Spatial Pyramid-Enhanced NetVLAD With Weighted Triplet Loss for Place Recognition

Jun Yu, *Member, IEEE*, Chaoyang Zhu, Jian Zhang, Qingming Huang, *Fellow, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— We propose an end-to-end place recognition model based on a novel deep neural network. First, we propose to exploit the spatial pyramid structure of the images to enhance the vector of locally aggregated descriptors (VLAD) such that the enhanced VLAD features can reflect the structural information of the images. To encode this feature extraction into the deep learning method, we build a spatial pyramid-enhanced VLAD (SPE-VLAD) layer. Next, we impose weight constraints on the terms of the traditional triplet loss (T-loss) function such that the weighted T-loss (WT-loss) function avoids the suboptimal convergence of the learning process. The loss function can work well under weakly supervised scenarios in that it determines the semantically positive and negative samples of each query through not only the GPS tags but also the Euclidean distance between the image representations. The SPE-VLAD layer and the WT-loss layer are integrated with the VGG-16 network or ResNet-18 network to form a novel end-to-end deep neural network that can be easily trained via the standard backpropagation method. We conduct experiments on three benchmark data sets, and the results demonstrate that the proposed model defeats the state-of-the-art deep learning approaches applied to place recognition.**

***Index Terms*— Place recognition, spatial pyramid pooling, triplet loss (T-loss), vector of locally aggregated descriptors (VLAD).**

I. Introduction

PLACE recognition has received considerable attention from both academic and industrial communities in the recent decades. Provided a query image, place recognition [1]–[3] means recognizing the place where the image was captured based on a group of previously collected images, likely with a certain form of tag information, e.g., text labels provided by human or GPS data captured by sensors. Place recognition is the core technique of many important applications such as simultaneous localization and mapping (SLAM) [4] and autonomous navigation [5], [6]. However, owing to the large variation in the appearance of places, e.g., viewpoint change, seasonal change, illumination change, weather change, and irrelevant objects such as trees and cars, place recognition has always been challenging in the computer vision and machine learning.

Provided various types of tag information, place recognition can be cast into an image retrieval problem [7] or a classification problem. Prevalent works fit the former context [8]–[14]. This group of methods follows a very similar pipeline in which the local feature descriptors [scale invariant feature transform (SIFT) features] are first extracted, and vocabularies are subsequently built via clustering on the local descriptors to form global feature representations for image retrieval. The typical global feature representations include bag-of-word (BoW) [12], [15], vector of locally aggregated descriptors (VLAD) [16], Hamming embedding [17], and Fisher vector (FV) [18]. One of the weaknesses of the aforementioned approaches is that they rely on holistic feature representations without explicitly distinguishing between different regions of an image. Therefore, the feature representations may lack the image locality information. Hence, some approaches directly use local descriptors [19], [20] for retrieving or reweighting the local descriptors before the global feature creation [21] to emphasize the most informative regions in the images. As a comparison, the classification-based approaches [22], [23] train classifiers for each location using the feature representations derived from quantizing the local feature descriptors.

An apparent trend in place recognition community is the increasing usage of deep learning techniques, which have been proven highly useful in feature learning [24], [25] and pattern classification [26], [27]. Rocco *et al.* [28] transformed place recognition into geometric mapping estimation, in which the deep convolutional neural network (DCNN) was used to simulate the traditional pipeline from image feature extraction to feature correspondence and tune the network parameters using the ground-truth geometric mappings between the

Manuscript received July 21, 2018; revised December 23, 2018; accepted March 30, 2019. Date of publication April 26, 2019; date of current version February 5, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61836002, Grant 61622205, Grant 61620106009, and Grant U1636214, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY17F020009, and in part by the Australian Research Council Projects under Grant FL-170100117, Grant DP-180103424, and Grant IH-180100002. *(Corresponding author: Jian Zhang.)*

J. Yu and C. Zhu are with the School of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: zju.yujun@gmail.com; zack.zcy@gmail.com).

J. Zhang is with the School of Science and Technology, Zhejiang International Studies University, Hangzhou 310012, China (e-mail: jeyzhang@outlook.com).

Q. Huang is with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: qmhuang@ucas.ac.cn).

D. Tao is with the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney, Darlington, NSW 2008, Australia, and also with the School of Computer Science, Faculty of Engineering and Information Technologies, The University of Sydney, Darlington, NSW 2008, Australia (e-mail: dacheng.tao@sydney.edu.au).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2908982

training images. The need for ground-truth geometric mapping, however, limits the application of this approach. Arandjelovic *et al.* [29] modified the traditional VLAD method such that it can be inserted into the DCNN to form a net vector of locally aggregated descriptors (NetVLAD) method for deep feature extraction. Typically, the case of place recognition is that only inexact GPS data of the training images are provided as tag information, and NetVLAD can handle this problem well. However, NetVLAD focuses on global feature learning, and the global representations do not reflect the regional discrepancy contained in an image. Therefore, it should be difficult for NetVLAD to retrieve images based on partial matching in the presence of occlusion and background clutter. Hence, Noh *et al.* [30] proposed to highlight important regions in the feature maps using an attention layer that learns an attention score for each location of the feature map through a two-layer convolutional neural network (CNN) with softplus activation. Chen *et al.* [31] further extended the attention mechanism to the channel domain such that the important channels could also be highlighted.

However, the current research on deep learning-based place recognition still exhibits some weaknesses. First, the attention-based approaches concentrate only on the local regions of an image without leveraging the spatial structures of the objects in the image that are paramount to place recognition. Neither does the NetVLAD method because the NetVLAD features are formed by quantizing the local descriptors everywhere in the feature maps. Li *et al.* [32] solved this problem by applying spatial pyramid partition to images, extracting deep-level feature descriptors from each partition through a pretrained DCNN, and subsequently computing the VLAD feature for each partition and concatenating them to form the final image feature representation. Nevertheless, this is not an end-to-end scheme because it does not train the DCNN. Next, the current methods extensively use the triplet loss (T-loss) function that is designed to congregate the feature representations of similar semantic together and disperse those with different semantics. However, the optimization of the T-loss considers only the topology constraints among the feature representations and ignores the temporal constraints between adjacent epochs. This may lead to a problem in that similar data may be farther from one another in certain epochs than in the previous one. Therefore, the training may converge to suboptimal feature representations.

Here, we propose the spatial pyramid-enhanced NetVLAD (SPE-NetVLAD) approach with the weighted T-loss (WT-loss) for place recognition. The key in the deep learning scheme is twofold. First, we propose the spatial pyramid-enhanced VLAD (SPE-VLAD) layer in the feature learning. The SPE-VLAD layer first divides the feature maps into patches according to certain pyramid structure and computes each patch's VLAD feature that is subsequently stacked together to form the enhanced feature representation. The VLAD feature is the concatenation of the sum of errors between each cluster center and the local feature descriptors related to this center with certain membership probability [29]. Next, our WT-loss imposes weights on the terms of traditional T-loss. The weights take effect only when the semantically similar data are found to be farther away than in the previous epoch during the optimization. We carefully designed the weights such that the WT-loss can push these data closer again in the next epoch under this situation. It is noteworthy that the membership probabilities and the weights in the loss are all determined adaptively in the optimization process, which can be easily achieved via backpropagation.

Several aspects of the proposed method are noteworthy, which are as follows.

1) The proposed SPE-VLAD layer combines the image pyramid structure and VLAD together to extract the feature representations that contain both spatial structural information and local regional characteristics of the images. Meanwhile, the SPE-VLAD layer can be inserted into any DCNN framework to enhance the descriptive power of the learned features.
2) The proposed WT-loss is carefully designed to congregate the semantically similar feature representations tighter whenever the distances between them tend to increase in the adjacent epochs of the optimization process. The opposite constraint can be simultaneously imposed on the dissimilar features.
3) We encode the GPS information of the training images into the WT-loss by selecting the positive and negative samples with respect to a query image through the compositional usage of the GPS data and Euclidean distance.

The remainder of this paper is organized as follows: the relevant works are reviewed in Section II, and the detailed description of our model is provided in Section III. Section IV presents the experimental results performed on two benchmark data sets and the comparison with the state-of-the-art methods. Section V concludes this paper.

II. Related Works

A. Feature Extraction in Place Recognition

The most typically used feature extraction method in place recognition is extracting SIFT feature descriptors first and building the fine vocabulary using these descriptors, and subsequently quantizing the SIFT descriptors of an image into a histogram of visual words. The typical works of this type include [8]–[10], [12]–[14], and [23]. Torii *et al.* [8] performed a view synthesis using the standard ray tracing with bilinear interpolation before the feature extraction. Sattler *et al.* [11] transferred the typically used scheme to a 3-D scenario via quantizing the 3-D point descriptors recovered by the shape from motion algorithm [33] into vocabularies and matching an image descriptor to a set of 3-D points using the vocabularies. Torii *et al.* [13] combined the histogram of visual words with the repetitive structure as useful clues for place recognition.

Recently, a similar feature extraction scheme is introduced into the construction of DCNNs [29]. In [29], the CNN without fully connected layers uses an input image and produces a feature map of size $H \times W \times D$, which can be interpreted as a set of $H \times W$ D-dimensional local descriptors. Subsequently, the authors performed K-means clustering and quantized the local descriptors of an image into a vector representation of

size $K \times D$. Each D-dimensional part of this vector records the sum of residuals between each cluster center and the local descriptors that are members of this cluster. The membership variants are learned during the training of the DCNN.

The histogram of the visual words representations in the methods above is all global features that employ the local descriptors extracted everywhere from an image and ignore the spatial structural information in the image. Therefore, a collection of approaches was considered to encode the regional characteristics in feature learning. Knopp *et al.* [21] computed the confusing scores with respect to different regions of the database images and suppressed the local descriptors with confusing scores greater than a certain threshold before quantization. Sattler *et al.* [19] and Li *et al.* [20] conducted geometric matching directly on local descriptors for the similarity evaluation. Rocco *et al.* [28] transferred this scheme into the DCNN framework, in which the local feature descriptors were similarly extracted as in [29], and geometric matching was achieved via a matching layer. Noh *et al.* [30] computed a distinct attention score for each location of a feature map using a two-layer CNN with softplus activation to highlight the local area important to feature discrimination. Chen *et al.* [31] extended this strategy to the channel domain by computing the distinct weight for each feature map. The spatial and channel attention can be simultaneously used in the DCNNs.

Despite the locality-aware ability, the attentive features still cannot reflect the structural information contained in the image, such as the contour of a building and the layout of a scene. Hence, some researchers combined the image spatial pyramid structure and VLAD to extract the locality-aware global feature representations [32]. Specifically, they first constructed the image pyramid patch structure and extracted the patchwise deep-level features using the pretrained DCNN; subsequently, they computed the VLAD feature for each pyramid patch and stacked all the patchwise VLAD features to form the final feature representation of an image. The advantage is that the image pyramid structure encodes the structural information of the image, but this method treats the DCNN feature extraction and VLAD feature computing as separate stages without training the DCNN in an end-to-end manner.

B. Usage of Tag Information and the Loss Function

Supervised place recognition methods use ground-truth labels to evaluate the retrieval result [10], [21] or train the classification model [22], [23], [28], [30]. In [21], the ground truth was obtained manually by inspecting the visual correspondence between the query and the top retrieved image. Rocco *et al.* [28] and Noh *et al.* [30] trained the DCNN through loss functions that denote the error between the estimated labels and ground-truth labels. Song and Snavely [22] and Gront *et al.* [23] built local SVMs for each local area or each location for place recognition.

However, the accurate labels for the places are not always available. In many cases, only inexact GPS data are available and this motivates the requirement for weakly supervised [34] place recognition. Chen *et al.* [9] used GPS to narrow the searching space in an image database. Arandjelovic *et al.* [29] and Chen *et al.* [31] achieved network training using T-loss, in which the positive samples (PSs) and negative samples with respect to each query image in the feature space were determined based on the GPS data.

The typical loss functions used by deep neural network-based place recognition are the L_2 loss [35], cross-entropy loss [36], and T-loss. Using the ground-truth geometric transformations between the training images, Rocco *et al.* [28] trained the DCNN for image matching by minimizing the sum of squared distances between the estimated transformations and the ground-truth transformations. The L_2 loss is prone to the influence of outliers.

Furthermore, the DCNN parameters can be estimated through the cross-entropy loss when the ground-truth place labels are available [30]. However, more typically emerged tag information in place recognition problems is the inexact GPS information, which significantly degrades the effect of the supervised models above.

In this situation, many approaches adopt the T-loss, aiming to reduce the distances between a query image and the PSs about it while enlarging the distances between the query image and the negative samples about it in the feature space [29], [31]. The T-loss exploits only the spatial topology constraints that congregate semantically similar feature representations together and disperse the dissimilar feature representations. It does not consider the distance changes between any two adjacent epochs. Therefore, it is possible that the distance between the query image and the PSs becomes larger than the distance computed in the previous epoch. The current loss functions have not captured this problem.

III. Place Recognition via Spatial Pyramid-Enhanced NetVLAD Approach

A. Framework

In view of the typically praised performance of the VGG-16 [37] and ResNet-18 [38], we use the feature learning parts of them as the base networks in our framework. To equip the traditional VLAD features with the capability of describing the spatial structural information of an image, we propose to partition the image into multiple nonoverlapping local patches according to a hierarchical image pyramid structure and concatenate the VLAD features of these patches to form the SPE-VLAD feature representations. To enable this feature computation in the deep neural network, we built an SPE-VLAD layer and inserted it after the last convolutional layer in the base networks. The SPE-VLAD layer encodes the locality of the feature maps, such that it can be viewed as a locality-enhanced module.

In addition, we propose a WT-loss function that can push a query image closer to its PSs in the feature space, whenever the distances between the query image and the PSs tend to increase with the training progress. This is to alleviate the problem existing in the T-loss function that exploits only the distance constraints among training data within each separate epoch. We can also impose the opposite constraint on the negative samples. It is noteworthy that in the selection of the positive and negative samples, we used both the image feature

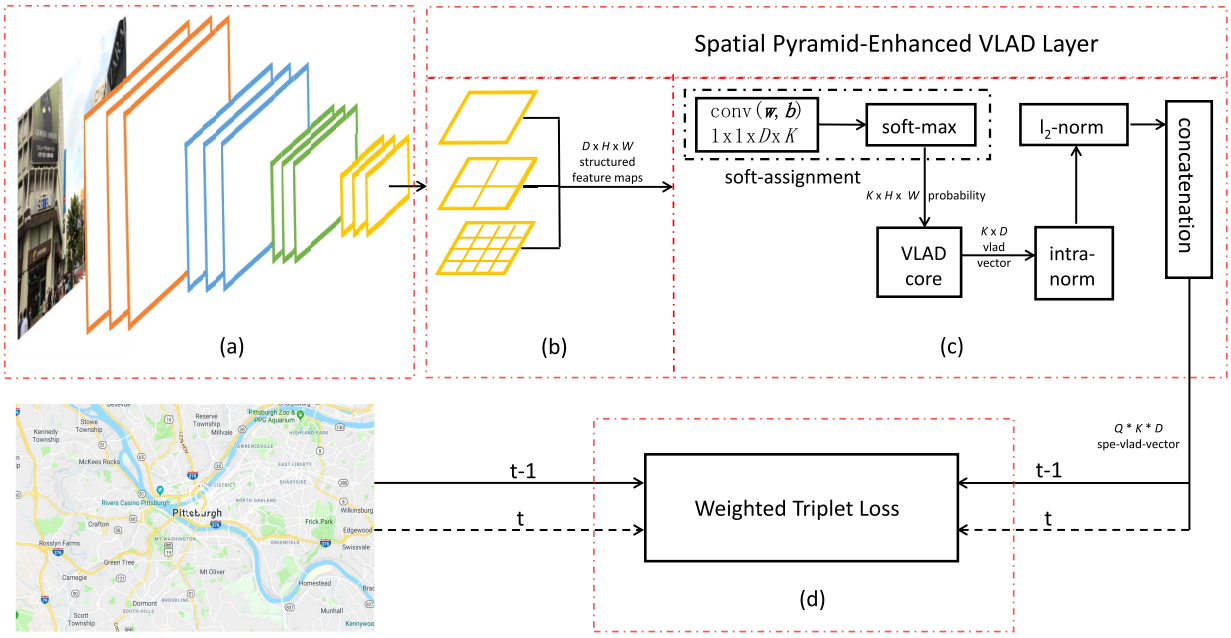


Fig. 1. Illustration of the proposed framework for place recognition, in which we learn the SPE-VLAD feature representations through a WT-loss function with GPS data as tag information. The feature extraction part is a (a) base network (the same as in [37] or [38] except that the max5 layer in VGG-16 and the global average pooling layer in ResNet-18 are removed) followed by a locality-enhanced module named the SPE-VLAD layer that encodes the structural characteristics via (b) image pyramid into the (c) NetVLAD feature computation module, in which we exploit the softmax layer to achieve the soft assignment of the local descriptors to the visual words into the codebook, and subsequently compute the normalized VLAD features following [29]. (d) WT-loss layer uses the feature representations generated in the adjacent epochs to force the queries to approach the PSs and avoid the negative ones. We will spread the WT-loss layer in detail in Fig. 3.

representations and GPS data, thereby rendering our approach, particularly useful in weakly supervised place recognition.

Fig. 1 shows the architecture of the proposed deep learning framework for place recognition. The feature extraction scheme contains three parts: Fig. 1(a) shows a base network that shares the same pipeline as in [37] or [38], except that the max5 layer in VGG-16 and global average pooling layer in ResNet-18 are removed; Fig. 1(b) shows an image pyramid structure that divides the feature maps into hierarchical and nonoverlapping patches with the aim of encoding image localities into the feature learning; Fig. 1(c) shows the NetVLAD feature computation module that learns the SPE-VLAD feature representations by concatenating the NetVLAD features extracted from the patches of the feature maps. The combination of Fig. 1(b) and (c) is the proposed SPE-VLAD layer that learns not only the global but also the structural characteristics of the images; Fig. 1(d) shows the proposed WT-loss layer that uses the feature representations generated in adjacent epochs to force the queries to approach the PSs and avoid the negative ones. We will discuss the WT-loss layer in detail in Section III-C.

To provide the readers a good insight into the formal descriptions of the methods, we list all the mathematical notations to be used herein in Table I.

B. SPE-NetVLAD

1) VLAD Feature Extraction: In the extraction of the VLAD, the local image feature descriptors, typically the SIFT descriptors, are first extracted and grouped into K clusters with the K-means algorithm; subsequently, a codebook of K visual words is constructed using the center descriptor of each cluster. Furthermore, each local descriptor is assigned to its nearest visual word and the VLAD feature descriptor is computed by accumulating the residual vector between each visual word and the local descriptors that have been assigned to it. This characterizes the distribution of the vectors with respect to the center.

Let $\mathbf{x}_i = [\mathbf{x}_{1i}, \mathbf{x}_{2i}, \ldots, \mathbf{x}_{Di}]^T (i = 1, \ldots, N)$ be the extracted SIFT descriptors, and $\mathbf{c}_k = [\mathbf{c}_{1k}, \mathbf{c}_{2k}, \ldots, \mathbf{c}_{Dk}]^T (k = 1, \ldots, K)$ be the center of each cluster. Each entry of the VLAD feature representation $\mathbf{v}_{ji}$ can be computed by adding up all the errors between the entry of each center and the entries of the local descriptors assigned to this center

$$\mathbf{v}_{jk} = \sum_{NN(\mathbf{x}_i)=\mathbf{c}_k} \mathbf{x}_{ji} - \mathbf{c}_{jk}, (1 \le j \le D, 1 \le k \le K) \quad (1)$$

where $NN(\mathbf{x}_i)$ denotes the nearest visual word of $\mathbf{x}_i$. It is obvious that the VLAD feature representation $\mathbf{v}$ is a feature vector of size $D \times K$.

2) NetVLAD Layer in Neural Network: Due to the typically known significance in image retrieval, VLAD feature extraction has been transferred to the deep learning framework [29]. Because the original VLAD is derived from the SIFT feature descriptors, some strategy to mimic the SIFT descriptors using CNN features should be available. Suppose $\mathbf{X} \in R^{D \times H \times W}$ are the feature maps of an input image output by the last convolution layer of the neural network. We can decompose $\mathbf{X}$ into $H \times W$ number of D-dimensional feature descriptors

TABLE I
MATHEMATICAL NOTATIONS USED IN THIS PAPER

Notations	Description
$\mathbf{x}_i$	a $D \times 1$-dimensional local feature descriptor
$\mathbf{C}$	the codebook ($D \times 1$-dimensional center vectors of K clusters)
$\mathbf{c}_k$	a visual word in the codebook $\mathbf{C}$ (the $D \times 1$-dimensional center vector of one cluster)
$\mathbf{x}_{ji}$	the j_{th} entry of $\mathbf{x}_i$
$\mathbf{c}_{jk}$	the j_{th} entry of $\mathbf{c}_k$
$\mathbf{V}$	the $(K \times D) \times 1$-dimensional VLAD/NetVLAD descriptor of an image
$\mathbf{v}_k$	the k_{th} part of $\mathbf{V}$ w.r.t. $\mathbf{c}_k$ with dimension $D \times 1$
$\mathbf{v}_{jk}$	the j_{th} entry of $\mathbf{v}_k$
$NN(\mathbf{x}_i)$	the nearest visual word to $\mathbf{x}_i$ with dimension $D \times 1$ in the codebook
$\mathbf{X}$	the $D \times H \times W$-dimensional feature maps of an image (H and W are the height and width of a feature map)
$\mathbf{W}_p$	the $1 \times 1 \times D \times K$-dimensional convolution tensor (1×1 convolutional kernels w.r.t. K cluster centers applied to D feature maps)
$\mathbf{b}_p$	the $K \times 1$-dimensional bias vector
$\otimes$	the convolution operation
$\oplus$	the addition operation between a tensor and a vector (automatically stretches the vector into compatible form)
$\mathbf{P}$	the $K \times H \times W$-dimensional probability tensor (the probabilities each local feature descriptor $\mathbf{x}_i (1 \leq i \leq H \times W)$ is assigned to one of the clusters)
$\mathbf{X}_{h,w}$	the $D \times 1$-dimensional local feature descriptor $\mathbf{x}_i$ extracted from the h_{th} line and w_{th} column of the D feature maps $\mathbf{X}$
$\mathbf{P}_{k,h,w}$	a scalar indicating the probability the local feature descriptor $\mathbf{X}_{h,w}$ belongs to the k_{th} cluster
$\mathbf{V}_i^{sp}$	the i_{th} $(Q \times K \times D) \times 1$-dimensional SPE-NetVLAD descriptor of an image (Q is the patch number of the spatial pyramid structure)
N_i^p	the number of positive samples of $\mathbf{V}^{sp}$
N_i^n	the number of negative samples of $\mathbf{V}^{sp}$

$\mathbf{x}_i (i = 1, \ldots, H \times W)$ with each of them representing the local features at specific local positions of the input image. These local feature descriptors can be regarded as the counterparts of the SIFT descriptors used by traditional VLAD in NetVLAD.

To ensure that the NetVLAD equipped neural network can be trained in an end-to-end manner, the NetVLAD layer has to be differentiable. However, the traditional VLAD assigns each SIFT descriptor to a cluster center by nearest-neighbor searching, whose derivative with respect to the feature descriptors is ambiguous. Hence, the hard assignment of VLAD is substituted by a soft assignment through a convolutional layer and a softmax layer in NetVLAD

$$\mathbf{P} = \text{softmax}(\mathbf{W}_p \otimes \mathbf{X} \oplus \mathbf{b}_p) \tag{2}$$

where $\mathbf{W}_p \in R^{1 \times 1 \times D \times K}$ represents a series of 1×1 convolutional kernels with respect to the K cluster centers applied to $\mathbf{X}$, $\mathbf{P} \in R^{K \times H \times W}$ yields the probabilities according to which each local deep feature descriptor $\mathbf{x}_i$ is assigned to one of the clusters, $\mathbf{b}_p$ is the $K \times 1$-D bias vector, $\otimes$ is the convolution operation, and $\oplus$ stretches a vector such that it can be added to a tensor. The trained codebook $\mathbf{C} \in R^{D \times K}$ can be stored in the variable class of Pytorch, such that it can be learned through backpropagation. Unless otherwise denoted, K is set to 64 throughout our implementation.

If we divide the NetVLAD feature representation $\mathbf{V}$ into K parts, the kth part of $\mathbf{V}$ is provided by

$$\mathbf{v}_k = \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{P}_{k,h,w} (\mathbf{X}_{h,w} - \mathbf{c}_k) \tag{3}$$

where $\mathbf{X}_{h,w}$ is equivalent to a certain $\mathbf{x}_i$, and $\mathbf{v}_k \in R^D$. For the robustness, we align $\mathbf{v}_k$ into a matrix, and subsequently normalize it rowwise (intranormalization) first, and stretch it into a vector $\mathbf{V}$ with dimensions $K \times D$, and finally, L_2-normalize in its entirety.

3) Spatial Pyramid-Enhanced NetVLAD: Equation (3) sums over the residuals within each cluster, weighted by the probabilities according to which each local feature descriptor $\mathbf{X}_{h,w}$ is assigned to cluster $\mathbf{c}_k$. However, if two descriptors locate symmetrically in the opposite direction with respect to the cluster center, the summation of their residuals could be close to zero. Similar NetVLAD feature representation will emerge if all the local descriptors that have been assigned to $\mathbf{c}_k$ locate at approximately the same position as $\mathbf{c}_k$.

This degradation in the discrimination capability can be resolved by appending the spatial structural information to the NetVLAD feature representation. Specifically, the structural information can be depicted through a spatial pyramid, which is a layered structure representing the coarse-to-fine partition of an image. There should be $2^{n-1} \times 2^{n-1}$ patches at level n for a spatial pyramid of depth N. It is noteworthy that for the first level, the pyramid contains only one patch that is the holistic feature map. Subsequently, we propose to first pyramidalize the feature maps produced by the last convolutional layer of the base network and subsequently concatenate the NetVLAD representations of all the patches in the pyramid to create the SPE-NetVLAD representation.

Fig. 2 shows a three-level spatial pyramid that contains 21 patches, i.e., the holistic feature map in level 1, the four patches in level 2, and the 16 patches in level 3. The pipeline of the SPE-NetVLAD extraction is shown in Fig. 1 where Fig. 2(a) shows the base network, Fig. 2(b) shows a spatial pyramid structure of the feature map, and Fig. 2(c) shows the VLAD computation module.

We follow Algorithm 1 to compute the SPE-NetVLAD representation.

C. Weighted Triplet Loss Function

The typically used loss function in place recognition is the T-loss function. It is designed to minimize the distances between the query image and the database images with the closest semantic while maximizing the distances between the query image and the database images with the farthest semantic in the feature space. These most similar images are named as the PSs, and the most dissimilar images are the negative samples.

Suppose the SPE-NetVLAD feature representations of the query images in the training set are $\mathbf{V}_i^{\text{sp}} (i = 1, \ldots, N)$, and

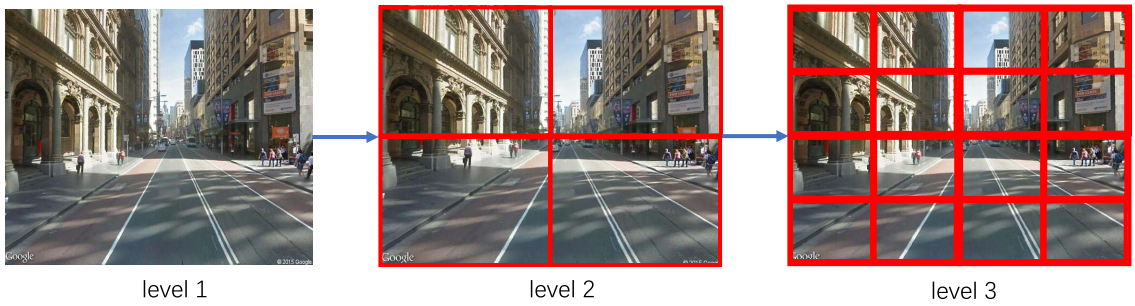


Fig. 2. Three-level spatial pyramid of an image containing 21 patches.

Algorithm 1 Computation of the Proposed SPE-NetVLAD Feature Representation

Input: Feature map $\mathbf{X} \in R^{D\times H\times W}$ and pyramid level N.
Output: SPE-NetVLAD feature representation $\mathbf{V}^{sp}$.
1: **while** $N! = 0$ **do**
2: Calculate the $2^{N-1} \times 2^{N-1}$ partition of the feature map.
3: Calculate NetVLAD feature of image patches separately according to (3).
4: $N = N - 1$.
5: **end while**
6: Concatenate the NetVLAD features of all the patches to form $\mathbf{V}^{sp}$.

the corresponding positive and negative samples are $\mathbf{V}_{i_p}^{\text{sp}}$ and $\mathbf{V}_{i_n}^{\text{sp}}$, respectively, the traditional T-loss function minimizes

$$\sum_{i=1}^{N}\sum_{p=1}^{N_i^p}\sum_{n=1}^{N_i^n} \text{hinge}\left(d\left(\mathbf{V}_i^{\text{sp}}, \mathbf{V}_{i_p}^{\text{sp}}\right) + m - d\left(\mathbf{V}_i^{\text{sp}}, \mathbf{V}_{i_n}^{\text{sp}}\right)\right) \tag{4}$$

where $\text{hinge}(\lambda) = \max(\lambda, 0)$, $d(\cdot)$ represents the semantic distance between the queries and the corresponding positive and negative samples in the feature space, N_i^p and N_i^n denote the number of positive and negative samples of $\mathbf{V}_i^{\text{sp}}$, respectively, and m is a constant parameter defining the margin.

However, the traditional T-loss function exploits the spatial distance constraints between features within each individual epoch, and the interdependence between epochs is completely ignored. This could occasionally lead to a result where the query image is somehow pulled farther away from its PSs than in the previous epoch. Hence, the optimization process may be prolonged and even converge to a suboptimal solution. Hence, we propose to multiply the first and third terms of formula (3) by certain weights $\{\mathbf{w}_{\text{ip}}\}^t$ and $\{\mathbf{w}_{\text{in}}\}^t$ to reformulate (4) into a WT-loss function

$$\sum_{i=1}^{N}\sum_{p=1}^{N_i^p}\sum_{n=1}^{N_i^n} hinge\left(d\left(\mathbf{V}_i^{sp}, \mathbf{V}_{i_p}^{\text{sp}}\right)\{\mathbf{w}_{\text{ip}}\}^t + m - d\left(\mathbf{V}_i^{\text{sp}}, \mathbf{V}_{i_n}^{\text{sp}}\right)\{\mathbf{w}_{\text{in}}\}^t\right) \tag{5}$$

where the superscript t indicates the ordinal of the epochs, and d is the Euclidean distance.

In this loss function, we anticipate that these weights can be beneficial to learning more discriminative feature representations at each epoch than the features learned at the previous epoch during the course of optimization. Hence, if we discover that the distance between a query image and one of its PSs is larger than the distance computed in the previous epoch, we must push them closer again at the next epoch using these weights. Apparently, this constraint is imposed in the temporal domain, and this is the difference from the spatial constraint used in the traditional T-loss. Of course, the opposite operation can be performed for the negative samples simultaneously in the feature space.

Specifically, the weights used in (5) can be described as

$$\begin{cases} \{\mathbf{w}_{\text{ip}}\}^t = \max(\exp(\{d_{ii_p}\}^t - \{d_{ii_p}\}^{t-1}), 1) \\ \{\mathbf{w}_{\text{in}}\}^t = \min(\exp(\{d_{ii_n}\}^t - \{d_{ii_n}\}^{t-1}), 1) \end{cases} \tag{6}$$

where d_{ii_p} represents the distance between $\mathbf{V}_i^{\text{sp}}$ and $\mathbf{V}_{i_p}^{\text{sp}}$; d_{ii_n} represents the distance between $\mathbf{V}_i^{\text{sp}}$ and $\mathbf{V}_{i_n}^{\text{sp}}$. Ideally, the distance between a query image and one of the PSs should decrease monotonously during training, and accordingly $\{\mathbf{w}_{\text{ip}}\}^t \leq 1$. We set $\{\mathbf{w}_{\text{ip}}\}^t = 1$ in this situation because the gradients have been correctly calculated and passed backward, and extra constraints are not required. Whenever the distance grows larger in two adjacent epochs, $\{\mathbf{w}_{\text{ip}}\}^t \geq 1$, it can guarantee that the distance becomes smaller again at the next epoch. This ensures that the true PS continually gather in the feature space instead of accidentally being pulled away owing to the inaccurate gradient direction calculated per batch. $\{\mathbf{w}_{\text{in}}\}^t$ functions similarly for the negative samples. The working scheme of the proposed WT-loss can be described in Fig. 3, where we show the first three epochs of the optimization procedure. The red dots represent one query image and its two PSs, and we observed that the distances between them after epoch 2 are somehow larger than those after epoch 1. By explicitly imposing the weights, they become closer to one another after epoch 3.

We used the existing PyTorch-based framework to optimize the deep neural network.

1) Connection to Manifold Learning: The graph Laplacian has been widely accepted and often combined with manifold assumptions for unsupervised learning [39]–[42]. The traditional Laplacian-driven manifold methods establish a neighborhood for each data sample using its nearby samples and minimize the distances between the feature representation of the current sample and those of a few closest samples in

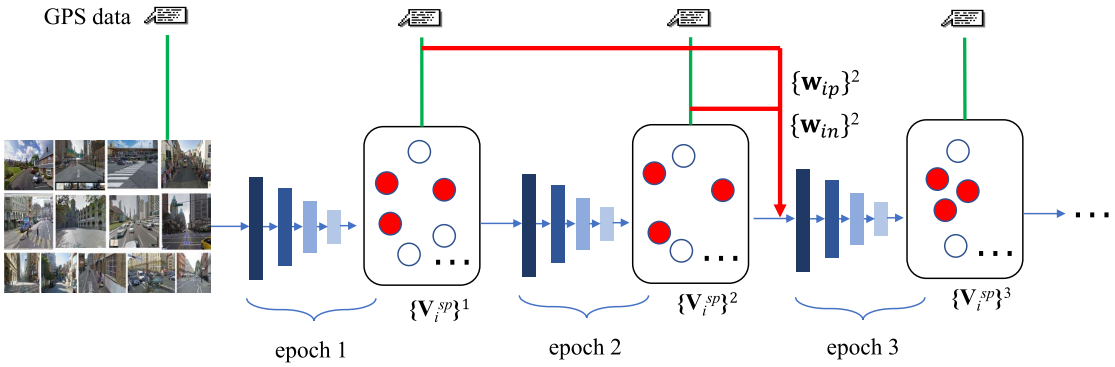


Fig. 3. Working scheme of the proposed WT-loss function, where we use the $\{d_{ii_p}\}^t$ and $\{d_{ii_p}\}^{t-1}$ constructed at the previous epoch to construct the weights $\{\mathbf{w}_{\text{ip}}\}^t$ to be used in the current epoch and produce better feature representations $\{\mathbf{V}\}^{t+1}$ in the next epoch. Similarly, we can also impose constraints on the negative samples.

the neighborhood. If we define these closest samples as PSs, those samples far from the current sample can subsequently be regarded as the negative samples. Accordingly, manifold learning can be formulated as

$$\min_{\mathbf{y}_i} \sum_{i=1}^{N} \sum_{p=1}^{N_i^p} ||\mathbf{y}_i - \mathbf{y}_{i_p}||^2 \mathbf{w}_{\text{ip}} \tag{7}$$

where $\mathbf{y}_{i_p}$ means one of the unknown feature representations of the PSs with regard to $\mathbf{y}_i$, and

$$\mathbf{w}_{\text{ip}} = \exp(-(d_{ii_p})) \tag{8}$$

with d the Euclidean distance.

It is apparent that the proposed WT-loss function shares the similar form with the objective of manifold learning. The major differences between them are twofold. First, the manifold methods exploit only the neighboring samples of each sample to learn the topology of the data set, whereas the WT-loss function can exploit not only the PSs but also the negative samples of each image in the feature space. Second, similar to the traditional T-loss function, the manifold methods exploit only the spatial distance constraint in feature learning, whereas the WT-loss function encodes temporal constraints in feature learning. Whenever the distances between the query image and the PSs tend to increase in the adjacent epochs, the WT-loss function is triggered to congregate them closer in the next epoch. WT-loss functions similarly for the query image and its negative samples.

2) Difference Between Weighting Scheme and Learning Rate Decay Scheme: Although to some extent, the learning rate scheme can be viewed as a weighting scheme, two distinguishable differences exist between the proposed weighting scheme and the learning rate decay scheme. First, this weighting scheme is dynamic among the epochs and dependent on the previous confidence of similarity, while the learning rate stays rather stale. Second, the proposed weighting scheme is different for the positive and negative samples, such that the PSs are pushed closer together and the negative samples pulled farther apart, while the learning rate imposes the same effect on the positive and negative samples.

IV. Experiments

With the theoretical analysis above of the proposed SPE-NetVLAD deep learning framework, we now demonstrate its significance on the place recognition problem. First, we testify the effect of the proposed WT-loss function by inserting it directly behind the VGG-16 base network without employing the SPE-VLAD layer. Second, we evaluate the proposed SPE-NetVLAD feature representations by comparison with several state-of-the-art feature representations including the VLAD features, NetVLAD features, and spatial attention (SA) features. Moreover, we show the place recognition results derived from the SPE-NetVLAD feature representations learned using the WT-loss function. These experiments were conducted on three typically used benchmark databases. The place recognition on two of the three databases is cast as an image retrieval problem, in which the query images are first transformed into the feature space where the images from the same place should be adjacent to one another, whereas the images from different places should be relatively far. Hence, the results are denoted by the recall of the retrieval. The place recognition on the rest, one database is a typical image classification task with class labels. To provide the readers a better insight into the proposed approach, we also show the retrieved images by different methods and investigate why our results are better than the results of other approaches.

A. Data Sets

We conducted experiments on three publicly available data sets.

1) *Pittsburgh (Pitts30k):* It contains 30 k database images downloaded from Google Street View, and 22 k test queries derived at the same place but captured at different times. The train, validation, and test sets are of approximately the equal size, each containing 10 k database images and 7 k queries. The three sets are constructed such that the images from different sets are captured from geographically different places.
2) *TokyoTimeMachine:* It contains 49 k database images and 7 k queries that are divided into training and

TABLE II
BRIEF DESCRIPTION OF THE DATA SETS USED IN EXPERIMENTS

Dataset	number of database images	number of queries	weather variation	season variation	illumination variation	structure variation	viewpoint variation
Pitts30k-train	10,000	7,416	small	small	small	small	large
Pitts30k-val	10,000	7,608	small	small	small	small	large
Pitts30k-test	10,000	6,816	small	small	small	small	large
TokyoTimeMachine-train	49,104	7,277	medium	large	large	small	large
TokyoTimeMachine-val	49,056	7,186	medium	large	large	small	large
Places365-Standard-train	142,100	\	medium	medium	medium	large	\
Places365-Standard-val	2,900	\	medium	medium	medium	large	\
Places365-Standard-test	2,900	\	medium	medium	medium	large	\

validation sets with approximately the same size. Similarly, the images in the training and validation sets are captured from different places.

3) *Places365-Standard:* It contains over 1.8 million training images from 365 scene categories, each of which contains around 5000 training images and 100 validation images. For the sake of computational efficiency, we randomly select 29 scene categories for our experiments. Since the testing images are not labeled, we use the validation images as the testing set, and randomly select 100 images per category from the training images as a validation set. Accordingly, the number of training images per category is reduced to 4900.

For more details on the three data sets, see Table II.

B. Evaluation Metric

For Pittsburgh and TokyoTimeMachine data sets, the query image is deemed as correctly recognized if at least one of the retrieved database images is within 10 m from the ground-truth position (GPS data) of the query. We will show the recall performance (the percentage of correctly recognized queries) against the number of retrieved database images.

For Places365-Standard data set, the recognition rates are described by five metrics, top-1 to top-5. Instead of WT-loss function, the neural network is trained via a softmax loss function. Given a test image, the softmax function, therefore, produces a probability for each category denoting the possibility that the image belongs to the category. top-1 means the occurrence rate that the top one probability just represents the ground-truth category. Similarly, top-n4 means the occurrence rate that the ground-truth category is included in the categories denoted by the top n probabilities.

C. Implementation Details

The VGG-16 [29] network and ResNet-18 [38] network pretrained on ImageNet are adopted as our base networks to verify the generalization ability of the proposed approach. During the training and inference, we replaced all descendant layers behind the last convolutional layer with the NetVLAD layer, SPE-VLAD layer, and SA layer, tailed by the WT-loss, T-loss layer, or a softmax layer, to construct various network architectures. Unless otherwise stated, none of the following experiments perform principal component analysis dimension reduction and whitening, or any other postprocessing techniques, as our primary concern is on the SPE-VLAD and WT-loss layer.

The feature map produced by the last convolutional layer is of three dimensions, $W \times H \times D$, representing the width, height, and number of filters, respectively. The images are resized to 256×256. In order to balance the recognition accuracy and the computational efficiency, we set $N = 2$ for the image pyramid used in our experiments, which means the pyramid has two layers, wherein the first layer contains only one patch representing the whole feature map, and the second layer contains four patches with each patch denoting a quarter of the feature map. We set the batch size to eight tuples for all experiments (a tuple contains a query image, the positive images, and the negative images). The learning rate $\alpha = 0.001$ is reduced by half every five epochs, and the momentum and weight decay are 0.9 and 0.001, respectively.

The WT-loss layer exploits not only the PSs but also the negative samples with respect to the query images. As the Pittsburgh and TokyoTimeMachine data sets involved in the experiments do not contain the exact label information provided by humans, we determine the positive as well as negative samples based on the GPS data, which denote the places where the images are captured. Among all the images, we conjecture that the images whose GPS data are within 10 m from the query image are more likely to be the PSs, while the images whose GPS data are beyond 25 m from the query image are more likely to be the negative samples. Suppose this criterion is described as θ_{GPS}. In choosing the potential PSs

$$\theta^{p}_{\text{GPS}}\left(\left\|\mathbf{q}_i^{\text{GPS}} - \mathbf{q}_j^{\text{GPS}}\right\|^2\right) = \begin{cases} 1, & \left\|\mathbf{q}_i^{\text{GPS}} - \mathbf{q}_j^{\text{GPS}}\right\|^2 < 10 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

whereas in choosing the potential negative samples

$$\theta^{n}_{\text{GPS}}\left(\left\|\mathbf{q}_i^{\text{GPS}} - \mathbf{q}_j^{\text{GPS}}\right\|^2\right) = \begin{cases} 1, & \left\|\mathbf{q}_i^{\text{GPS}} - \mathbf{q}_j^{\text{GPS}}\right\|^2 > 25 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $\mathbf{q}_i$ and $\mathbf{q}_j$ are the images, and $\mathbf{q}_i^{\text{GPS}}$ and $\mathbf{q}_j^{\text{GPS}}$ are the corresponding GPS data.

With the positive and negative candidates, we can determine the exact positive and negative samples based on the Euclidean distances between them and the query image in the feature space. We designate a number of positive candidates with

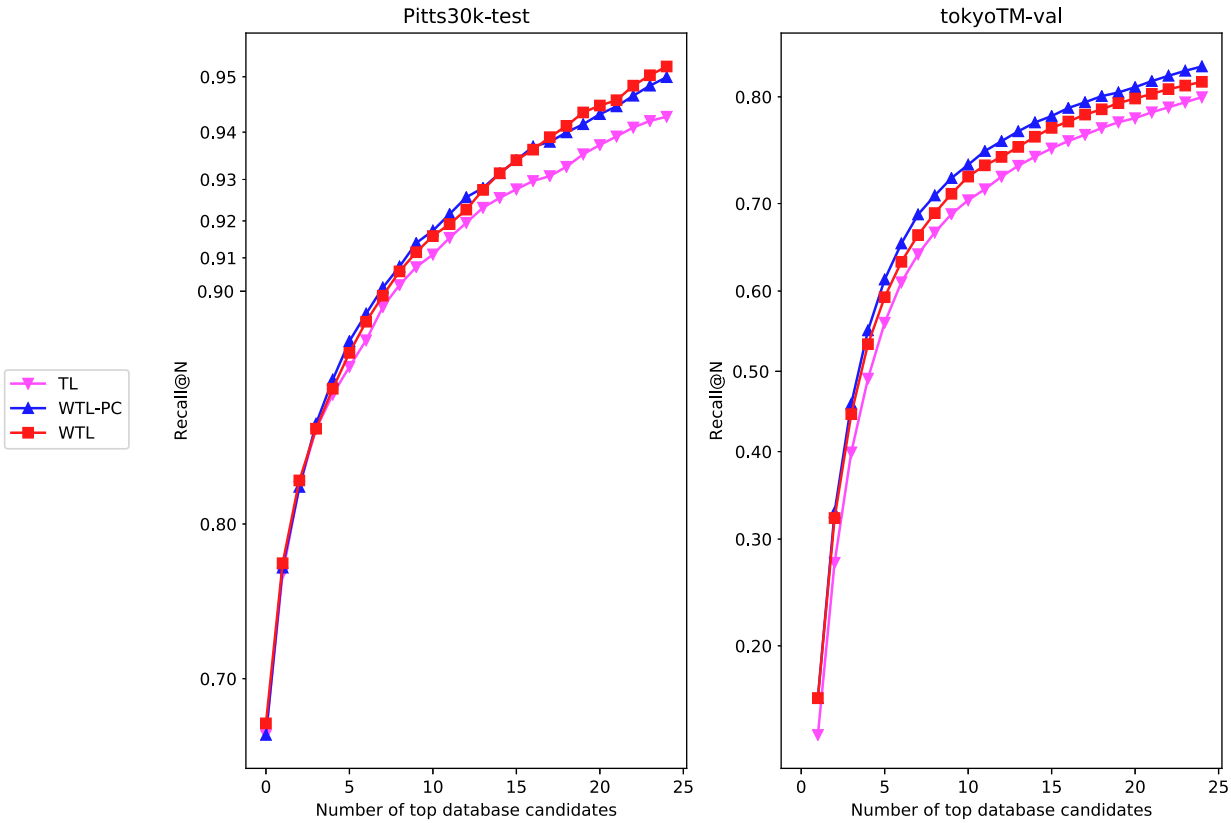


Fig. 4. Comparison of our WT-loss function versus T-loss function. The convolutional architecture is the VGG-16 base network. WT-loss-PC (imposing constraints only on one PS) (-▲-) and WT-loss (imposing constraints on both one positive and 10 negative samples for each query) (-■-) outperform T-loss on the Pitts30k-test data set by 0.502% and 0.238%, respectively; outperform T-loss on the TokyoTM-val data set by 6.068% and 4.342%, respectively. For simplicity, we denote WT-loss-PC as WTL-PC, WT-loss as WTL, and T-loss as TL in the legend.

TABLE III

RECOGNITION RESULTS OF THE BASE (VGG-16) NETWORK WITH WT-LOSS THAT USES DIFFERENT NUMBERS OF PSS FOR EACH QUERY ON THE PITTS30K-TEST DATA SET

Method	PS Number	recall@2	recall@3	recall@4	recall@5	recall@10	recall@15	recall@20	recall@25
base network	1	77.72	82.32	84.79	86.50	91.16	93.14	94.38	95.17
	3	60.08	65.72	68.98	71.83	78.56	82.16	84.43	86.11
	5	33.57	37.83	40.97	43.85	52.72	58.37	61.96	64.59

the smallest distances as the PSs while randomly selecting a number of images from the negative candidates as the negative samples. The distance can be computed as

$$d_{ij} = \begin{cases} \left(\left\|\mathbf{y}_i^{\text{image}} - \mathbf{y}_j^{\text{image}}\right\|^2\right) \cdot \theta_{\text{GPS}}^{p}\left(\left\|\mathbf{q}_i^{\text{GPS}} - \mathbf{q}_j^{\text{GPS}}\right\|^2\right) \\ \left(\left\|\mathbf{y}_i^{\text{image}} - \mathbf{y}_j^{\text{image}}\right\|^2\right) \cdot \theta_{\text{GPS}}^{n}\left(\left\|\mathbf{q}_i^{\text{GPS}} - \mathbf{q}_j^{\text{GPS}}\right\|^2\right) \end{cases} \tag{11}$$

where $\mathbf{y}^{\text{image}}$ is the feature representation of an image.

D. Discussion of Weighted Triplet Loss Layer

We first compare the proposed WT-loss layer against the traditional T-loss layer using the VGG-16 base network to verify the effectiveness. Fig. 4 demonstrates the recall performances derived from the features that are generated by WT-loss and T-loss. We analyze our observations from the following three aspects.

1) Different Setups About the Weighted Triplet Loss in the Number of Positive Samples: We first explore the recognition performances of different setups of the proposed WT-loss function. This is to give the readers better insight about the WT-loss layer. The difference between the WT-loss and traditional T-loss lies in the weights imposed on both the positive and negative samples of each query image. Therefore, the most probable factor that may influence the performance of the WT-loss function is the number of positive as well as negative samples. Since the huge amount of negative samples for each query image, it is quite difficult to define an optimal way, the negative samples are selected. Typically, the negative samples are randomly selected from those images taken from different places, therefore the appearance variation among the negative samples is very large. Based on this reason, we believe the number of the negative samples of each query image will also have a random impact on the recognition results. Therefore, we concentrate on the number of the PSs.

Table III includes the recognition results on Pitts30k-test data set. We demonstrate the recalls of the base network with WT-loss that uses different numbers of PSs for each query image. The number of negative samples is fixed at ten in all the implementations. It can be observed that we obtain

TABLE IV

PERFORMANCE OF THE BASE NETWORK, SA LAYER, MH-SA LAYER, NETVLAD LAYER, AND THE PROPOSED SPE-VLAD LAYER ON THE PITTSBURGH AND TOKYOTIMEMACHINE DATA SETS. ALL THE NETWORKS ARE TRAINED USING THE T-LOSS LAYER. WE SET THE ATTENDED LOCAL REGIONS TO 2 IN THE MH-SA LAYER, AND ALL LAYERS ARE IMPLEMENTED USING THE PYTORCH [43] DEEP LEARNING FRAMEWORK

Dataset	Loss	Method	recall@2	recall@3	recall@4	recall@5	recall@10	recall@15	recall@20	recall@25
Pitts30k-test	T-Loss	base	77.11	82.00	84.78	86.26	90.73	92.60	93.55	94.30
		SA	78.36	82.92	85.63	87.45	91.44	93.31	94.47	95.01
		MH-SA	77.33	82.30	85.20	87.34	91.60	**93.71**	**94.79**	95.50
		NetVLAD	**79.61**	**83.76**	**86.30**	**87.96**	91.71	93.60	94.74	95.44
		SPE-VLAD	79.29	83.65	86.04	87.77	**91.85**	93.58	94.60	**95.55**
TokyoTM-test	T-Loss	base	13.75	27.55	39.91	49.08	68.87	74.73	77.87	79.97
		SA	15.50	29.97	44.14	54.05	73.92	79.56	82.62	84.44
		MH-SA	16.14	31.46	45.06	54.15	73.02	78.90	81.95	83.89
		NetVLAD	18.83	36.45	50.14	58.98	76.21	81.29	**83.82**	**85.51**
		SPE-VLAD	**19.93**	**37.14**	**50.46**	**59.48**	**76.43**	**81.47**	83.77	85.47

the best results when only one PS is used. The recall at 5 even outperforms the recall at 5 s with three and five PSs by 14.67% and 42.65%, respectively. The reason is closely related to the intrinsic characteristic of the place recognition problem even the images taken at the sample place may have different visual appearances owing to the viewpoint variations. Involving multiple PSs for one query image may prevent the neural network from distinguishing the true similar image from other images with dissimilar appearances and, therefore, influence the convergence result of the neural network.

2) Weighted Triplet Loss Versus Weighted Triplet Loss With Only Positive Constraint: It is noteworthy that the weighting scheme described by (5) is flexible in that we can impose the weight constraints on either the positive/negative samples, or both the positive and negative samples to pursue the optimal performance.

In Fig. 4, the WT-loss with only positive constraints (WT-loss-PC) imposes weight constraints on only one PS exclusively during optimization, while the WT-loss imposes weight constraints on both one positive and ten negative samples. On the Pitts30k-test data set, the WT-loss-PC (recall at 5 86.866%) slightly outperforms the WT-loss (recall at 5 86.502%). On the TokyoTM-val data set, the WT-loss-PC reaches recall at 5 55.149%, while WT-loss reaches recall at 5 53.423%, and both are higher than the performance of the base network trained with the traditional T-loss layer.

When we imposed the weight constraints exclusively on the negative samples, we found that the recall performance is unstable and sometimes less impressive than that of the traditional T-loss. The most likely reason is that we defined the negative samples as some randomly selected images whose GPS data are beyond 25 m from the query image. In fact, an image captured far away from the place where the query image is captured could have a similar visual appearance to the query image. For example, photographs captured 10 mi away from the Eiffel Tower resembles those captured at a relatively close distance owing to the similar image layout and color of the buildings in the image. Hence, only imposing the weight constraints on the negative samples may deteriorate the performance, and we primarily verify the effectiveness of the WT-loss-PC layer in the following experiments.

3) WT-Loss and WT-Loss-PC Versus T-Loss: On the TokyoTM-val data set, the WT-loss and WT-loss-PC layer outperform the T-loss layer by 4.342% and 6.068% in the recall, respectively. Similar improvements can be observed on the Pitts30k-test data set. This demonstrates that our proposed WT-loss layer can guide the network to learn more discriminative features than the traditional T-loss layer. This is attributable to the proposed WT-loss function that exploits the distance constraints in temporal domain, aiming to monotonously decrease the distances between the query and its PSs and increase the distances between the query and its negative samples across the training epochs.

E. Comparison Between SPE-VLAD Layer and the State-of-the-Art Approaches

This section evaluates the performance of various network architectures with the SPE-VLAD layer and state-of-the-art approaches including the NetVLAD layer, SA layer, and multihead SA (MH-SA) layer. For the Pittsburgh and TokyoTimeMachine data sets, the networks are trained using the T-loss layer and WT-loss-PC (imposing constraints on one PS) layer. For Places365-Standard data set, the networks are trained using a softmax layer.

Table IV shows the performance of the aforementioned network architectures trained by the T-loss layer, where we observe that on the Pitts30k-test data set, NetVLAD and SPE-VLAD consistently outperform the SA layer and MH-SA layer in terms of recall at 2–10 by an average of 1%–2%, while the performance gain of recall at 2–10 is amplified to an average of 4%–8% on the TokyoTM-val data set. This is owing to the trait of the SA-based methods that focus only on one or several local regions that they believe are the most discriminative in the feature map. Therefore, the SA-based methods may not well reflect the holistic characteristics of the images. As a comparison, the VLAD-based methods consider the global statistical characteristics computed through the local descriptors of the feature maps; therefore, they perform better than the SA-based methods in the place recognition problem.

Among the VLAD-based methods, the proposed SPE-VLAD layer equips the feature representations with structural information by dividing the feature maps into patches and stacking the NetVLAD representations of the patches together. Theoretically, SPE-VLAD should be more effective than NetVLAD in feature learning. Table IV shows that the SPE-VLAD layer achieves comparable results on

TABLE V

PERFORMANCE OF THE BASE NETWORK, SA LAYER, MH-SA LAYER, NETVLAD LAYER, AND THE PROPOSED SPE-VLAD LAYER ON THE PITTSBURGH AND TOKYOTIMEMACHINE DATA SETS. ALL LAYERS ARE TRAINED USING THE PROPOSED WT-LOSS-PC

Dataset	Loss	Method	recall@2	recall@3	recall@4	recall@5	recall@10	recall@15	recall@20	recall@25
Pitts30k-test	WT-Loss-PC	base	77.46	81.99	85.01	86.87	91.41	93.14	94.15	94.99
		SA	79.36	83.89	86.69	88.53	92.16	**94.00**	**94.95**	95.66
		MH-SA	78.66	83.52	86.20	87.69	91.54	93.44	**94.95**	**95.69**
		NetVLAD	79.21	83.01	84.89	86.53	90.58	92.30	93.37	94.30
		SPE-VLAD	**81.25**	**85.52**	**87.77**	**89.23**	**92.50**	93.82	94.62	95.33
TokyoTM-test	WT-Loss-PC	base	16.10	32.93	45.91	55.15	72.63	77.85	80.35	82.34
		SA	16.78	33.55	46.21	55.89	74.53	79.70	82.41	84.12
		MH-SA	17.02	34.23	48.18	56.62	75.06	80.11	82.90	84.62
		NetVLAD	**24.35**	**42.97**	54.48	61.58	75.13	79.72	82.00	83.75
		SPE-VLAD	23.20	41.98	**55.15**	**63.90**	**77.28**	**81.54**	**83.97**	**85.61**

TABLE VI

PERFORMANCE OF THE BASE NETWORK, SA LAYER (SA), MH-SA LAYER, NETVLAD LAYER, AND THE PROPOSED SPE-VLAD LAYER ON THE PLACES365-STANDARD DATA SET. ALL LAYERS ARE TRAINED USING A SOFTMAX LOSS

Dataset	Loss	Method	top-1(%)	top-2(%)	top-3(%)	top-4(%)	top-5(%)
Places365-Standard-test	softmax	base	60.667	71.033	74.533	76.6	77.9
		SA	80.614	91.549	**95.55**	96.585	97.654
		MH-SA	61.6	72.233	75.633	77.267	78.533
		NetVLAD	79.855	91.549	95.24	96.792	**97.861**
		SPE-VLAD	**81.131**	**91.687**	95.481	**97.102**	97.827

the Pitts30k-data set with the NetVLAD layer. The recall at 2–5 generated by the SPE-VLAD layer is slightly lower than that by NetVLAD. Theoretically, this might be caused by the high dimension of the SPE-VLAD feature representation. Due to the introduced spatial pyramid, the feature dimension of the SPE-VLAD is typically several to dozens of times higher than that of NetVLAD. When the dimension surpasses the number of samples, the discriminative power of the learned features may degenerate. One way to tackle this problem is using more data, and the other approach is exploiting channel reduction through fully connected layers or 1×1 kernel size convolutional layer with less channels. The recall at 10 and recall at 25 are better than that by NetVLAD. On the TokyoTM-val data set, the SPE-VLAD surpasses the NetVLAD approach (state-of-the-art method). For example, the recall at 2 of NetVLAD is 18.83%, whereas the recall at 2 of SPE-VLAD reaches 19.93% with 1% performance gain, and the recall at 3–15 of SPE-VLAD is still higher than that by NetVLAD. This agrees with our expectation that the proposed SPE-VLAD layer performs better on a larger data set.

In summary, compared with the locality-intensive models such as the SA layer and MH-SA layer, the proposed SPE-VLAD layer successfully preserves the global structural information of an image, such as the layout of the scene and the contour of the building and, therefore, is more robust in feature learning. Interestingly, the SPE-VLAD layer incurs only slightly extra computational overhead to the deep learning framework.

Table V shows the performance of these network architectures trained by the WT-loss-PC layer. SPE-VLAD outperforms all other networks (base network, SA, MH-SA, NetVLAD) in terms of recall at 2–10 on the Pitts30k-test data set. Specifically, the proposed SPE-VLAD layer achieves better results compared with the NetVLAD layer in terms of recall at 2–25. On the TokyoTM-val data set, NetVLAD and SPE-VLAD outperform SA and MH-SA in terms of recall at 2–5 by at least approximately 6%. SPE-VLAD achieves comparable results with NetVLAD in terms of recall at 2 and recall at 3, but surpasses NetVLAD in terms of recall at 4–25, among which recall at 5 is of the most reference significance in evaluating the image retrieval methods. Therefore, the overall retrieval quality of SPE-VLAD is better than that of NetVLAD.

Comparing Table V with Table IV, we found that all the deep architectures (the base network, SA, MH-SA, NetVLAD, and SPE-VLAD) trained on the TokyoTM-val data set by the WT-loss-PC layer surpassed their counterparts trained by the T-loss layer in terms of all the experimental results except for several recall data, such as recall at 10 of NetVLAD and recall at 25 of NetVLAD. For the most important indicator recall at 5, the SA, MH-SA, NetVLAD, and SPE-VLAD attained the performance gain by 1.84%, 2.47%, 2.60%, and 4.42%, respectively, compared with their counterparts trained by the T-loss layer. On the Pitts30k-test data set, the aforementioned deep architectures trained by the WT-loss-PC layer yielded similar results as the architectures trained by the T-loss layer. However, the WT-loss-PC layer produced a higher recall at 5 than the T-loss layer in all architectures except for NetVLAD, and the performance gain of the base network, SA, MH-SA, and SPE-VLAD are 0.61%, 1.08%, 0.35%, and 1.46%, respectively. The advantage of using the WT-loss layer against the T-loss layer is that if the distances between a query image and its PSs are found to increase, the weight constraints subsequently take effect to decrease the distances in the next epoch (the temporal constraints can also be imposed to negative samples). Hence, the WT-loss layer generally ensures that the network converges to a better solution than the T-loss layer.

Table VI demonstrates the recognition rates on the Places365-Standard data set. In the implementation, the base network is a ResNet-18 network and the loss layer

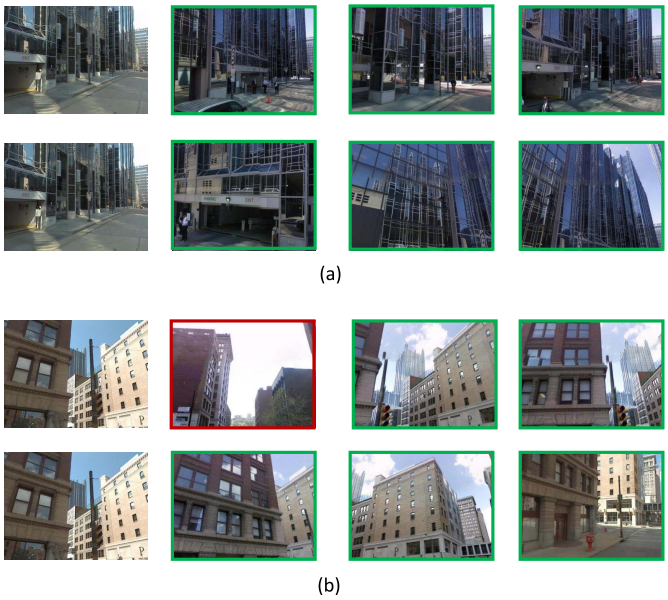


Fig. 5. Image retrieval results on Pitts30k-test data set. (a) Retrieval comparison between T-loss and the proposed WT-loss layer. Results of NetVLAD trained by T-loss layer (top row). Results of NetVLAD trained by WT-loss-PC layer (bottom row). (b) Retrieval comparison between NetVLAD and the proposed SPE-VLAD layer. Results of NetVLAD trained by our WT-loss-PC layer (top row). Results of the proposed SPE-VLAD trained by our WT-loss-PC layer (bottom row). In both (a) and (b), the leftmost column denotes the query images, and we show the top three retrieved images on the three right columns. Images that represent the same place as the query image are enclosed with green frames, whereas images that represent different places are enclosed with red frames.

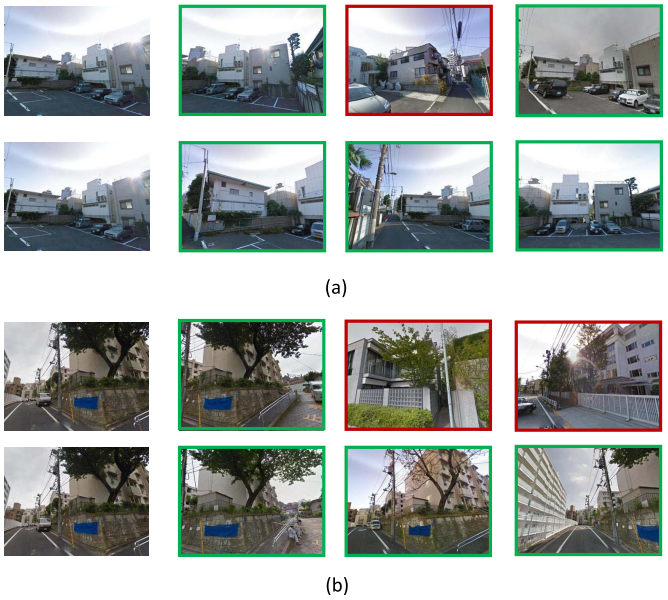


Fig. 6. Image retrieval results on the TokyoTM-val data set. The image layout is the same as in Fig. 5. (a) Retrieval comparison between T-loss and the proposed WT-loss layer. Results of NetVLAD trained by T-loss layer (top row). Results of NetVLAD trained by WT-loss-PC layer (bottom row). (b) Retrieval comparison between NetVLAD and the proposed SPE-VLAD layer. Results of NetVLAD trained by our WT-loss-PC layer (top row). Results of the proposed SPE-VLAD trained by our WT-loss-PC layer (bottom row). In both (a) and (b), the leftmost column denotes the query images, and we show the top three retrieved images on the three right columns. Images that represent the same place as the query image are enclosed with green frames, whereas images that represent different places are enclosed with red frames.

adopts a softmax loss function. As expected, the proposed SPE-VLAD outperforms other methods on top-1, top-2, and top-4, among which the top-1 is the most important evaluation metric because it indicates the probability that the ground-truth label is just the most probable label given by the classifier. The reason also lies in the fact that the SPE-VLAD layer encodes the spatial structural information of the image into the feature representations. The results of MH-SA are even worse than that of SA, the reason may be related to the fact that the images in Places365-standard data set have much more complicated image layout and randomly emerged background objects. When MH-SA method is seeking more useful local regions, it may unfortunately focus on those background objects, which may degenerate the discriminative power of the feature representations.

F. Qualitative Evaluation

To provide the readers with intuitive insights regarding the proposed framework, we show the retrieved images in Figs. 5 and 6 using the SPE-VLAD feature extraction layer and the WT-loss-PC (imposing constraints on one PS) layer. The query images are arbitrarily chosen from the Pitts30k-test data set and the TokyoTM-val data set.

We separately demonstrate the comparison of the SPE-VLAD layer with the NetVLAD layer, and the comparison of the WT-loss-PC layer with the T-loss layer in Fig. 5, where Fig. 5(a) represents the retrieval comparison between T-loss and the proposed WT-loss-PC layer, in which the top row stands for the results of NetVLAD trained by the T-loss layer, the bottom row is the results of NetVLAD trained by the WT-loss-PC layer, Fig. 5(b) shows the retrieval comparison between NetVLAD and the proposed SPE-VLAD layer, the top row shows the results of NetVLAD trained by our WT-loss-PC layer, and the bottom row shows the results of the proposed SPE-VLAD trained by our WT-loss-PC layer. In both Fig. 5(a) and (b), the leftmost column denotes the query images, and the remaining images on the three right columns are the retrieved ones. Images that represent the same place as the query image are enclosed with green frames, whereas images that represent different places are enclosed with red frames. Fig. 6 shows the same layout.

We can observe from Figs. 5(a) and 6(a) that the top three images retrieved by the WT-loss-PC layer are captured at exactly the same place as the query image, whereas the top three images retrieved by the T-loss layer may contain the wrong images captured somewhere else. In Fig. 5(a), although the T-loss layer also obtains the correct images, the retrieved images have approximately the same viewpoint as the query image. As a comparison, the retrieved images via the WT-loss-PC layer contain a large viewpoint variation. This indicates that the learned features via the WT-loss-PC layer are not sensitive to the viewpoint variation, thus rendering the WT-loss-PC layer more suitable for content-based image retrieval. We conjecture that the viewpoint invariance may stem from the optimized network parameters learned by the WT-loss-PC layer. Furthermore, Figs. 5(b) and 6(b) show that

the top three images retrieved using the SPE-VLAD feature representation are still better than the images retrieved using the NetVLAD feature representation.

V. Conclusion

We proposed a novel DCNN architecture that is trainable in an end-to-end manner for place recognition. The key components of this architecture include the following: 1) SPE-VLAD layer and 2) WT-loss layer. The SPE-VLAD layer encodes not only the global statistical characteristics of an image but also the spatial and structural information contained in the image into the learned feature representations, such that the feature representations possess more discriminant power and are more robust to the place recognition problem. The WT-loss layer imposes temporal constraints on the positive and negative samples with respect to each query image, to monotonously decrease the distances between the query image and the PSs while increasing the distances between the query image and the negative samples through training epochs. The experimental results on two widely used benchmark data sets for place recognition revealed that combining the SPE-VLAD layer and WT-loss layer yielded the state-of-the-art performance in place recognition, compared with the several recently proposed deep learning methods. Theoretically, the proposed framework can be generalized and applied to other real-world image retrieval tasks except place recognition.

References

[1] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 1, Oct. 2003, pp. 273–280.

[2] S. Lowry *et al.*, "Visual place recognition: A survey," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[3] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 11, pp. 2660–2673, Nov. 2017.

[4] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[5] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for BigDog," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4736–4741.

[6] E. Chalmers, E. B. Contreras, B. Robertson, A. Luczak, and A. Gruber, "Learning to predict consequences as a method of knowledge transfer in reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2259–2270, Jun. 2018.

[7] L. Zhu, Z. Huang, Z. Li, L. Xie, and H. Shen, "Exploring auxiliary context: Discrete semantic transfer hashing for scalable image retrieval," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5264–5276, Nov. 2018.

[8] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 257–271, Feb. 2018.

[9] D. M. Chen *et al.*, "City-scale landmark identification on mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 737–744.

[10] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–7.

[11] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys, "Hyperpoints and fine vocabularies for large-scale location recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2102–2110.

[12] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[13] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla, "Visual place recognition with repetitive structures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 883–890.

[14] R. Arandjelovi and A. Zisserman, *DisLocation: Scalable Descriptor Distinctiveness for Location Recognition*. Berlin, Germany: Springer, 2014.

[15] H. Wang *et al.*, "Identifying objective and subjective words via topic modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 718–730, Mar. 2018.

[16] H. Jégou, M. Douze, C. Schmid, and P. Prez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.

[17] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Computer Vision* (Lecture Notes of Computer Science), vol. 5302. Berlin, Germany: Springer, 2008.

[18] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[19] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys, "Large-scale location recognition and the geometric burstiness problem," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1582–1590.

[20] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5153–5161.

[21] J. Knopp, J. Sivic, and T. Pajdla, "Avoiding confusing features in place recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 748–761.

[22] S. Cao and N. Snavely, "Graph-based discriminative learning for location recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 700–707.

[23] P. Gronát, G. Obozinski, J. Sivic, and T. Pajdla, "Learning and calibrating per-location classifiers for visual place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 907–914.

[24] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[25] C. O. Sakar and O. Kursun, "Discriminative feature extraction by a neural implementation of canonical correlation analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 164–176, Jan. 2017.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[27] W. Luo, J. Li, J. Yang, W. Xu, and J. Zhang, "Convolutional sparse autoencoders for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3289–3294, Jul. 2018.

[28] I. Rocco, R. Arandjelović, and J. Sivic. (2017). "Convolutional neural network architecture for geometric matching." pp. 1–15. [Online]. Available: https://arxiv.org/abs/1703.05593

[29] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1437–1451, Jun. 2018.

[30] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 3476–3485.

[31] L. Chen *et al.*, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 6298–6306.

[32] Q. Li, Q. Peng, and C. Yan, "Multiple VLAD encoding of CNNs for image classification," *Comput. Sci. Eng.*, vol. 20, no. 2, pp. 52–63, Mar./Apr. 2018.

[33] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–154, Nov. 1992.

[34] Y. Shen, R. Ji, C. Wang, X. Li, and X. Li, "Weakly supervised object detection via object-specific pixel gradient," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 5960–5970, Dec. 2018.

[35] Q. Tao, G. Wu, and D. Chu, "Improving sparsity and scalability in regularized nonconvex truncated-loss learning problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2782–2793, Jul. 2018.

[36] L. E. B. da Silva and D. C. Wunsch, "An information-theoretic-cluster visualization for self-organizing maps," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2595–2613, Jun. 2017.

[37] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[39] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.
[40] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[41] Z.-Y. Zhang and H.-Y. Zha, *Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment*. Philadelphia, PA, USA: SIAM, 2005.
[42] S. Ying, Z. Wen, J. Shi, Y. Peng, J. Peng, and H. Qiao, "Manifold preserving: An intrinsic approach for semisupervised distance metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2731–2742, Jul. 2018.
[43] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017.

Jun Yu (M'13) received the B.Eng. and Ph.D. degrees from Zhejiang University, Hangzhou, China.

He was an Associate Professor with the School of Information Science and Technology, Xiamen University, Xiamen, China. From 2009 to 2011, he was with Nanyang Technological University, Singapore. From 2012 to 2013, he was a Visiting Researcher with Microsoft Research Asia (MSRA), Beijing, China. He is currently a Professor with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou. He has authored or coauthored more than 60 scientific articles. His current research interests includes multimedia analysis, machine learning, and image processing.

Dr. Yu is a Professional Member of the Association for Computing Machinery and the China Computer Federation. He was a recipient of the IEEE SPS Best Paper Award in 2017. He was the Co-Chair of several special sessions, invited sessions, and workshops. He served as a Program Committee Member or a reviewer for top conferences and prestigious journals.

Chaoyang Zhu is currently pursuing the bachelor's degree with the Honorary School, Hangzhou Dianzi University, Hangzhou, China.

His current research interests include image retrieval, computer vision, and deep learning.

Jian Zhang received the Ph.D. degree from Zhejiang University, Hangzhou, China.

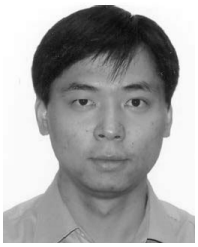
From 2009 to 2011, he was a Post-Doctoral Research Fellow with the Department of Mathematics, Zhejiang University. He was a Visiting Scholar with Simon Fraser University (SFU), Burnaby, BC, Canada, in 2016, where he was involved in machine learning. He is currently an Associate Professor with the School of Science and Technology, Zhejiang International Studies University, Hangzhou. His current research interests include machine learning, computer animation, and image processing.

Dr. Zhang is a reviewer of several prestigious journals in his research domain.

Qingming Huang (F'15) received the bachelor's degree in computer science and the Ph.D. degree in computer engineering from the Harbin Institute of Technology, Harbin, China, in 1988 and 1994, respectively.

He is currently a Professor with the University of Chinese Academy of Sciences, Beijing, China, and an Adjunct Research Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. He has authored or coauthored more than 400 academic papers in prestigious international journals and top-level international conferences. His current research interests include multimedia computing, image processing, computer vision, and pattern recognition.

Dr. Huang has served as the General Chair, the Program Chair, the Track Chair, and a TPC Member for various conferences, including ACM Multimedia, CVPR, ICCV, ICME, ICMR, PCM, BigMM, PSIVT, and so on. He is the Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (CSVT) and *Acta Automatica Sinica*. He is a reviewer for various international journals including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, and so on.



Dacheng Tao (F'15) is currently a Professor of computer science and an ARC Laureate Fellow with the School of Computer Science, Faculty of Engineering and Information Technologies and the Inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney, Darlington, NSW, Australia. He mainly applies statistics and mathematics to artificial intelligence and data science. His research results have expounded in 1 monograph and more than 200 publications at prestigious journals and prominent conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (IEEE T-PAMI), IEEE TRANSACTIONS ON IMAGE PROCESSING (T-IP), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (T-NNLS), IEEE TRANSACTIONS ON CYBERNETICS, *International Journal of Computer Vision* (T-CYB), *Journal of Machine Learning Research* (JMLR), *Neural Information Processing Systems* (NIPS), *International Conference on Machine Learning* (ICML), *Computer Vision and Pattern Recognition* (CVPR), ICCV, ECCV, ICDM, and ACM SIGKDD.

Mr. Tao is a fellow of the Australian Academy of Science, AAAS, IAPR, OSA, and SPIE. He was a recipient of several best paper awards such as the Best Theory/Algorithm Paper Runner Up Award in IEEE ICDM07, the Best Student Paper Award in IEEE ICDM13, the 2014 ICDM 10-Year Highest-Impact Paper Award, the 2017 IEEE Signal Processing Society Best Paper Award, the Distinguished Paper Award in the 2018 IJCAI, the 2015 Australian Scopus-Eureka Prize, and the 2018 IEEE ICDM Research Contributions Award.