct elimination of dynamic objects may lead to the loss of part of the high-value discriminative information.

Finally, on the Aachen dataset, our algorithm shows absolute advantages. In the challenge of combating the day-night change, our model exceeds the landmark NetVLAD algorithm by 16.75%, and exceeds the 2023 DeAttention algorithm by 34.05%. Although there is a 6.3% difference compared with the state-of-the-art NocPlace (2024) [28], NocPlace is a specialized algorithm against day-night variation, and its training set is a self-made dataset with day-night variation, which is the lack of the training set of Pitts30k. Thus, it shows that our DIEONet can effectively cope with the task challenge with obvious domain variation style, that is, it has a stronger ability to mine salient static information in scene images.

5 Conclusions

This paper proposes a new feature enhancement module DIMM and a new semantic metric function MJT Loss, which are dedicated to mining domain-invariant information in scene images and enhancing the discrimination between positive and negative samples of the model. On the one hand, DIMM implicitly models the interdependencies between different spatial regions of the feature map by cascading specially designed convolutional unit groups, which effectively relabeling the salient information and non-salient information in the feature. On the other hand, MJT Loss combines the "joint processing of multiple samples" mechanism with the original triplet loss, and introduces a new distance constraint term, so that the optimization path of the model stably proceeds in the direction of the global optimum. After extensive comparative experiments and analysis, we demonstrate the superiority of the proposed schemes. In future work, we will explore the lightweighting of DIMM and the clustering of homogeneous samples in MJT Loss, with a view to further improving the generalisation of the VPR model while controlling the complexity of the proposed model.

Acknowledgement: The authors would like to thank the editors and reviewers for their detailed reviews and suggestions on the manuscript.

Funding Statement: This work was supported by the Natural Science Foundation of Xinjiang Uygur Autonomous Region under grant number 2022D01B186.

Author Contributions: The authors confirm contribution to the paper as follows: Shaoqi Hou: Conceptualization, Methodology. Zebang Qin: Implementation, Writing. Chenyu Wu: Translation, Typesetting. Guangqiang Yin, Xinzhong Wang and Zhiguo Wang: Resources, Supervision. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Zhang X, Wang L, Su Y. Visual place recognition: a survey from deep learning perspective. Pattern Recognit. 2021;113:107760. doi:10.1016/j.patcog.2020.107760.
2. Mei C, Sibley G, Cummins M, Newman P, Reid I. A constant-time efficient stereo SLAM system. Paper presented at: The British Machine Vision Conference (BMVC); 2009; London, UK.
3. Churchill W, Newman P. Experience-based navigation for long-term localisation. Int J Robot Res. 2013;32(14):1645–61. doi:10.1177/0278364913499193.
4. Arandjelović R, Gronat P, Torii A, Pajdla T, Sivic J. NetVLAD: CNN architecture for weakly supervised place recognition. IEEE Trans Pattern Anal Mach Intell. 2018;40(6):1437–51. doi:10.1109/TPAMI.2017.2711011.
5. Berton G, Mereu R, Trivigno G, Masone C, Csurka G, Sattler T, et al. Deep visual geo-localization benchmark. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022; New Orleans, LA, USA. p. 5386–97.
6. Oquab M, Darcet T, Moutakanni T, Vo H, Szafraniec M, Khalidov V, et al. DINOv2: learning robust visual features without supervision. arXiv:2304.07193. 2023.
7. Keetha N, Mishra A, Karhade J, Jatavallabhula M, Scherer S, Krishna M, et al. AnyLoc: Towards universal visual place recognition. arXiv:2308.00688. 2023.
8. Galvez-López D, Tardos JD. Bags of binary words for fast place recognition in image sequences. IEEE Trans Robot. 2012;28(5):1188–97. doi:10.1109/TRO.2012.2197158.
9. Perronnin F, Liu Y, Sánchez J, Poirier H. Large-scale image retrieval with compressed fisher vectors. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2010; Francisco, CA, USA. p. 3384–91.
10. Jegou H, Perronnin F, Douze M, Sánchez J, Perez P, Schmid C. Aggregating local image descriptors into compact codes. IEEE Trans Pattern Anal Mach Intell. 2011;34(9):1704–16. doi:10.1109/TPAMI.2011.235.
11. Ali-bey A, Chaib-Draa B, Giguère P. MixVPR: feature mixing for visual place recognition. In: IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); 2023; Waikoloa, HI, USA. p. 2997–3006.
12. Wang R, Shen Y, Zuo W, Zhou S, Zheng N. TransVPR: transformer-based place recognition with multi-level attention aggregation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022; New Orleans, LA, USA. p. 13638–47.
13. Gao H, Zhu X, Lin S, Dai J. Deformable kernels: adapting effective receptive fields for object deformation. In: Proceeding of the International Conference on Learning Representations (ICLR); 2020; Addis Ababa, Ethiopia.
14. Chen Y, Dai X, Liu M, Chen D, Yuan L, Liu Z. Dynamic convolution: attention over convolution kernels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern RecognItion (CVPR); 2020; Seattle, WA, USA. p. 11027–36.
15. Gordo A, Almazán J, Revaud J, Larlus D. End-to-end learning of deep visual representations for image retrieval. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017; Honolulu, HI, USA. p. 237–54.
16. Schroff F, Kalenichenko D, Philbin J. FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015; Boston, MA, USA.

17. Chen W, Chen X, Zhang J, Huang K. Beyond triplet loss: a deep quadruplet network for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017; Honolulu, HI, USA. p. 1320–9.
18. Berton G, Masone CG, Caputo B. Rethinking visual geo-localization for large-scale applications. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022; New Orleans, LA, USA. p. 4868–78.
19. Wang H, Wang Y, Zhou Z, Ji X, Gong D, Zhou J, et al. CosFace: large margin cosine loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2018; Salt Lake City, UT, USA. p. 5265–74.
20. Torii A, Sivic J, Pajdla T, Okutomi M. Visual place recognition with repetitive structures. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2013; Portland, OR, USA.
21. Sattler T, Maddern W, Toft C, Torii A, Hammarstrand L, Stenborg E, et al. Benchmarking 6DOF outdoor visual localization in changing conditions. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018; Salt Lake City, UT, USA. p. 8601–10.
22. Xiong Y, Xu S, Meng G. Distance-ranking-based weighted triplet loss for visual place recognition. In: International Conference on Artificial Intelligence, Human-Computer Interaction and Robotics (AIHCIR); 2023; Tianjin, China.
23. Yin J, Dai K, Cheng L, Xu X, Zhang Z. End-to-end image feature extraction-aggregation loop closure detection network for visual SLAM. In: 35th Chinese Control and Decision Conference (CCDC); 2023; Yichang, China. p. 857–63.
24. Hausler S, Garg S, Xu M, Milford M, Fischer T. Patch-NetVLAD: multi-scale fusion of locally-global descriptors for place recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021; Nashville, TN, USA. p. 14136–47.
25. Radenovic F, Tolias G, Chum O. Fine-tuning CNN image retrieval with no human annotation. IEEE Trans Pattern Anal Mach Intell. 2017;41:1655–68. doi:10.1109/TPAMI.2018.2846566.
26. Li X, Wang S, Zhao Y, Verbeek J, Kannala J. Hierarchical scene coordinate classification and regression for visual localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2020; Seattle, WA, USA. p. 11980–9.
27. Choi SM, Lee SI, Lee JY, Kweon IS. Semantic-guided de-attention with sharpened triplet marginal loss for visual place recognition. Pattern Recognit. 2023;141:109645. doi:10.1016/j.patcog.2023.109645.
28. Liu B, Wang Y, Tao H, Huang T, Tang F, Wu Y, et al. NocPlace: nocturnal visual place recognition via generative and inherited knowledge transfer. arXiv:2402.17159. 2024.