Attentional Pyramid Pooling of Salient Visual Residuals for Place Recognition

Guohao Peng[1], Jun Zhang[1], Heshan Li[2], Danwei Wang[2]
Nanyang Technological University, Singapore
[1]{peng0086,jzhang061}@e.ntu.edu.sg, [2]{heshan.li,edwwang}@ntu.edu.sg

Abstract

The core of visual place recognition (VPR) lies in how to identify task-relevant visual cues and embed them into discriminative representations. Focusing on these two points, we propose a novel encoding strategy named Attentional Pyramid Pooling of Salient Visual Residuals (APPSVR). It incorporates three types of attention modules to model the saliency of local features in individual, spatial and cluster dimensions respectively. (1) To inhibit task-irrelevant local features, a semantic-reinforced local weighting scheme is employed for local feature refinement; (2) To leverage the spatial context, an attentional pyramid structure is constructed to adaptively encode regional features according to their relative spatial saliency; (3) To distinguish the different importance of visual clusters to the task, a parametric normalization is proposed to adjust their contribution to image descriptor generation. Experiments demonstrate APPSVR outperforms the existing techniques and achieves a new state-of-the-art performance on VPR benchmark datasets. The visualization shows the saliency map learned in a weakly supervised manner is largely consistent with human cognition.

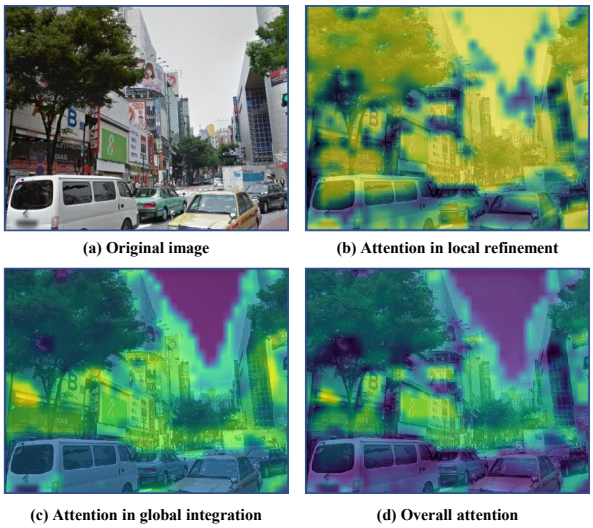


Figure 1. APPSVR consists of two main steps: local refinement and global integration. Refining the prior knowledge of "preserving the semantics of buildings", local refinement (b) can adaptively highlight billboards and inhibit repeated structures on buildings. Certain visual cues improperly retained in (b) (*e.g.*, vehicle parts that look like architectural windows) can be suppressed in the subsequent global integration (c). Finally, the overall attention (d) of APPSVR is consistent with human perception habit of valuing static structures and omitting misleading visual elements.

1. Introduction

Visual place recognition (VPR) has become the core technique of many promising applications in the field of computer vision [1,2,5,39,41,45] and robotics [7,10,11,25], such as autonomous driving [6,24,28], geo-localization [22, 23, 40], 3D reconstruction [9] and virtual reality [26].

VPR in large-scale environments is typically solved as an instance retrieval task [1, 2, 18, 19, 40, 41, 45], where the goal is to find the most visually similar database images for a given query image. The main challenge is different viewpoints, weather and illumination may cause dramatic changes in the appearance of the same scene. Partial occlusion and dynamic objects also bring additional challenges to the task. Therefore, how to construct powerful image representations has raised widespread concerns in the field.

In the exploratory work carried out over the past decades, VLAD [3] and its variants [1, 19, 48] stand out from other counterparts by introducing residual that can better characterize the nuances of local details. Drawing on their wisdom, we follow the basic idea of aggregating cluster-wise residuals for feature embedding. Considering that not every visual element in the image is helpful to the VPR task, it is necessary to emphasize the task-relevant ones in the image representation.

With similar motivations, early attempts [2, 20, 39] have been made to accentuate task-relevant local features. Following the pace of deep learning, the recent attention-aware methods for VPR can be broadly divided into two categories. The data-driven methods [19, 31, 51] usually integrate trainable attention modules into the encoding network. Through end-to-end learning, these modules essen-

tially act as a black box weighting of local features. The rule-based methods [27, 29, 35] typically use artificial rules to filter specific visual cues for subsequent encoding. Their performance is susceptible to the bias of rough prior knowledge. To combine the advantages of both categories, we adopt a semantic reinforced attention module [32] for local feature refinement, where semantic priors can be reflected by the initial weights of the parametric model. With further fine-tuning, the model can learn comprehensive reasoning habits from prior knowledge and data-driven training.

Besides the individual distinctiveness, local features' task relevance also greatly depends on their context in the scene. Some existing methods, such as contextual reweighting [19] and multi-scale regional pooling [44, 48, 51], have demonstrated the advantages of incorporating spatial information into the encoding strategy. Inspired by them, we develop an attentional pyramid pooling to leverage the features' regional context. Specifically, an overlapping pyramid structure is constructed, where regional features are formulated by aggregating salient visual residuals within each grid. Then through a spatial attention module, regional features are weighted by their relative spatial saliency before being embedded into the visual word vector. In particular, our spatial saliency weight is derived based on the global context, not just based on the regions in the rigid grid.

Moreover, the final representation of VLAD variants [1, 17,34] is normally the concatenation of visual word vectors, whose scales are equalized by intra-normalization [17]. Consequently, all visual words contribute the same to the descriptor generation and similarity metric. To distinguish their different importance to the task, we propose a parametric normalization, through which visual word vectors are rescaled according to their task relevance and then concatenated as a unit image descriptor. In this way, the different saliency of visual clusters can be intuitively highlighted in the similarity interpretation during indexing.

In summary, we propose an attentional encoding strategy for VPR, named Attentional Pyramid Pooling of Salient Visual Residuals (APPSVR). Particularly, we introduce three types of attention modules to model the saliency of local features in individual, spatial and cluster dimensions respectively. Incorporating the triple attention, our model can adaptively identify and embed salient visual cues into discriminative image descriptor. Experiments verify the effectiveness of all proposed modules, and demonstrate that our model significantly outperforms baseline methods on city-scale benchmark datasets. In summary, our contributions are as follows:

- We propose an encoding strategy APPSVR for VPR, which integrates the triple attention from individual, spatial and cluster saliency into feature embedding.
- For local feature refinement, we adopt a semantic-reinforced local weighting scheme, where crompre-hensive local saliency can be learned from both prior knowledge and data-driven fine-tuning.
- We propose an attentional pyramid pooling and a parametric normalization for global integration, through which spatial and cluster saliency can be incorporated into the encoding strategy.
- Experiments demonstrate that APPSVR outperforms existing methods and achieves a new state-of-the-art performance on benchmark datasets. The visualization shows the attention learned under weak supervision is highly consistent with human cognition.

2. Related Work

Traditionally, the large-scale VPR is framed as an image retrieval task, where the key is to find a discriminative representation for accurate and fast indexing. In this section, we will briefly review the literature on the global image representation for VPR that is related to our method.

Early methods employ variants of Bag of Words [39, 41, 45], Vector of Locally Aggregated Descriptors (VLAD) [3] or Fisher Vectors (FV) [33, 34] to encode hand-crafted local features [4, 12]. With the rise of deep learning, the community has gradually shifted to exploiting the intermediate activations of pretrained CNNs as deep local features. Replacing hand-crafted local features with deep learned ones can bring stable performance improvements for aggregation methods, such as sum pooling [3], max pooling [38], VLAD [30] and FV [47]. Additionally, combining the spatial information of multi-scale patches [18, 43, 48] into feature embedding has also proven to lead to better image representation. Recent studies [1, 13, 36] explore the deep learning based architectures that can be used for task-specific training. Arandjelovic *et al*. [1] introduce the seminal VLAD pooling layer named NetVLAD. Yu *et al*. [48] propose SPENetVLAD, which encodes spatial information by stacking regional features. Gordo *et al*. [13] retrain the R-MAC [44] on a landmark dataset and obtain an outstanding improvement. Despite the considerable advantages, the aforementioned methods allow all local features to participate in feature embedding, where misleading visual cues are also encoded into the image representation.

In order to reduce the influence of task-irrelevant local features, researchers put forward the consolidation of attention blocks with CNN architectures. Kim *et al*. [19] enhance the NetVLAD [1] through a contextual reweighting network (CRN). Zhu *et al*. [51] propose APAnet that aggregates multi-scale regional features weighted by cascaded attention blocks. Concurrently, some approaches use artificial prior knowledge for semantic-guided feature filtering [27, 29, 35]. They retain the local features with specified semantics for subsequent embedding. While the previous attention strategies are either completely data-driven

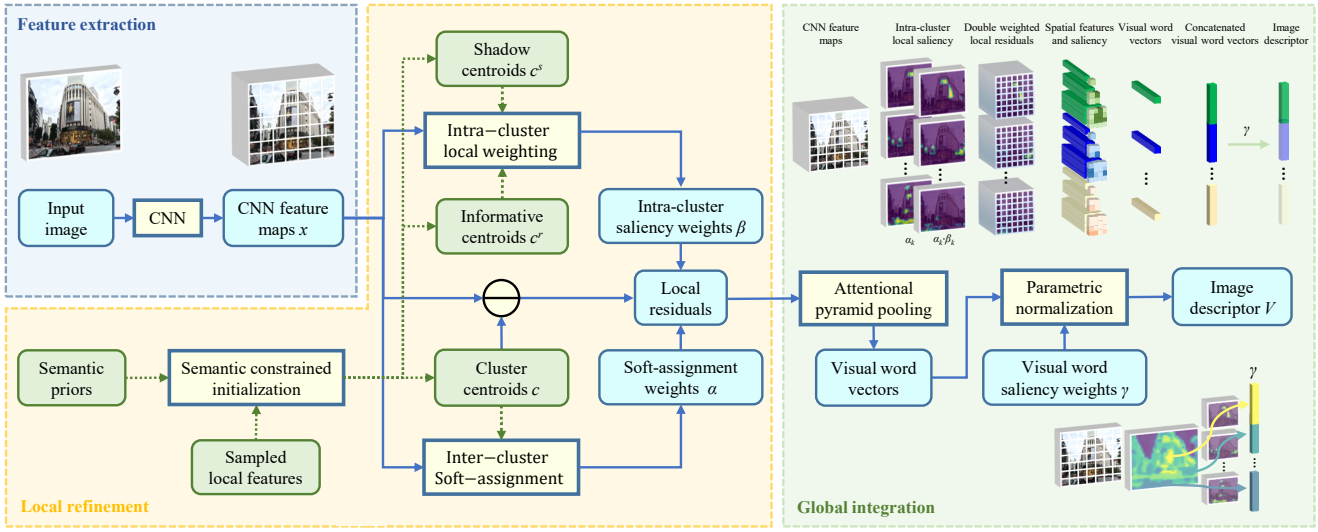


Figure 2. The overall flowchart of APPSVR. For local refinement, a hierarchical weighting scheme reinforced by semantic priors is employed to suppress the misleading local features within each cluster. For global integration, spatial saliency is first leveraged to highlight the salient regional features in visual word vector encoding. Then cluster saliency is incorporated to adjust the contribution of visual word vectors to the final image representation.

or based on artificial rules, Peng *et al*. [32] propose SRALNet that introduces semantic priors to enhance the differential local weighting scheme. However, it overlooks the contextual information of local features in the process of feature weighting. The role of intra-cluster local saliency on the final image representation is also restricted by intra-normalization [17]. On top of SRALNet, through attentional pyramid pooling, APPSVR utilizes the multi-scale spatial information for more discriminative feature embedding. Additionally, APPSVR provides data-driven weighting for clusters, which adjusts their contribution to both image descriptor generation and similarity metric.

3. Preliminaries

Before introducing the proposed method, we first provide an overview of the baseline NetVLAD [1].

Deep local features. A cropped convolutional neuron network (*e.g*., VGG-16 [42] or AlexNet [21]) is first exploited as the base network. The spatial activations from the output feature maps $M \in R^{D\times H\times W}$ are normalized and regarded as deep local features $\boldsymbol{x} \in R^{D\times 1\times 1}$.

Soft-assignment. The deep local features are then divided into K visual word clusters through soft-assignment. As in Eq.(1), the soft-assignment weight $\alpha_k(\boldsymbol{x_i})$ of a local feature $\boldsymbol{x}_i$ being allocated to the k^{th} cluster is related to its proximities to the cluster centroids $\{\boldsymbol{c}_k\}_{k=1}^K$. The constant a is selected to be a large positive number, which controls the decay of the response with the magnitude of the distance.

$$\alpha_k(\boldsymbol{x}_i) = \frac{e^{-a\|\boldsymbol{x}_i-\boldsymbol{c}_k\|^2}}{\sum_{j=1}^K e^{-a\|\boldsymbol{x}_i-\boldsymbol{c}_j\|^2}} = \frac{e^{\boldsymbol{w}_{k0}^{\mathrm{T}}\boldsymbol{x}_i+b_{k0}}}{\sum_{j=1}^K e^{\boldsymbol{w}_{k0}^{\mathrm{T}}\boldsymbol{x}_i+b_{k0}}} \quad (1)$$

Descriptor generation. Through Eq.(2), each visual word vector V_k is formulated by indiscriminately aggregating the local residuals belonging to the cluster.

$$\boldsymbol{V}_k = \sum_{i=1}^{HW} \alpha_k(\boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{c}_k^r) \quad (2)$$

Then as in Eq.(3), the descriptor $\boldsymbol{V}(\boldsymbol{X})$ of an image $\boldsymbol{X}$ is obtained by stacking visual word vectors $\{\boldsymbol{V}_k(\boldsymbol{X})\}_{k=1}^K$ and performing intra-normalization [17] and L_2-normalization. Let $\tilde{\boldsymbol{V}}_k(\boldsymbol{X})$ denote a normalized visual word vector. The L_2-normalization actually conducts an element-wise multiplication by $\frac{1}{\sqrt{K}}$ for all subvectors, so that $\|\boldsymbol{V}(\boldsymbol{X})\|$=1.

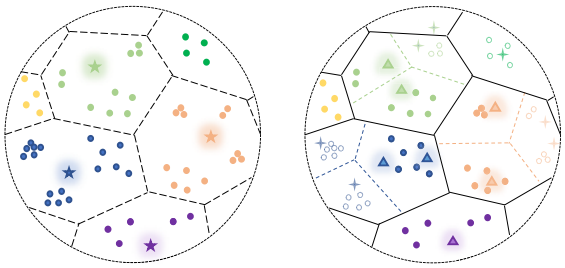
$$\boldsymbol{V}(\boldsymbol{X}) = \left[\frac{\tilde{\boldsymbol{V}}_1(\boldsymbol{X})}{\sqrt{K}}, \frac{\tilde{\boldsymbol{V}}_2(\boldsymbol{X})}{\sqrt{K}}, ..., \frac{\tilde{\boldsymbol{V}}_K(\boldsymbol{X})}{\sqrt{K}}\right] \quad (3)$$

4. Proposed Method

In this section, we describe the proposed method in detail. Fig.2 provides an overview of our encoding strategy, which can be summarized into two main steps: local refinement and global integration. Each functional step will be explained sequentially in the following subsections.

4.1. Local refinement

Since not all local features describe the task-relevant visual cues in an image, indiscriminatively encoding all of them as in Eq.(2) will result in misleading features degrading the image representation. Therefore, we adopt semantic reinforced intra-cluster weighting [32] for local feature refinement, which uses individual distinctiveness to suppress misleading features with semantics unrelated to the task.



(a) Inter-cluster soft-assignment (b) Intra-cluster local weighting

Figure 3. Illustration of the hierarchical weighting for local feature refinement. Deep local features are first divided into different clusters through soft-assignment. Then intra-cluster local weighting is conducted to suppress misleading features in each cluster.

Intra-cluster local weighting. Following the common steps described in Section 3, we first extract deep local features $\{\boldsymbol{x}_i\}$ from an image and divide them into K visual word clusters through soft-assignment $\alpha_k(\boldsymbol{x}_i)$ as in Eq.(1).

As visual cues describing similar appearance and semantics are usually mapped closer in the feature space and have consistent task relevance, we regard the Voronoi cell of each feature cluster as a combination of multiple information areas and ambiguous areas. Each informative area I_n is represented by an informative centroid $\boldsymbol{c}^r_{kn}$ ($\triangle$ in Fig.3.b), while each ambiguous area S_l is represented by a shadow centroid $\boldsymbol{c}^s_{kl}$ ($+$ in Fig.3.b). The intra-cluster saliency weight $\beta_k(\boldsymbol{x}_i)$ of a local feature $\boldsymbol{x}_i$ is formulated as its probability of being located in the informative areas of the k^{th} cluster. Formally, as in Eq.(4), $\beta_k(\boldsymbol{x}_i)$ is related to the proximities of $\boldsymbol{x}_i$ to the intra-cluster centroids $\{\boldsymbol{c}^r_{kn}\}_{n=1}^N$ and $\{\boldsymbol{c}^s_{kl}\}_{l=1}^L$.

$$\beta_k(\boldsymbol{x}_i) = \frac{\sum_{n=1}^{N} e^{-b\|\boldsymbol{x}_i - \boldsymbol{c}^r_{kn}\|^2}}{\sum_{l=1}^{L} e^{-b\|\boldsymbol{x}_i - \boldsymbol{c}^s_{kl}\|^2} + \sum_{n=1}^{N} e^{-b\|\boldsymbol{x}_i - \boldsymbol{c}^r_{kn}\|^2}} \tag{4}$$

Intuitively, local features located in ambiguous areas have the smallest distance to their closest shadow centroid. The denominator of their $\beta_k(\boldsymbol{x}_i)$ will be much larger than the numerator, due to the large positive constant b that controls the response with the magnitude of the distance. As a result, they will be assigned a low saliency weight.

By reducing $e^{-b\|\boldsymbol{x}_i\|^2}$ and using the abbreviated symbol $\{\boldsymbol{c}_{kj}\}_{j=1}^{N+L}$ to represent $\{\boldsymbol{c}^r_{kn}\}_{n=1}^N$ and $\{\boldsymbol{c}^s_{kl}\}_{l=1}^L$, Eq.(4) can be further derived as Eq.(5) and implemented by a trainable convolutional layer followed by summing Softmax logits of the specified channels.

$$\beta_k(\boldsymbol{x}_i) = \frac{\sum_{n=1}^{N} e^{2b\boldsymbol{c}_{kn}^{\mathrm{T}}\boldsymbol{x}_i - b\|\boldsymbol{c}_{kn}\|^2}}{\sum_{j=1}^{N+L} e^{2b\boldsymbol{c}_{kj}^{\mathrm{T}}\boldsymbol{x}_i - b\|\boldsymbol{c}_{kj}\|^2}} = \frac{\sum_{n=1}^{N} e^{\boldsymbol{w}_{kn}^{\mathrm{T}}\boldsymbol{x}_i + b_{kn}}}{\sum_{j=1}^{N+L} e^{\boldsymbol{w}_{kj}^{\mathrm{T}}\boldsymbol{x}_i + b_{kj}}} \tag{5}$$

Semantic constrained initialization. As in Fig.(3), the hierarchical weighting in Eq.(1) and Eq.(5) can be interpreted as allocating features into clusters and intra-cluster sub-areas. This signifies that prior attention can be provided by an initial feature space partition, which is actually determined by the centroids $\{\boldsymbol{c}_k\}$, $\{\boldsymbol{c}^r_{kn}\}$ and $\{\boldsymbol{c}^s_{kl}\}$ according to Eq.(1) and Eq.(5). Thus, we follow [32] to choose features with specified semantics for initializing these centroids.

Specifically, a pre-trained semantic segmentation model (DeepLabV3 [8] in this case) is used to predict the semantic label of local features. The features predicted to be from static semantics are sampled and clustered to generate the K cluster centroids $\{\boldsymbol{c}_k\}$ and the N informative centroids $\{\boldsymbol{c}^r_{kn}\}$ of each cluster. Those features predicted to belong to task-irrelevant semantics are clustered to generate a list of shadow candidates. The L shadow centroids $\{\boldsymbol{c}^s_{kl}\}$ of each cluster are chosen to be the top L candidates with the shortest distances to the cluster centroid.

Via the special initialization, local features with misleading semantics are most likely to be closest to a shadow centroid and are assigned a low saliency weight. This provides initial attention for the local weighting scheme. Considering that rough priors and initial attention may not be perfect for the VPR task, we further fine-tune the network through end-to-end training to obtain the optimal attention reasoning. In this way, the ultimate reasoning habits can be learned from both prior knowledge and data-driven learning. Fig.4 visualizes the inferred local saliency of some samples.

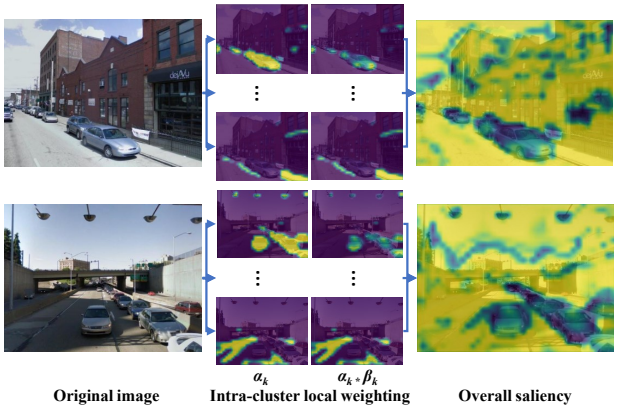


Figure 4. Visualization of local refinement on examples from Pittsburgh dataset. From the intra-cluster saliency of visual cues before and after local weighting (2^{nd} column), it can be seen that vehicles are suppressed and sidewalks are preserved.

4.2. Global Integration

After leveraging individual saliency, we further combine spatial and cluster saliency to embed the refined local features into the global image representation.

Attentional Pyramid Pooling. Conventionally, a visual word vector is generated through Eq.(2), where spatial information between local features is decoupled and overlooked. However, a feature's task relevance also largely depends on the context in the scene. (*e.g.*, as in Fig.1, window structure of vehicles or buildings varies in importance to the

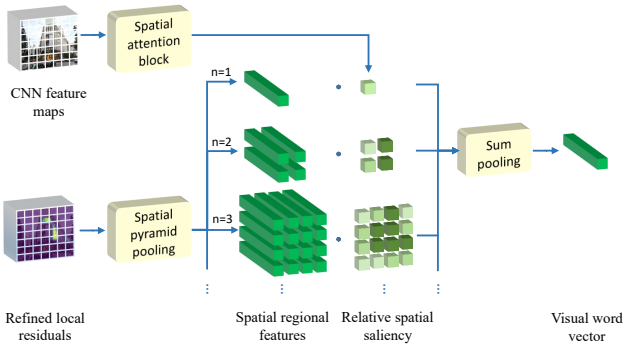


Figure 5. The diagram of attentional pyramid pooling for generating cluster-wise visual word vectors. Spatial pyramid pooling is performed on the refined local residuals to generate multi-scale regional features, the relative spatial saliency of which is evaluated by the spatial attention block. The final visual word vector is formulated by weighted sum pooling of the regional features.

task.) Therefore, we propose an attentional pyramid pooling module to highlight the local residuals with informative context when generating visual word vectors. Specifically, an overlapping pyramid structure [51] is first employed to divide the feature maps into multi-scale regions. Through a sliding window of size $(\lceil 2W/(2^{n-1}{+}1)\rceil, \lceil 2H/(2^{n-1}{+}1)\rceil)$ and stride $(\lceil W/(2^{n-1}{+}1)\rceil, \lceil H/(2^{n-1}{+}1)\rceil)$, a total of 4^{n-1} spatial grids will be generated at level n of the pyramid, where adjacent grids overlap by approximately 50% for better entities alignment. Let $\boldsymbol{f}_k^{n,m}$ be the m^{th} regional feature at the n^{th} pyramid level. It can be obtained by aggregating the salient cluster-wise residuals within the spatial grid.

$$\boldsymbol{f}_k^{n,m} = \sum_{i=1}^{H_{grid}W_{grid}} \alpha_k(\boldsymbol{x}_i)\beta_k(\boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{c}_k^r) \qquad (6)$$

Since not all regional features describe the informative area, a spatial attention block is introduced to adjust their contribution to feature embedding. The spatial attention block consists of the same number of convolutional layers as the pyramid levels. At each pyramid level, a convolutional layer with the same kernel size and stride as the sliding window is employed to evaluate the distinctiveness of each spatial region. Then the multi-scale regional features $\boldsymbol{f}_k^{n,m}$ and their distinctiveness $\mu_k^{n,m}$ (which denotes the spatial distinctiveness of the m^{th} region at the n^{th} pyramid level with regard to the k^{th} cluster) are stacked as $\boldsymbol{f}_k = [\boldsymbol{f}_k^{1,1}...\boldsymbol{f}_k^{n,m}...\boldsymbol{f}_k^{N,4^{N-1}}]$ and $\boldsymbol{\mu}_k = [\mu_k^{1,1}...\mu_k^{n,m}...\mu_k^{N,4^{N-1}}]$ respectively. The relative spatial saliency $\tilde{\boldsymbol{\mu}}_k$ is calculated by $L2$-normalizing the $\boldsymbol{\mu}_k$, so that each element $\tilde{\mu}_k^{n,m}$ is related to the global context. Finally, the k^{th} visual word vector $\boldsymbol{V}_k$ is generated by aggregating the multi-scale regional features weighted by their relative spatial saliency.

$$\boldsymbol{V}_k = \boldsymbol{f}_k^{\intercal}\tilde{\boldsymbol{\mu}}_k = \sum_{n=1}^{N}\sum_{m=1}^{4^{n-1}} \tilde{\mu}_k^{n,m}\boldsymbol{f}_k^{n,m} \qquad (7)$$

Parametric Normalization. After getting the visual word vectors, the image descriptor of VLAD variants [1, 17, 34] is normally obtained through Eq.(3). During indexing, the similarity of two images $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ is measured by the inner product of their descriptors $\boldsymbol{V}$, which can be further decomposed as the average sum of similarities between the corresponding normalized visual word vectors $\tilde{\boldsymbol{V}}_k$:

$$\begin{aligned} Sim\langle \boldsymbol{X}_1, \boldsymbol{X}_2\rangle &= \boldsymbol{V}(\boldsymbol{X_1}) \cdot \boldsymbol{V}(\boldsymbol{X_2}) \\ &= \frac{1}{K}\sum\nolimits_{k=1}^{K} \tilde{\boldsymbol{V}}_{\boldsymbol{k}}(\boldsymbol{X}_1) \cdot \tilde{\boldsymbol{V}}_{\boldsymbol{k}}(\boldsymbol{X}_2) \end{aligned} \qquad (8)$$

It can be seen that all clusters have the same contribution to the similarity metric, while their significance to the task may not be the same. *e.g.*, a cluster containing architectural features is more important than that containing sky features.

Therefore, we propose a parametric normalization, where trainable parameter γ_k is introduced to quantify the importance of the k^{th} visual cluster to the task. In implementation, the trainable weights $\gamma = [\gamma_1, \gamma_2, ..., \gamma_K]$ is first L_2-normalized as the cluster saliency $\tilde{\gamma} = [\tilde{\gamma}_1, \tilde{\gamma}_2, ..., \tilde{\gamma}_K]$. Then a unit image descriptor can be generated by concatenating the normalized visual word vectors $\tilde{V}_K$ rescaled by their corresponding cluster saliency $\tilde{\gamma}_K$.

$$\boldsymbol{V}(\boldsymbol{X}) = [\tilde{\gamma}_1 \cdot \tilde{\boldsymbol{V}}_1(\boldsymbol{X}), \tilde{\gamma}_2 \cdot \tilde{\boldsymbol{V}}_2(\boldsymbol{X}), ..., \tilde{\gamma}_K \cdot \tilde{\boldsymbol{V}}_K(\boldsymbol{X})] \qquad (9)$$

In this way, the similarity of two image descriptors can be derived as Eq.(10), where $\tilde{\gamma}_k^2$ distinguishes the contribution of visual word clusters to the similarity metric.

$$Sim\langle \boldsymbol{X}_1, \boldsymbol{X}_2\rangle = \sum\nolimits_{k=1}^{K} \tilde{\gamma}_k^2 \cdot \tilde{\boldsymbol{V}}_k(\boldsymbol{X}_1) \cdot \tilde{\boldsymbol{V}}_k(\boldsymbol{X}_2) \qquad (10)$$

Compare Eq.(9) with Eq.(3), the representations differ in the element-wise scalar γ_k. If $\tilde{\gamma}_k$ equals to $\frac{1}{\sqrt{K}}$ for each $k \in [1, K]$, the similarity metric in Eq.(10) will degenerate into Eq.(8). Thus, the intra-normalization followed by L_2-normalization in [1, 17] can be regarded as a special case of our proposed parametric normalization. During training, $\tilde{\gamma}_k$ is initialized by $\frac{1}{\sqrt{K}}$ to mimic the conventional normalization pipeline, where all visual word vectors are considered equally important to the similarity metric in the beginning.

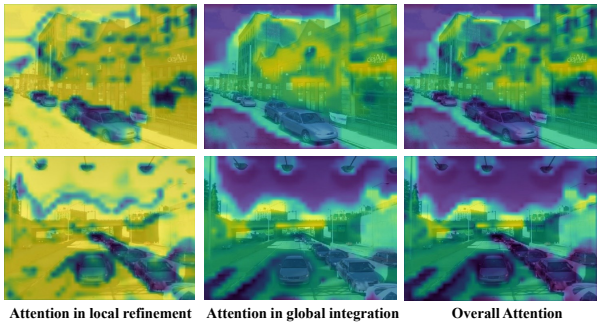


Figure 6. Using the same images in Fig.4 for visualization, global integration (2^{nd} column) places greater emphasis on building structures. Local and global attention complement each other.

5. Loss Function

In the retrieval-based VPR task, reference images with the representations closest to the query are considered as potential positive candidates. For accurate indexing, a positive reference image X_r^p is expected to be closer to the query X_q than any negative reference X_r^n in the feature space. Thus, we use the triplet ranking loss [14–16,36] in Eq.(11) to train the image representation f_θ, where $[y]_+ = max(y, 0)$ ensures non-negative output and m is the empirical margin.

$$l_\theta(X_q, X_r^p, X_r^n) = [d^2(f_\theta(X_q), f_\theta(X_r^p)) - d^2(f_\theta(X_q), f_\theta(X_r^n)) + m]_+ \quad (11)$$

Since the loss takes triplets as input, we use the same tuple mining strategy as in [1] to collect a set of tuples $(X_q, X_r^{p*}, \{X_r^n\})$ for each training iteration. A tuple consists of 1 query, 1 positive and N negatives, which can be combined into N triplets $(X_q, X_r^{p*}, X_r^{nj})$. The training loss of each tuple is formulated as Eq.(12), and the parametric model is trained by minimizing it. To enhance the robustness of the learned representation, data augmentation is performed, including random cropping and color jitter.

$$L_\theta(X_q, X_r^{p*}, \{X_r^n\}) = \frac{1}{N}\sum_{j=1}^{N} l_\theta(X_q, X_r^{p*}, X_r^{nj}) \quad (12)$$

6. Experiments

6.1. Benchmark Datasets

We use the three common VPR benchmark datasets, namely Pitts250k, Pitts30k, and Tokyo24/7, to evaluate the proposed APPSVR. Pitts250k [46] collects 254k images from the Pittsburgh area, with changes in both appearance and viewpoint. Pitts30k [1] is proposed as a subset of Pitts250k to speed up the training and evaluation process. Tokyo24/7 [1, 45] contains 76k images captured in Tokyo during daytime, sunset and night. It is relatively more challenging in terms of larger illumination change, more clutters and dynamic objects. Following the latest SOTA [32, 48], we train all evaluated models on Pitts30k-train and test them on Pitts30k-test, Pitts250k-test, Tokyo24/7 respectively. Fig.7 visualizes some examples of challenging image pairs in the employed datasets.

6.2. Evaluation Metric

The performance of evaluated models is measured by $Recall@N$, which is the percentage of the queries that are correctly retrieved when given N potential positive candidates. The candidates are the first N reference images with the most similar representations to the query. For Pitts250k, Pitts30k and Tokyo24/7, the retrieval reasoning of a query is correct if any of the candidates is not more than $d_r = 25m$ away from the query image.

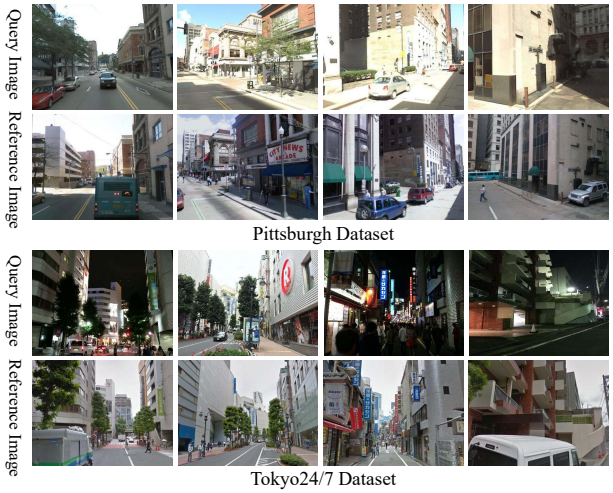


Figure 7. Examples of challenging image pairs in Pittsburgh and Tokyo24/7. (APPSVR correctly retrieves all queries above)

6.3. Implementation Details

We crop the VGG-16 [42] pretrained on ImageNet at the last convolutional layer, and use it as the unified base network to extract local features. APPSVR and other comparative models are appended as a pooling layer to generate the global image descriptor. PCA whitening (PCA-W) is further performed to formulate more compact representations with 4096 and 512 dimensions (4096-D and 512-D). Through hyperparameter tuning, the number of informative centroids N and shadow centroids L of each cluster in APPSVR are selected as 1 and 4 respectively. Other common hyperparameters for model building and training configuration are chosen to be the same as in [1, 32]. For a fair comparison, we implement all methods in PyTorch, using the unified protocol to train and evaluate them.

6.4. Ablation Study

To analyze the advantages of each component in our proposed method, we compare APPSVR variants that apply different component combinations. In the following experiment, APPSVR with only the local weighting scheme (LW) is set as the basic model. We use additional abbreviations to denote the application of semantic constrained initialization (SC), attentional pyramid pooling (AP) and parametric normalization (PN) to the basic APPSVR.

As shown in Table.1, applying each component progressively can bring steady performance improvements to our model. It demonstrates the effectiveness of individual modules, and shows that their advantages can be accumulated. Increasing the number of scales in the AP module further improves the results, which can be attributed to the multi-scale regional weighting that provides finer entity alignment and spatial saliency regulation. Compared with the baseline NetVLAD, our optimal model with all components en-

Table 1. Ablation study on the proposed components. '-Ln' denotes applying **AP** with n scales. All representations are 4096-D.

Method	Components				Pitts30k-test			Pitts250k-test			Tokyo24/7		
	LW	SC	PN	AP	r@1	r@5	r@10	r@1	r@5	r@10	r@1	r@5	r@10
NetVLAD [1]	×	×	×	×	85.2	92.8	94.9	86.5	93.8	95.5	68.9	78.7	81.3
APPSVR	√	×	×	×	86.1	93.1	94.9	87.5	94.4	95.9	70.8	80.0	82.9
APPSVR-SC	√	√	×	×	86.3	93.4	95.4	87.8	94.8	96.3	72.1	83.2	87.3
APPSVR-PN	√	×	√	×	86.4	93.2	95.3	88.1	94.4	96.0	73.0	82.2	84.1
APPSVR-SC-PN	√	√	√	×	86.4	93.6	95.5	88.1	95.2	96.5	73.7	84.8	88.3
APPSVR-SC-PN-L2	√	√	√	√	87.3	94.2	95.8	88.5	95.4	96.8	75.2	83.8	88.7
APPSVR-SC-PN-L3	√	√	√	√	**87.4**	**94.3**	**95.8**	**88.8**	**95.6**	**96.8**	**77.1**	**85.7**	**89.5**

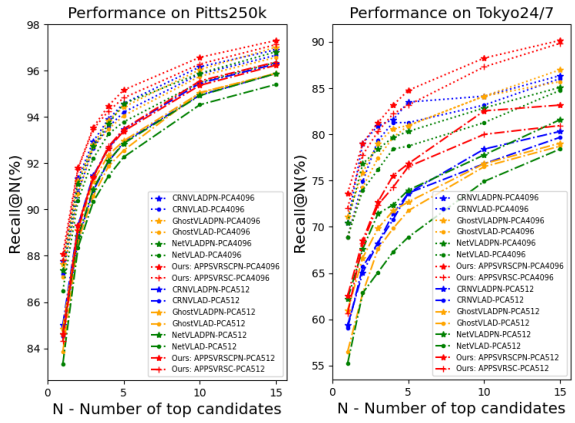


Figure 8. Applying **PN**(-∗-) achieves a coherent refinement in the performances for all models(-··-). Our method(red) shows obvious advantages in both the 4096-D (··) and 512-D (-·) representations.

abled (APPSVR-SC-PN-L3) shows a comprehensive performance advantage. An increase of about 2% and 7% can be observed on Pittsburgh and Tokyo respectively.

We also evaluate the generalization ability of the PN module on other VLAD variants. It can be seen from Fig.8 that applying PN brings stable improvements for all the baseline models. This reflects the advantages of introducing cluster saliency and shows that the proposed PN is generally effective for different VLAD-centric architectures.

6.5. Comparison with State-of-The-Arts

We compare our model with VPR benchmark methods based on global descriptor retrieval. Positioned as an attentional VLAD pooling layer, our APPSVR is first compared with other VLAD-centric architectures. CRN [19] extends NetVLAD [1] with a contextual reweighting layer. SPENetVLAD [49] stacks the regional NetVLAD descriptors to preserve spatial information. GhostVLAD [50] specifies the ghost clusters for misleading features and eliminate them from feature embedding. SRALNet [32] introduces semantic priors to enhance local attention learning. Table 2 shows the comparison results based on the original and 4096-D representations. As can be seen, GhostVLAD, CRN, and APPSVR-SC (SRALNet) all surpass NetVLAD, proving the necessity of introducing local weighting for feature refinement. The methods applying SC (GhostVLAD-SC, SRALNet and ours) consistently outperform the others without SC, which shows the superiority of reinforcing attention learning with semantic priors. Futher applying PN and AP expands our advantages, where APPSVR-SC-PN-L3 achieves a new SOTA performance on both benchmark datasets. This validates our global integration that integrates spatial and cluster saliency into the feature embedding.

Table 2. Performance comparison to other VLAD-centric architectures with original and 4096-D (**bold**) representations.

Method	PCA-W	Pitts250k-test			Tokyo24/7		
		r@1	r@5	r@10	r@1	r@5	r@10
NetVLAD [1]	w/o	84.1	92.5	94.5	60.0	76.2	79.7
	4096D	**86.5**	**93.8**	**95.5**	**68.9**	**78.7**	**81.3**
GhostVLAD [50]	w/o	84.1	92.7	95.1	61.6	76.5	80.6
	4096D	**87.1**	**94.1**	**95.8**	**68.9**	**81.0**	**84.1**
CRN [19]	w/o	84.7	92.9	95.3	61.9	75.6	79.7
	4096D	**87.2**	**94.2**	**95.9**	**69.0**	**81.3**	**83.2**
SPENetVLAD [48]	w/o	85.2	93.4	95.3	60.0	74.6	81.3
	4096D	**87.4**	**94.5**	**96.1**	**71.8**	**82.9**	**87.6**
GhostVLAD-SC [32]	w/o	85.5	93.8	95.6	66.0	78.4	84.4
	4096D	**87.6**	**94.6**	**96.1**	**71.4**	**81.9**	**85.7**
SRALNet [32]	w/o	85.8	94.1	95.9	68.6	80.0	83.8
	4096D	**87.8**	**94.8**	**96.3**	**72.1**	**83.2**	**87.3**
APPSVR-SC-PN-L3	w/o	86.6	94.6	96.3	68.3	81.3	85.7
	4096D	**88.8**	**95.6**	**96.8**	**77.1**	**85.7**	**89.5**

Then we compare the performance of models on a more compact 512-D representation. We include more methods whose original descriptor is 512-D. R-MAC [44] formulates the descriptor by summing the maximum activations within spatial grids. APAnet [51] aggregates multi-scale regional features weighted by cascaded attention blocks. GeM [37] generalizes max and average pooling by introducing a learnable pooling parameter. Their performance are given in Table 3. It can be observed that the VLAD-centric methods show an overall advantage over the global pooling methods, which may be attributed to residuals and clusters that can better characterize local details and distinguish subtle differences. It verifies the rationality of our use of cluster-wise residual aggregation for feature embedding. Besides, our method outperforms all comparative models in Table 3, which further demonstrates the compelling advantages of APPSVR in different dimensional representations. [1]

[1] More results and visualization can refer to the supplementary material.

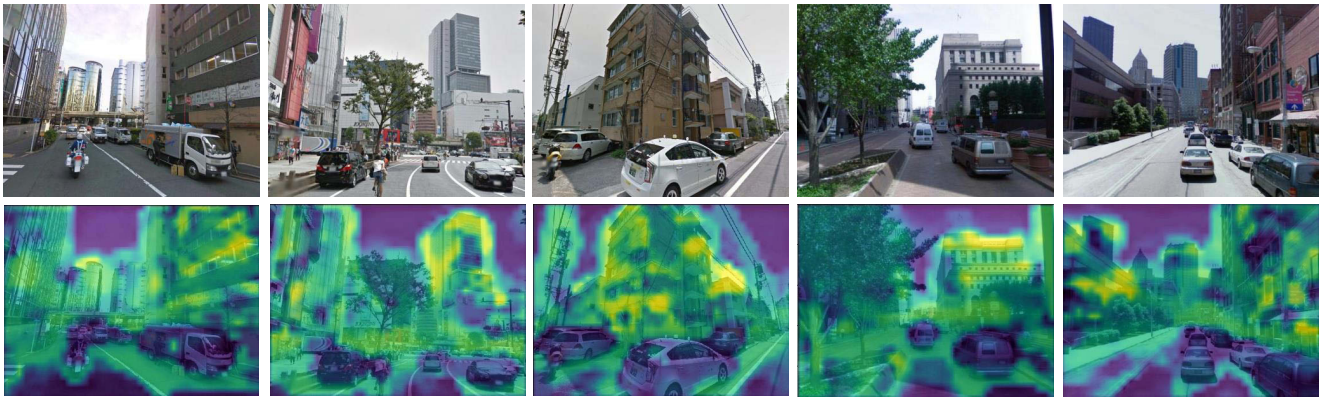


Figure 9. The illustration of visual cue saliency in the image representation learned by APPSVR. Overall, static building structures are the most highlighted (bright yellow) while dynamic objects, such as vehicles and riders, are strongly suppressed (dark purple). Some other interesting details can also be observed: for roadside trees (4^{th} column), the trunks are assigned with higher weights than the canopy.

Table 3. Performance comparison of methods based on 512-D representation. For a fair comparison, all evaluated models are implemented in PyTorch and trained using the same protocol.

Method	PCA-W	Pitts250k-test			Tokyo24/7		
		r@1	r@5	r@10	r@1	r@5	r@10
Max pooling [38]	512D	39.3	59.0	67.2	11.8	23.5	33.3
Sum pooling [3]	512D	70.7	84.1	88.5	28.6	43.8	53.0
R-Mac [44]	512D	76.3	88.6	92.0	40.3	60.6	67.0
APAnet [51]	512D	76.7	88.8	91.7	51.1	66.7	71.1
GeM [37]	512D	80.4	91.0	93.7	56.2	72.4	80.0
NetVLAD [1]	512D	83.3	92.3	94.5	55.2	68.9	74.9
GhostVLAD [50]	512D	83.9	92.6	95.1	56.5	71.8	76.5
SPENetVLAD [48]	512D	84.5	93.0	94.8	59.0	71.8	78.7
CRN [19]	512D	84.5	92.9	95.0	59.1	73.7	76.8
SRALNet [32]	512D	84.8	93.5	95.6	60.6	76.5	80.0
APPSVR-SC-PN-L3	512D	**85.3**	**94.0**	**95.8**	**62.0**	**76.5**	**80.0**

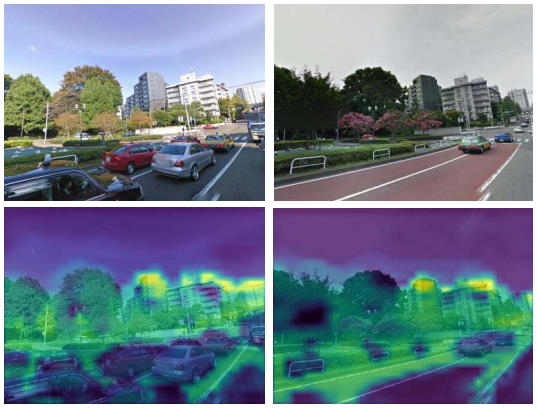


Figure 10. The two images depict the same scene under different illumination and seasonal conditions. APPSVR focuses on the static objects (buildings & road marks) and ignores the dynamic vehicles, which guarantees the robustness of the representation.

6.6. Qualitative Results.

In Fig.9, we take several images from Pittsburgh and TokyoTM datasets as examples to visualize the overall attention learned by APPSVR. The super-imposed heat maps illustrate the divergent importance of visual cues to the image representation. It can be seen that the regions associated with architectures are highlighted, while the others representing vehicles and riders are mostly suppressed. Impressively, for roadside trees, the trunks are assigned with higher weights than the canopy, from which one can further conclude that APPSVR can automatically learn to distinguish the long-term static objects from those time-varying ones. Fig.10 visually shows how APPSVR perceives two images from the same scene but with different appearances. [1]

7. Conclusions

In this paper, we propose an attention-aware encoding strategy APPSVR for visual place recognition. To inhibit misleading visual elements in feature embedding, we use a semantic-reinforced local weighting scheme for local feature refinement. To highlight the varying saliency of task-relevant visual elements in the image representation, we propose an attentional pyramid pooling and a parametric normalization to combine the spatial and cluster saliency in global integration. Experiments demonstrate that the introduced triple attention can bring stable performance advantages and better generalization capability to the model. Quantitatively, our architecture consistently outperforms the SOTA methods on varying dimensional representations. Qualitatively, the visualization shows that the learned attention under a weakly supervised manner is largely consistent with human cognition, which highlights long-term static objects while suppressing misleading ones.

Acknowledgement

This study is supported under the RIE2020 Industry Alignment Fund-Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

[1] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016.

[2] Relja Arandjelovic and Andrew Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *ACCV*, 2014.

[3] Artem Babenko and Victor S. Lempitsky. Aggregating local deep features for image retrieval. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277, 2015.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110:346–359, 2008.

[5] Song Cao and Noah Snavely. Graph-based discriminative learning for location recognition. In *CVPR*, 2013.

[6] Eric Chalmers, Edgar Bermudez Contreras, Brandon Robertson, Artur Luczak, and Aaron J. Gruber. Learning to predict consequences as a method of knowledge transfer in reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 29:2259–2270, 2018.

[7] David M. Chen, Georges Baatz, Kevin Köser, Sam S. Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, Bernd Girod, and Radek Grzeszczuk. City-scale landmark identification on mobile devices. *CVPR 2011*, pages 737–744, 2011.

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[9] David J. Crandall, Andrew Owens, Noah Snavely, and Daniel P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR*, 2011.

[10] Mark Cummins and Paul M. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *I. J. Robotics Res.*, 30(9):1100–1123, 2011.

[11] Mark Joseph Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *I. J. Robotics Res.*, 27:647–665, 2008.

[12] Matthijs C Dorst. Distinctive image features from scale-invariant keypoints. 2011.

[13] Albert Gordo, Jon Almazán, Jérôme Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, 2016.

[14] Albert Gordo, Jon Almazán, Jérôme Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124:237–254, 2016.

[15] Y. Gu, K. Vyas, M. Shen, J. Yang, and G. Yang. Deep graph-based multimodal feature embedding for endomicroscopy image retrieval. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–12, 2020.

[16] N. Gupta, S. Mujumdar, S. Samanta, and S. Mehta. Learning an order preserving image similarity through deep ranking. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1115–1120, Aug 2018.

[17] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010.

[18] Ahmad Khaliq, Shoaib Ehsan, Michael Milford, and Klaus D. McDonald-Maier. A holistic visual place recognition approach using lightweight cnns for severe viewpoint and appearance changes. *ArXiv*, abs/1811.03032, 2018.

[19] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3251–3260, 2017.

[20] Jan Knopp, Josef Sivic, and Tomás Pajdla. Avoiding confusing features in place recognition. In *ECCV*, 2010.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60:84–90, 2012.

[22] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 920–929, 2017.

[23] Hyon Lim, Sudipta N. Sinha, Michael F. Cohen, and Matthew Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1043–1050, 2012.

[24] Colin McManus, Winston Churchill, William P. Maddern, Alexander D. Stewart, and Paul Newman. Shady dealings: Robust, long-term visual localisation using illumination invariance. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 901–906, 2014.

[25] Nate Merrill and Guoquan Huang. Lightweight unsupervised deep loop closure. *ArXiv*, abs/1805.07703, 2018.

[26] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In *ECCV*, 2014.

[27] Arsalan Mousavian, Jana Kosecka, and Jyh-Ming Lien. Semantically guided location recognition for outdoors scenes. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4882–4889, 2015.

[28] Raul Mur-Artal, José Maria Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31:1147–1163, 2015.

[29] Tayyab Naseer, Gabriel L. Oliveira, Thomas Brox, and Wolfram Burgard. Semantics-aware visual localization under challenging perceptual conditions. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2614–2620, 2017.

[30] Joe Yue-Hei Ng, Fan Yang, and Larry S. Davis. Exploiting local features from deep networks for image retrieval. *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 53–61, 2015.

[31] Hyeonwoo Noh, André Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3476–3485, 2016.

[32] G Peng, Y Yue, J Zhang, Z Wu, X Tang, and D Wang. Semantic reinforced attention learning for visual place recognition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[33] Florent Perronnin and Christopher R. Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*. IEEE Computer Society, 2007.

[34] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3384–3391, 2010.

[35] Nathan Piasco, Desire Sidibé, Valérie Gouet-Brunet, and Cédric Demonceaux. Learning scene geometry for visual localization in challenging conditions. *2019 International Conference on Robotics and Automation (ICRA)*, pages 9094–9100, 2019.

[36] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016.

[37] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019.

[38] Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. A baseline for visual instance retrieval with deep convolutional networks. In *ICLR 2015*, 2014.

[39] Torsten Sattler, Michal Havlena, Filip Radenović, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2102–2110, 2015.

[40] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012.

[41] Grant Schindler, Matthew A. Brown, and Richard Szeliski. City-scale location recognition. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[43] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In *Robotics: Science and Systems*, 2015.

[44] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *CoRR*, abs/1511.05879, 2015.

[45] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015.

[46] Akihiko Torii, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. Visual place recognition with repetitive structures. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 883–890, 2013.

[47] Tiberio Uricchio, Marco Bertini, Lorenzo Seidenari, and Alberto Del Bimbo. Fisher encoded convolutional bag-of-windows for efficient image retrieval and social image tagging. *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 1020–1026, 2015.

[48] J. Yu, C. Zhu, J. Zhang, Q. Huang, and D. Tao. Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2019.

[49] Jun Yu, Chaoyang Zhu, Jian Zhang, Qingming Huang, and Dacheng Tao. Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. *IEEE transactions on neural networks and learning systems*, 2019.

[50] Yujie Zhong, Relja Arandjelovic, and Andrew Zisserman. Ghostvlad for set-based face recognition. In *ACCV*, 2018.

[51] Yingying Zhu, Jiong Wang, Lingxi Xie, and Liang Zheng. Attention-based pyramid aggregation network for visual place recognition. In *ACM Multimedia*, 2018.