NetVLAD: CNN architecture for weakly supervised place recognition

Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, Josef Sivic

NetVLAD: CNN architecture for weakly supervised place recognition

Relja Arandjelović
INRIA *

Petr Gronat
INRIA*

Akihiko Torii
Tokyo Tech †

Tomas Pajdla
CTU in Prague ‡

Josef Sivic
INRIA*

Abstract

We tackle the problem of large scale visual place recognition, where the task is to quickly and accurately recognize the location of a given query photograph. We present the following three principal contributions. First, we develop a convolutional neural network (CNN) architecture that is trainable in an end-to-end manner directly for the place recognition task. The main component of this architecture, NetVLAD, is a new generalized VLAD layer, inspired by the "Vector of Locally Aggregated Descriptors" image representation commonly used in image retrieval. The layer is readily pluggable into any CNN architecture and amenable to training via backpropagation. Second, we develop a training procedure, based on a new weakly supervised ranking loss, to learn parameters of the architecture in an end-to-end manner from images depicting the same places over time downloaded from Google Street View Time Machine. Finally, we show that the proposed architecture obtains a large improvement in performance over non-learnt image representations, significantly outperforms off-the-shelf CNN descriptors on two challenging place recognition benchmarks, and outperforms current state-of-the-art compact image representations on standard image retrieval benchmarks.

1. Introduction

Visual place recognition has received a significant amount of attention in the past years both in computer vision [4, 9, 10, 24, 35, 62, 63, 64, 65, 79, 80] and robotics communities [15, 16, 43, 45, 74] motivated by, *e.g.*, applications in autonomous driving [45], augmented reality [46] or geo-localizing archival imagery [5].

The place recognition problem, however, still remains extremely challenging. How can we recognize the same street-corner in the entire city or on the scale of the entire country despite the fact it can be captured in different illuminations or change its appearance over time? The fundamental scientific question is what is the appropriate representation of a place that is rich enough to distinguish similarly looking places yet compact to represent entire cities or countries.

(a) Mobile phone query (b) Retrieved image of same place

Figure 1. Our trained NetVLAD descriptor correctly recognizes the location (b) of the query photograph (a) despite the large amount of clutter (people, cars), changes in viewpoint and completely different illumination (night vs daytime). **Please see appendix C for more examples.**

The place recognition problem has been traditionally cast as an instance retrieval task, where the query image location is estimated using the locations of the most visually similar images obtained by querying the large geo-tagged database [4, 10, 35, 65, 79, 80]. Each database image is represented using local invariant features [82] such as SIFT [42] that are aggregated into a single vector representation for the entire image such as bag-of-visual-words [52, 73], VLAD [3, 29] or Fisher vector [31, 51]. The resulting representation is then usually compressed and efficiently indexed [28, 73]. The image database can be further augmented by 3D structure that enables recovery of accurate camera pose [40, 62, 63].

In the last few years convolutional neural networks (CNNs) [38, 39] have emerged as powerful image representations for various category-level recognition tasks such as object classification [37, 48, 72, 76], scene recognition [89] or object detection [21]. The basic principles of CNNs are known from 80's [38, 39] and the recent successes are a combination of advances in GPU-based computation power together with large labelled image datasets [37]. While it has been shown that the trained representations are, to some

*WILLOW project, Departement d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548.

†Department of Mechanical and Control Engineering, Graduate School of Science and Engineering, Tokyo Institute of Technology

‡Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague

extent, transferable between recognition tasks [19, 21, 48, 68, 87], a direct application of CNN representations trained for object classification [37] as black-box descriptor extractors has so far yielded limited improvements in performance on instance-level recognition tasks [6, 7, 22, 59, 61]. In this work we investigate whether this gap in performance can be bridged by CNN representations developed and trained directly for place recognition. This requires addressing the following three main challenges. First, what is a good CNN architecture for place recognition? Second, how to gather sufficient amount of annotated data for the training? Third, how can we train the developed architecture in an end-to-end manner tailored for the place recognition task? To address these challenges we bring the following three innovations.

First, building on the lessons learnt from the current well performing hand-engineered object retrieval and place recognition pipelines [2, 3, 25, 79] we develop a convolutional neural network architecture for place recognition that aggregates mid-level (conv5) convolutional features extracted from the entire image into a compact single vector representation amenable to efficient indexing. To achieve this, we design a new trainable generalized VLAD layer, NetVLAD, inspired by the Vector of Locally Aggregated Descriptors (VLAD) representation [29] that has shown excellent performance in image retrieval and place recognition. The layer is readily pluggable into any CNN architecture and amenable to training via backpropagation. The resulting aggregated representation is then compressed using Principal Component Analysis (PCA) to obtain the final compact descriptor of the image.

Second, to train the architecture for place recognition, we gather a large dataset of multiple panoramic images depicting the same place from different viewpoints over time from the Google Street View Time Machine. Such data is available for vast areas of the world, but provides only weak form of supervision: we know the two panoramas are captured at approximately similar positions based on their (noisy) GPS but we don't know which parts of the panoramas depict the same parts of the scene.

Third, we develop a learning procedure for place recognition that learns parameters of the architecture in an end-to-end manner tailored for the place recognition task from the weakly labelled Time Machine imagery. The resulting representation is robust to changes in viewpoint and lighting conditions, while simultaneously learns to focus on the relevant parts of the image such as the building façades and the skyline, while ignoring confusing elements such as cars and people that may occur at many different places.

We show that the proposed architecture obtains a large improvement in performance over non-learnt image representations on two challenging place recognition benchmarks as well as significantly outperforms off-the-shelf CNNs trained for object and scene classification.

1.1. Related work

While there have been many improvements in designing better image retrieval [2, 3, 11, 12, 17, 25, 26, 27, 29, 32, 47, 50, 51, 52, 53, 70, 77, 78, 81] and place recognition [4, 9, 10, 15, 16, 24, 35, 43, 45, 62, 63, 64, 74, 79, 80] systems, not many works have performed learning for these tasks. All relevant learning-based approaches fall into one or both of the following two categories: (i) learning for an auxiliary task (*e.g*. some form of distinctiveness of local features [4, 15, 30, 35, 57, 58, 88]), and (ii) learning on top of shallow hand-engineered descriptors that cannot be fine-tuned for the target task [2, 9, 24, 35, 56]. Both of these are in spirit opposite to the core idea behind deep learning that has provided a major boost in performance in various recognition tasks: end-to-end learning. We will indeed show in section 5.2 that training representations directly for the end-task, place recognition, is crucial for obtaining good performance.

Numerous works concentrate on learning better local descriptors or metrics to compare them [44, 47, 49, 54, 55, 69, 70, 86], but even though some of them show results on image retrieval, the descriptors are learnt on the task of matching local image patches, and not directly with image retrieval in mind. Some of them also make use of hand-engineered features to bootstrap the learning, *i.e*. to provide noisy training data [44, 47, 49, 54, 70].

Several works have investigated using CNN-based features for image retrieval. These include treating activations from certain layers directly as descriptors by concatenating them [8, 59], or by pooling [6, 7, 22]. However, none of these works actually train the CNNs for the task at hand, but use CNNs as black-box descriptor extractors. One exception is the work of Babenko *et al*. [8] in which the network is fine-tuned on an auxiliary task of classifying 700 landmarks. However, again the network is not trained directly on the target retrieval task.

Finally, recently [34] and [41] performed end-to-end learning for different but related tasks of ground-to-aerial matching [41] and camera pose estimation [34].

2. Method overview

Building on the success of current place recognition systems (*e.g*. [4, 10, 35, 62, 63, 64, 65, 79, 80]), we cast place recognition as image retrieval. The query image with unknown location is used to visually search a large geotagged image database, and the locations of top ranked images are used as suggestions for the location of the query. This is generally done by designing a function f which acts as the "image representation extractor", such that given an image I_i it produces a fixed size vector $f(I_i)$. The function is used to extract the representations for the entire database $\{I_i\}$,

which can be done offline, and to extract the query image representation $f(q)$, done online. The visual search is then performed by simply finding the nearest database image to the query, either exactly or through fast approximate nearest neighbour search, by sorting images based on the Euclidean distance $d(q, I_i)$ between $f(q)$ and $f(I_i)$.

While previous works have mainly used hand-engineered image representations (*e.g.* $f(I)$ corresponds to extracting SIFT descriptors [42], followed by pooling into a bag-of-words vector [73] or a VLAD vector [29]), here we propose to learn the representation $f(I)$ in an end-to-end manner, directly optimized for the task of place recognition. The representation is parametrized with a set of parameters θ and we emphasize this fact by referring to it as $f_\theta(I)$. It follows that the Euclidean distance $d_\theta(I_i, I_j) = \|f_\theta(I_i) - f_\theta(I_j)\|$ also depends on the same parameters. An alternative setup would be to learn the distance function itself, but here we choose to fix the distance function to be Euclidean distance, and to pose our problem as the search for the explicit feature map f_θ which works well under the Euclidean distance.

In section 3 we describe the proposed representation f_θ based on a new deep convolutional neural network architecture inspired by the compact aggregated image descriptors for instance retrieval. In section 4 we describe a method to learn the parameters θ of the network in an end-to-end manner using weakly supervised training data from the Google Street View Time Machine.

3. Deep architecture for place recognition

This section describes the proposed CNN architecture f_θ, guided by the best practises from the image retrieval community. Most image retrieval pipelines are based on (i) extracting local descriptors, which are then (ii) pooled in an orderless manner. The motivation behind this choice is that the procedure provides significant robustness to translation and partial occlusion. Robustness to lighting and viewpoint changes are provided by the descriptors themselves, and scale invariance is ensured through extracting descriptors at multiple scales.

In order to learn the representation end-to-end, we design a CNN architecture that mimics this standard retrieval pipeline in an unified and principled manner with differentiable modules. For step (i), we crop the CNN at the last convolutional layer and view it as a dense descriptor extractor. This has been observed to work well for instance retrieval [6, 7, 61] and texture recognition [13]. Namely, the output of the last convolutional layer is a $H \times W \times D$ map which can be considered as a set of D-dimensional descriptors extracted at $H \times W$ spatial locations. For step (ii) we design a new pooling layer inspired by the Vector Locally Aggregated Descriptors (VLAD) [29] that pools the extracted descriptors into a fixed image representation and its parameters are learnable via back-propagation. We call this new pooling layer "NetVLAD" layer and describe it in the next section.

3.1. NetVLAD: A Generalized VLAD layer (f_{VLAD})

Vector of Locally Aggregated Descriptors (VLAD) [29] is a popular descriptor pooling methods for both instance level retrieval [29] and image classification [22]. It captures information about the statistics of local descriptors aggregated over the image. Whereas bag-of-visual-words [14, 73] aggregation keeps counts of visual words, VLAD stores the sum of residuals (difference between descriptor and its corresponding cluster centre) for each visual word.

Formally, given N D-dimensional local image descriptors $\{\mathbf{x}_i\}$ as input, and K cluster centres ("visual words") $\{\mathbf{c}_k\}$ as VLAD parameters, the output VLAD image representation V is $K \times D$-dimensional. For convenience we will write V as a $K \times D$ matrix, but this matrix is converted into a vector and, after normalization, used as the image representation. The (j, k) element of V is computed as follows:

$$V(j,k) = \sum_{i=1}^{N} a_k(\mathbf{x}_i) \left(x_i(j) - c_k(j)\right), \qquad (1)$$

where $x_i(j)$ and $c_k(j)$ are the j-th dimensions of the i-th descriptor and k-th cluster centre, respectively. $a_k(\mathbf{x}_i)$ denotes the membership of the descriptor $\mathbf{x}_i$ to k-th visual word, *i.e.* it is 1 if cluster $\mathbf{c}_k$ is the closest cluster to descriptor $\mathbf{x}_i$ and 0 otherwise. Intuitively, each D-dimensional column k of V records the sum of residuals $(\mathbf{x}_i - \mathbf{c}_k)$ of descriptors which are assigned to cluster $\mathbf{c}_k$. The matrix V is then L2-normalized column-wise (intra-normalization [3]), converted into a vector, and finally L2-normalized in its entirety [29].

In order to profit from years of wisdom produced in image retrieval, we propose to mimic VLAD in a CNN framework and design a trainable generalized VLAD layer, *NetVLAD*. The result is a powerful image representation trainable end-to-end on the target task (in our case place recognition). To construct a layer amenable to training via backpropagation, it is required that the layer's operation is differentiable with respect to all its parameters and the input. Hence, the key challenge is to make the VLAD pooling differentiable, which we describe next.

The source of discontinuities in VLAD is the hard assignment $a_k(\mathbf{x}_i)$ of descriptors $\mathbf{x}_i$ to clusters centres $\mathbf{c}_k$. To make this operation differentiable, we replace it with soft assignment of descriptors to multiple clusters

$$\bar{a}_k(\mathbf{x}_i) = \frac{e^{-\alpha\|\mathbf{x}_i - \mathbf{c}_k\|^2}}{\sum_{k'} e^{-\alpha\|\mathbf{x}_i - \mathbf{c}_{k'}\|^2}}, \qquad (2)$$

which assigns the weight of descriptor $\mathbf{x}_i$ to cluster $\mathbf{c}_k$ proportional to their proximity, but relative to proximities to

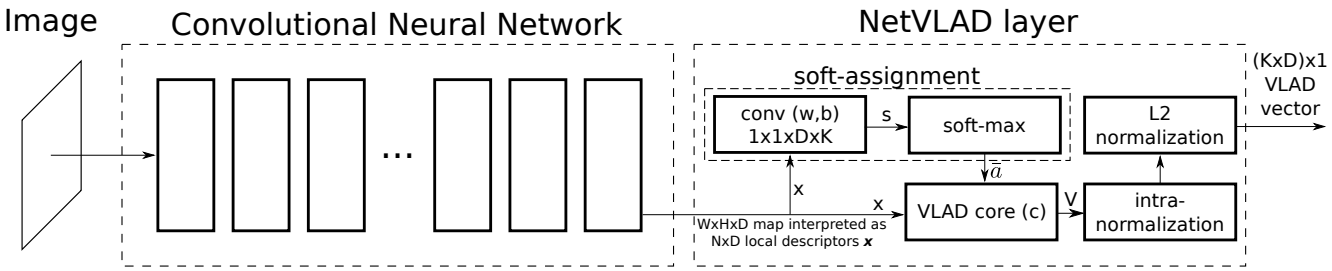


Figure 2. **CNN architecture with the NetVLAD layer.** The layer can be implemented using standard CNN layers (convolutions, softmax, L2-normalization) and one easy-to-implement aggregation layer to perform aggregation in equation (4) ("VLAD core"), joined up in a directed acyclic graph. Parameters are shown in brackets.

other cluster centres. $\bar{a}_k(\mathbf{x}_i)$ ranges between 0 and 1, with the highest weight assigned to the closest cluster centre. α is a parameter (positive constant) that controls the decay of the response with the magnitude of the distance. Note that for $\alpha \to +\infty$ this setup replicates the original VLAD exactly as $\bar{a}_k(\mathbf{x}_i)$ for the closest cluster would be 1 and 0 otherwise.

By expanding the squares in (2), it is easy to see that the term $e^{-\alpha\|\mathbf{x}_i\|^2}$ cancels between the numerator and the denominator resulting in a soft-assignment of the following form

$$\bar{a}_k(\mathbf{x}_i) = \frac{e^{\mathbf{w}_k^T \mathbf{x}_i + b_k}}{\sum_{k'} e^{\mathbf{w}_{k'}^T \mathbf{x}_i + b_{k'}}}, \tag{3}$$

where vector $\mathbf{w}_k = 2\alpha\mathbf{c}_k$ and scalar $b_k = -\alpha\|\mathbf{c}_k\|^2$. The final form of the NetVLAD layer is obtained by plugging the soft-assignment (3) into the VLAD descriptor (1) resulting in

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{\mathbf{w}_k^T \mathbf{x}_i + b_k}}{\sum_{k'} e^{\mathbf{w}_{k'}^T \mathbf{x}_i + b_{k'}}} \left(x_i(j) - c_k(j)\right), \tag{4}$$

where $\{\mathbf{w}_k\}$, $\{b_k\}$ and $\{\mathbf{c}_k\}$ are sets of trainable parameters for each cluster k. Similarly to the original VLAD descriptor, the NetVLAD layer aggregates the first order statistics of residuals $(\mathbf{x}_i - \mathbf{c}_k)$ in different parts of the descriptor space weighted by the soft-assignment $\bar{a}_k(\mathbf{x}_i)$ of descriptor $\mathbf{x}_i$ to cluster k. Note however, that the NetVLAD layer has three independent sets of parameters $\{\mathbf{w}_k\}$, $\{b_k\}$ and $\{\mathbf{c}_k\}$, compared to just $\{\mathbf{c}_k\}$ of the original VLAD. This enables greater flexibility than the original VLAD, as explained in figure 3. For example, decoupling $\{\mathbf{w}_k, b_k\}$ from $\{\mathbf{c}_k\}$ has been proposed in [3] as a means to adapt the VLAD to a new dataset. All parameters of NetVLAD are learnt for the specific task in an end-to-end manner.

As illustrated in figure 2 the NetVLAD layer can be visualized as a meta-layer that is further decomposed into basic CNN layers connected up in a directed acyclic graph. First, note that the first term in eq. (4) is a soft-max function $\sigma_k(\mathbf{z}) = \frac{\exp(z_k)}{\sum_{k'} \exp(z_{k'})}$. Therefore, the soft-assignment of the input array of descriptors $\mathbf{x}_i$ into K clusters can be seen as a two step process: (i) a convolution with a set of K filters $\{\mathbf{w}_k\}$ that have spatial support 1×1 and biases $\{b_k\}$, producing the output $s_k(\mathbf{x}_i) = \mathbf{w}_k^T \mathbf{x}_i + b_k$; (ii) the convolution output is then passed through the soft-max function σ_k to obtain the final soft-assignment $\bar{a}_k(\mathbf{x}_i)$ that weights the different terms in the aggregation layer that implements eq. (4). The output after normalization is a $(K \times D) \times 1$ descriptor.

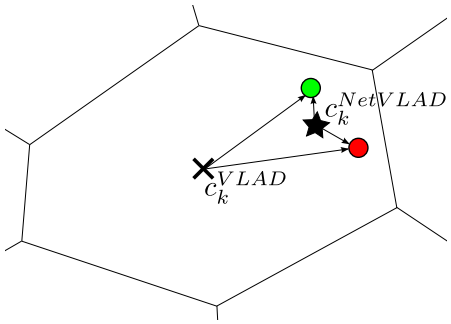


Figure 3. **Benefits of supervised VLAD.** Red and green circles are local descriptors from two different images, assigned to the same cluster (Voronoi cell). Under the VLAD encoding, their contribution to the similarity score between the two images is the scalar product (as final VLAD vectors are L2-normalized) between the corresponding residuals, where a residual vector is computed as the difference between the descriptor and the cluster's anchor point. The anchor point $\mathbf{c}_k$ can be interpreted as the origin of a new coordinate system local to the the specific cluster k. In standard VLAD, the anchor is chosen as the cluster centre ($\times$) in order to evenly distribute the residuals across the database. However, in a supervised setting where the two descriptors are known to belong to images which should not match, it is possible to learn a better anchor ($\star$) which causes the scalar product between the new residuals to be small.

Relations to other methods. Other works have proposed to pool CNN activations using VLAD or Fisher Vectors (FV) [13, 22], but do not learn the VLAD/FV parameters nor the input descriptors. The most related method to ours is the one of Sydorov *et al*. [75], which proposes to learn FV parameters jointly with an SVM for the end classification objective. However, in their work it is not possible to learn the input descriptors as they are hand-engineered (SIFT), while our VLAD layer is easily pluggable into any CNN architecture as it is amenable to backpropagation. Results

(a) (b) (c)

Figure 4. **Google Street View Time Machine examples.** Each column shows perspective images generated from panoramas from nearby locations, taken at different times. A well designed method can use this source of imagery to learn to be invariant to changes in viewpoint and lighting (a-c), and to moderate occlusions (b). It can also learn to suppress confusing visual information such as clouds (a), vehicles and people (b-c), and to chose to either ignore vegetation or to learn a season-invariant vegetation representation (a-c). More examples are given in appendix B.

(section 5.2) will show this difference to make a large impact on performance. Finally, [71] propose "Fisher Networks" where Fisher Vector layers are stacked on top of each other, but they do not train the system end-to-end and instead use hand-crafted features and train the layers greedily in a bottom-up fashion.

Max pooling (f_{max}). We also experiment with Max-pooling of the D-dimensional features across the spatial locations, thus producing a D-dimensional output vector, which is then L2-normalized. Both of these operations can be implemented using standard layers in public CNN packages. This setup mirrors the method of [6, 61], but a crucial difference is that we will learn the representation (section 4) while [6, 59, 61] only use pretrained networks. Results will show (section 5.2) that simply using CNNs off-the-shelf [59] results in poor performance, and that training for the end-task is crucial. Additionally, VLAD will prove itself to be superior to the Max-pooling baseline.

4. Learning from Time Machine data

In the previous section we have designed a new CNN architecture as an image representation for place recognition. Here we describe how to learn its parameters in an end-to-end manner for the place recognition task. The two main challenges are (i) how to gather enough annotated training data and (ii) what is the appropriate loss for the place recognition task. To address theses issues, we will first show that it is possible to obtain large amounts of weakly labelled imagery depicting the same places over time from the Google Street Time Machine. Second, we will design a new weakly supervised triplet ranking loss that can deal with the incomplete and noisy position annotations of the Street View Time Machine imagery. The details are below.

Weak supervision from the Time Machine. We propose to exploit a new source of data – Google Street View Time Machine – which provides multiple street-level panoramic images taken at different times at close-by spatial locations on the map. As it will be seen in section 5.2, this novel data source is precious for learning an image representation for place recognition. The same locations are depicted at different times and seasons, providing the learning algorithm with crucial information it can use to discover which features are useful or distracting, and what changes should the image representation be invariant to, in order to achieve good place recognition performance. For example, as illustrated in figure 4, the learnt image representation should be invariant to viewpoint and lighting changes, as well as partial occlusions. In addition, the learn representation could capture that cars, people and potentially vegetation should be ignored, but that buildings are very discriminative.

The downside of the Time Machine imagery is that it provides only incomplete and noisy supervision. Each Time Machine panorama comes with a GPS tag giving only its approximate location on the map, which can be used to identify close-by panoramas but does not provide correspondences between parts of the depicted scenes. In detail, as the test queries are perspective images from camera phones, each panorama is represented by a set of perspective images sampled evenly in different orientations and two elevation angles [10, 24, 35, 80]. Each perspective image is labelled with the GPS position of the source panorama. As a result, two geographically close perspective images do not necessarily depict the same objects since they could be facing different directions or occlusions could take place (*e.g.* the two images are around a corner from each other), *etc*. Therefore, for a given training query q, the GPS information can only be used as a source of (i) *potential* positives $\{p_i^q\}$, *i.e.* images that are geographically close to the query, and (ii) *definite* negatives $\{n_j^q\}$, *i.e.* images that are geographically far from the query.[1]

Weakly supervised triplet ranking loss. We wish to learn a representation f_θ that will optimize place recognition performance. That is, for a given test query image q, the goal is to rank a database image I_{i*} from a close-by location higher than all other far away images I_i in the database. In other words, we wish the Euclidean distance $d_\theta(q, I)$ be-

[1] Note that even faraway images can depict the same object. For example, the Eiffel Tower can be visible from two faraway locations in Paris. But, for the purpose of localization we consider in this paper such image pairs as negative examples because they are not taken from the same place.

tween the query q and a close-by image I_{i*} to be smaller than the distance to far away images in the database I_i, *i.e.* $d_\theta(q, I_{i*}) < d_\theta(q, I_i)$, for all images I_i further than a certain distance from the query on the map. Next we show how this requirement can be translated into a ranking loss between training triplets $\{q, I_{i*}, I_i\}$.

From the Google Street View Time Machine data, we obtain a training dataset of tuples $(q, \{p_i^q\}, \{n_j^q\})$, where for each training query image q we have a set of potential positives $\{p_i^q\}$ and the set of definite negatives $\{n_j^q\}$. The set of potential positives contains *at least one* positive image that should match the query, but we do not know which one. To address this ambiguity, we propose to identify the best matching potential positive image p_{i*}^q

$$p_{i*}^q = \underset{p_i^q}{\operatorname{argmin}}\, d_\theta(q, p_i^q) \tag{5}$$

for each training tuple $(q, \{p_i^q\}, \{n_j^q\})$. The goal then becomes to learn an image representation f_θ so that distance $d_\theta(q, p_{i*}^q)$ between the training query q and the best matching potential positive p_{i*}^q is smaller than the distance $d_\theta(q, n_j^q)$ between the query q and *all* negative images q_j:

$$d_\theta(q, p_{i*}^q) < d_\theta(q, n_j^q), \quad \forall j. \tag{6}$$

Based on this intuition we define a *weakly supervised ranking loss* L_θ for a training tuple $(q, \{p_i^q\}, \{n_j^q\})$ as

$$L_\theta = \sum_j l\left(\min_i d_\theta^2(q, p_i^q) + m - d_\theta^2(q, n_j^q)\right), \tag{7}$$

where l is the hinge loss $l(x) = \max(x, 0)$, and m is a constant parameter giving the margin. Note that equation (7) is a sum of individual losses for negative images n_j^q. For each negative, the loss l is zero if the distance between the query and the negative is greater by a margin than the distance between the query and the best matching positive. Conversely, if the margin between the distance to the negative image and to the best matching positive is violated, the loss is proportional to the amount of violation. Note that the above loss is related to the commonly used triplet loss [66, 67, 84, 85], but adapted to our weakly supervised scenario using a formulation (given by equation (5)) similar to multiple instance learning [20, 36, 83].

We train the parameters θ of the representation f_θ using Stochastic Gradient Descent (SGD) on a large set of training tuples from Time Machine data. Details of the training procedure are given in appendix E.

5. Experiments

In this section we describe the used datasets and evaluation methodology (section 5.1), and give quantitative (section 5.2) and qualitative (section 5.3) results to validate our approach. Finally, we also test the method on the standard image retrieval benchmarks (section 5.4).

5.1. Datasets and evaluation methodology

We report results on two publicly available datasets.

Pittsburgh (Pitts250k) [80] contains 250k database images downloaded from Google Street View and 24k test queries generated from Street View but taken at different times, years apart. We divide this dataset into three roughly equal parts for training, validation and testing, each containing around 83k database images and 8k queries, where the division was done geographically to ensure the sets contain independent images. To facilitate faster training, for some experiments, a smaller subset (Pitts30k) is used, containing 10k database images in each of the train/val(idation)/test sets, which are also geographically disjoint.

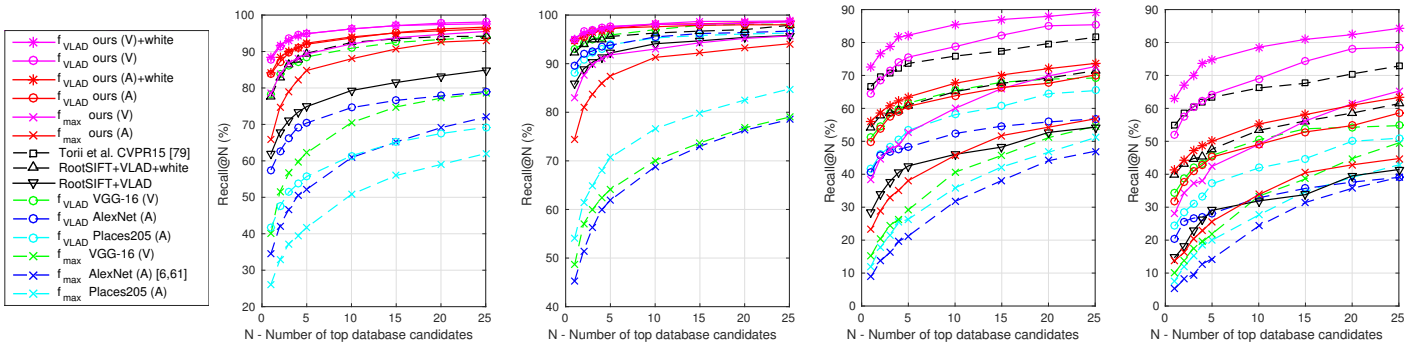
Tokyo 24/7 [79] contains 76k database images and 315 query images taken using mobile phone cameras. This is an extremely challenging dataset where the queries were taken at daytime, sunset and night, while the database images were only taken at daytime as they originate from Google Street View as described above. To form the train/val sets we collected additional Google Street View panoramas of Tokyo using the Time Machine feature, and name this set **TokyoTM**; Tokyo 24/7 (=test) and TokyoTM train/val are all geographically disjoint. Further details on the splits are given in appendix A.

Evaluation metric. We follow the standard place recognition evaluation procedure [4, 24, 64, 79, 80]. The query image is deemed correctly localized if at least one of the top N retrieved database images is within $d = 25$ meters from the ground truth position of the query. The percentage of correctly recognized queries (Recall) is then plotted for different values of N. For Tokyo 24/7 we follow [79] and perform spatial non-maximal suppression on ranked database images before evaluation.

Implementation details. We use two base architectures which are extended with Max pooling (f_{max}) and our NetVLAD (f_{VLAD}) layers: AlexNet [37] and VGG-16 [72]; both are cropped at the last convolutional layer (conv5), before ReLU. The initialization procedure, parameters used for training, procedure for sampling training tuples and other implementation details are given in appendix E. All training and evaluation code, as well as our trained networks, will be released online at [1].

5.2. Results and discussion

Baselines and state-of-the-art. To assess benefits of our approach we compare our representations trained for place recognition against "off-the-shelf" networks pretrained on other tasks. Namely, given a base network cropped at conv5, the baselines either use Max pooling (f_{max}), or aggregate the descriptors into VLAD (f_{VLAD}), but perform no further task-specific training. The three base networks are: AlexNet [37], VGG-16 [72], both are pretrained for



(a) Pitts250k-test (b) TokyoTM-val (c) Tokyo 24/7 all queries (d) Tokyo 24/7 sunset/night

Figure 5. **Comparison of our methods versus off-the-shelf networks and state-of-the-art.** The base CNN architecture is denoted in brackets: (A)lexNet and (V)GG-16. Trained representations (red and magenta for AlexNet and VGG-16) outperform by a large margin off-the-shelf ones (blue, cyan, green for AlexNet, Places205, VGG-16), f_{VLAD} (-o-) works better than f_{max} (-x-), and our f_{VLAD}+whitening (-$*$-) representation based on VGG-16 sets the state-of-the-art on all datasets. [79] only evaluated on Tokyo 24/7 as the method relies on depth data not available in other datasets. Additional results are shown in appendix C.

ImageNet classification [18], and Places205 [89], reusing the same architecture as AlexNet but pretrained for scene classification [89]. Pretrained networks have been recently used as off-the-shelf dense descriptor extractors for instance retrieval [6, 7, 22, 59, 61] and the untrained f_{max} network corresponds to the method of [6, 61].

Furthermore we compare our CNN representations trained for place recognition against the state-of-the-art local feature based compact descriptor, which consists of VLAD pooling [29] with intra-normalization [3] on top of densely extracted RootSIFTs [2, 42]. The descriptor is optionally reduced to 4096 dimensions using PCA (learnt on the training set) combined with whitening and L2-normalization [25]; this setup together with view synthesis yields the state-of-the-art results on the challenging Tokyo 24/7 dataset (*c.f.* [79]).

In the following we discuss figure 5, which compares place recognition performance of our method to the baselines outlined above on the Pittsburgh and Tokyo 24/7 benchmarks.

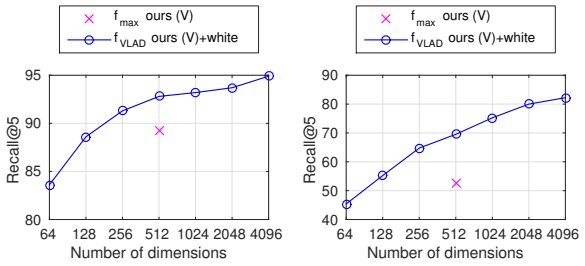
Dimensionality reduction. We follow the standard state-of-the-art procedure to perform dimensionality reduction of VLAD, as described earlier, *i.e.* the reduction into 4096-D is performed using PCA with whitening followed by L2-normalization [25, 79]. Figure 5 shows that the lower dimensional f_{VLAD} performs at least as well as the vector of full dimensionality, while giving a large boost on Tokyo 24/7.

Benefits of end-to-end training for place recognition. Representations trained on the end-task of place recognition consistently outperform by a large margin off-the-shelf CNNs on both benchmarks. For example, on the Pitts250k-test our trained AlexNet with (trained) NetVLAD aggregation layer achieves recall@1 of 83.9% compared to only 57.5% obtained by off-the-shelf AlexNet with standard VLAD aggregation, *i.e.* a relative improvement in recall of 46%. Similar improvements can be observed on all three datasets. This confirms two important premises of this work: (i) our approach can learn rich yet compact image representations for place recognition, and (ii) the popular idea of using pretrained networks "off-the-shelf" [6, 7, 22, 59, 61] is sub-optimal as the networks trained for object or scene classification are not necessary suitable for the end-task of place recognition.

The failure of the "off-the-shelf networks" is not surprising – apart from the obvious benefits of training, it is not clear why it would be meaningful to directly compare conv5 activations using Euclidean distance as they are trained to be part of the whole network architecture. For example, one can insert an arbitrary affine transformation of the features that can be countered exactly by the following fully connected layer (fc6). This is not a problem when transferring the pre-trained representation for object classification [48, 87] or detection [21] tasks, as the transformation can be countered by follow-up adaptation [48] or classification [21, 87] layers that are trained for the target task. However, this is not the case for retrieval [6, 7, 22, 59, 61] when Euclidean distance is used directly on the "off-the-shelf" descriptors.

Comparison with state-of-the-art. Figure 5 also shows that our trained f_{VLAD} representation with whitening based on VGG-16 (magenta -$*$-) convincingly outperforms RootSIFT+VLAD+whitening, as well as the method of Torii *et al.* [79], and therefore sets the state-of-the-art for compact descriptors on all benchmarks. Note that these are strong baselines that outperform most off-the-shelf CNN descriptors on the place recognition task.

VLAD versus Max. By comparing f_{VLAD} (-o-) methods with their corresponding f_{max} (-x-) counterparts it is clear that VLAD pooling is much better than Max pooling for both off-the-shelf and trained representations. Figure 6 shows that NetVLAD performance decreases grace-



(a) Pitts250k-test (b) Tokyo 24/7

Figure 6. **Place recognition accuracy versus dimensionality.** Note the log scale of the x-axis. 128-D NetVLAD performs comparably to the $4\times$ larger 512-D f_{max}. For the same dimension (512-D) NetVLAD convincingly outperforms f_{max}.

fully with dimensionality: 128-D NetVLAD performs similarly to Max (512-D), resulting in *four* times more compact representation for the same performance. Furthermore, NetVLAD+whitening outperforms Max pooling convincingly when reduced to the same dimensionality (512-D).

Max versus Sum. Recent work [7] suggests that Sum pooling performs better than Max pooling. Indeed, in our experiments Sum outperforms Max in the off-the-shelf set-up (recall@5 on Pitts250k-test – Sum: 67.8%, Max: 62.2%), but only for VGG-16, not AlexNet. Our training also works for Sum getting a significant improvement over the off-the-shelf set-up (+22% relative), but after training Max still performs better than Sum (Max: 89.3%, Sum: 82.7%).

Which layers should be trained? In Table 1 we study the benefits of training different layers for the end-task of place recognition. The largest improvements are thanks to training the conv5 layer, but training other layers results in further improvements, with some overfitting occurring below conv2. It is interesting to comment on these results in the light of the work by Sydorov *et al*. [75], where Fisher Vector parameters are learnt jointly with an SVM for the end classification objective, but without an ability to learn better descriptors (SIFT is used). Their setting is similar to our f_{VLAD} method where backpropagation is restricted only down to the NetVLAD layer. Table 1 shows that this approach does give a boost of 4-8%, but learning the weights of the layers below together with the NetVLAD layer (which is not possible in [75]) pushes the recall by additional 17-27%.

Importance of Time Machine training. Here examine whether the network can be trained without the Time Machine data. In detail, we have modified the training query set for Pitts30k-train to be sampled from the same set as the training database images, *i.e*. the tuples of query and database images used in training were captured at the same time. Table 2 shows that Time Machine (TM) data is crucial for good place recognition accuracy as without it the network does not generalize well. The network learns, for example, that recognizing cars is important for place recognition, as the same parked cars appear in all images of a place.

Lowest trained layer	f_{max} r@1	r@5	r@10	f_{VLAD} r@1	r@5	r@10
none (off-the-shelf)	33.5	57.3	68.4	54.5	69.8	76.1
NetVLAD	—	—	—	58.1	76.3	80.1
conv5	63.8	83.8	89.0	83.9	93.4	95.6
conv4	62.1	83.6	89.2	84.5	94.5	95.9
conv3	**69.8**	86.7	90.3	85.0	**94.6**	**97.1**
conv2	69.1	**87.6**	**91.5**	**85.6**	93.7	95.4
conv1 (full)	68.5	86.2	90.8	83.2	94.2	95.5

Table 1. **Partial training.** Effects of performing backpropagation only down to a certain layer of AlexNet, *e.g*. 'conv4' means that weights of layers from conv4 and above are learnt, while weights of layers below conv4 are fixed to their pretrained state; r@N signifies recall@N. Results are shown on the Pitts30k-val dataset.

Training data	recall@1	recall@10
Pretrained on ImageNet [37]	33.5	68.5
Pretrained on Places205 [89]	24.8	54.4
Trained without Time Machine	38.7	68.1
Trained with Time Machine	**68.5**	**90.8**

Table 2. **Time Machine importance.** Recall of f_{max} on Pitts30k-val (AlexNet) with vs without using Time Machine data for training. Training using Time Machine is essential for generalization.

Generalization. A natural question to ask is whether the learnt image representations are generic, suitable for place recognition across cities, or whether a representation has to be trained for each given city. The full set of results for this experiment are in appendix D. In summary, we have observed some loss in performance, for example, f_{max} AlexNet trained on Pittsburgh achieves 68.7% recall@1 on Tokyo (compared to 79.0% for Tokyo trained network). This is presumably because the two cities are quite different, *e.g*. the Pittsburgh network wouldn't have seen any Japanese characters. However, the network trained on Pittsburgh does recognize Tokyo places much better than off-the-shelf networks trained for other tasks (AlexNet 45.3%, Places205 54.0%).

5.3. Qualitative evaluation

To visualize what is being learnt by our place recognition architectures, we adapt the method of Zeiler and Fergus [87] for examining occlusion sensitivity of classification networks. In particular, we measure the distance between the representation $f(I)$ of the original image I, and the representation $f(I_{o(x,y)})$ of the image $I_{o(x,y)}$, generated by occluding I with a grey patch at position (x, y). Examples are shown in figure 7. Patches which, when occluded, yield large changes in the image representation (compared to the the non-occluded image) are shown in dark red colours. It can be seen that off-the-shelf AlexNet (pretrained on Ima-

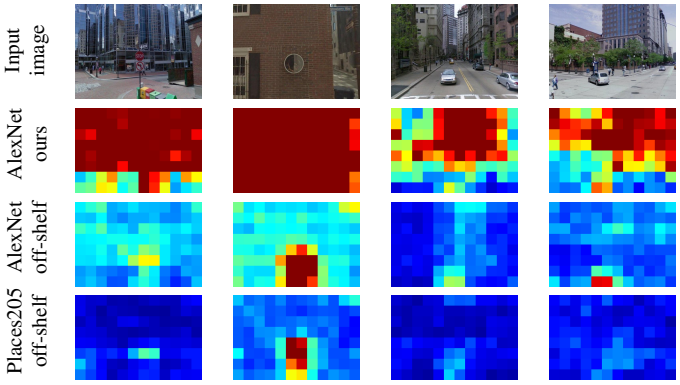


Figure 7. **What has been learnt?** Each column corresponds to one image (top row) and the emphasis various networks (under f_{max}) give to different patches. Each pixel in the heatmap corresponds to the change in representation when a large gray occluding square (100×100) is placed over the image in the same position (*c.f.* section 5.3); all heatmaps have the same colour scale. Note that the original image and the heatmaps are not in perfect alignment as nearby patches overlap 50% and patches touching an image edge are discarded to prevent border effects. All images are from Pitts250k-val that the network hasn't seen at training. Further examples are given in the supplementary material.

geNet) focuses very much on categories it has been trained to recognize, such as cars and certain shapes, such as circular blobs useful for distinguishing 12 different ball types in the ImageNet categories. The Place205 network is fairly unresponsive to all occlusions as it does not aim to recognize specific places but scene-level categories, so even if an important part of the image is occluded, such as a characteristic part of a building façade, it still provides a similar output feature which corresponds to an uninformative "a building façade" image descriptor. In contrast to these two, our network trained for specific place recognition automatically learns to ignore confusing features, such as cars and people, which are not discriminative for specific locations, and instead focuses on describing building façades and skylines. More qualitative examples are provided in appendix C.

5.4. Image retrieval

We use our best performing network (VGG-16, f_{VLAD} with whitening down to 256-D) trained completely on Pittsburgh, to extract image representations for standard object and image retrieval benchmarks (Oxford 5k [52], Paris 6k [53], Holidays [26]). Table 3 compares NetVLAD to the state-of-the-art compact image representations (256-D). Our representation achieves the best mAP on Oxford and Paris by a large margin, *e.g.* +19% relative improvement on Oxford 5k (crop). It also sets the state-of-the-art on Holidays, but here training is detrimental as the dataset is less building-oriented (*e.g.* it also contains paysages, underwater photos, boats, cars, bears, *etc.*), while our training only sees images from urban areas. We believe training on data more diverse than Pittsburgh streets can further improve performance.

6. Conclusions

We have designed a new convolutional neural network architecture that is trained for place recognition in an end-to-end manner from weakly supervised street-view time machine data. Our trained representation significantly outperforms off-the-shelf CNN models and significantly improves over the state-of-the-art on the challenging 24/7 Tokyo dataset, as well as on the Oxford and Paris image retrieval benchmarks. The two main components of our architecture – (i) the NetVLAD pooling layer and (ii) weakly supervised ranking loss – are generic CNN building blocks applicable beyond the place recognition task. The NetVLAD layer offers a powerful pooling mechanism with learnable parameters that can be easily plugged into any other CNN architecture. The weakly supervised ranking loss opens up the possibility of end-to-end learning for other ranking tasks where large amounts of weakly labelled data are available, for example, images described with natural language [33].

Acknowledgements. This work was partly supported by EU FP7-SPACE-2012-312377 PRoViDE, the ERC grant LEAP (no. 336845), ANR project Semapolis (ANR-13-CORD-0003), JSPS KAKENHI Grant Number 15H05313, the Inria CityLab IPL, and the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory, contract FA8650-12-C-7212. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

References

[1] http://www.di.ens.fr/willow/research/netvlad/. 6, 13
[2] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012. 2, 7
[3] R. Arandjelović and A. Zisserman. All about VLAD. In *Proc. CVPR*, 2013. 1, 2, 3, 4, 7
[4] R. Arandjelović and A. Zisserman. DisLocation: Scalable descriptor distinctiveness for location recognition. In *Proc. ACCV*, 2014. 1, 2, 6
[5] M. Aubry, B. C. Russell, and J. Sivic. Painting-to-3D model alignment via discriminative visual elements. *ACM Transactions on Graphics (TOG)*, 33(2):14, 2014. 1
[6] H. Azizpour, A. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Factors of transferability from a generic ConvNet representation. *CoRR*, abs/1406.5774, 2014. 2, 3, 5, 7
[7] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *Proc. ICCV*. 2, 3, 7, 8, 10
[8] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *Proc. ECCV*, 2014. 2, 10

Method	Oxford 5k (full)	Oxford 5k (crop)	Paris 6k (full)	Paris 6k (crop)	Holidays (orig)	Holidays (rot)
Jégou and Zisserman [32]	–	47.2	–	–	65.7	65.7
Gordo *et al*. [23]	–	–	–	–	78.3	–
Razavian [61]	53.3†	–	67.0†	–	74.2	–
Babenko and Lempitsky [7]	58.9	53.1	–	–	–	80.2
a. Ours: NetVLAD off-the-shelf	53.4	55.5	64.3	67.7	**82.1**	**86.0**
b. Ours: NetVLAD trained	**62.8**	**63.4**	**72.5**	**71.5**	76.8	80.7

Table 3. **Comparison with state-of-the-art compact image representations (256-D) on image and object retrieval.** We compare (b.) our best trained network, (a.) the corresponding off-the-shelf network (whitening learnt on Pittsburgh), and the state-of-the-art for compact image representations on standard image and object retrieval benchmarks. "orig" and "rot" for Holidays denote whether the original or the manually rotated dataset [7, 8] is used. The "crop" and "full" for Oxford/Paris correspond to the testing procedures when the query ROI is respected (the image is cropped as in [7]), or ignored (the full image is used as the query), respectively. † [61] use square patches whose side is equal to $1.5\times$ the maximal dimension of the query ROI (the detail is available in version 2 of the arXiv paper [60]), so the setting is somewhere in between "crop" and "full", arguably closer to "full" as ROIs become very large.

[9] S. Cao and N. Snavely. Graph-based discriminative learning for location recognition. In *Proc. CVPR*, 2013. 1, 2

[10] D. M. Chen, G. Baatz, K. Koeser, S. S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *Proc. CVPR*, 2011. 1, 2, 5, 12

[11] O. Chum, A. Mikulik, M. Perďoch, and J. Matas. Total recall II: Query expansion revisited. In *Proc. CVPR*, 2011. 2

[12] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. ICCV*, 2007. 2

[13] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proc. CVPR*, 2015. 3, 4

[14] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004. 3

[15] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 2008. 1, 2

[16] M. Cummins and P. Newman. Highly scalable appearance-only SLAM - FAB-MAP 2.0. In *RSS*, 2009. 1, 2

[17] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez. Revisiting the VLAD image representation. In *Proc. ACMM*, 2013. 2

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 7

[19] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013. 2

[20] J. Foulds and E. Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(01):1–25, 2010. 6

[21] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014. 1, 2, 7

[22] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Proc. ECCV*, 2014. 2, 3, 4, 7

[23] A. Gordo, J. A. Rodríguez-Serrano, F. Perronnin, and E. Valveny. Leveraging category-level labels for instance-level image retrieval. In *Proc. CVPR*, pages 3045–3052, 2012. 10

[24] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *Proc. CVPR*, 2013. 1, 2, 5, 6, 12

[25] H. Jégou and O. Chum. Negative evidences and cooccurrences in image retrieval: the benefit of PCA and whitening. In *Proc. ECCV*, 2012. 2, 7

[26] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. ECCV*, pages 304–317, 2008. 2, 9

[27] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *Proc. CVPR*, Jun 2009. 2

[28] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE PAMI*, 2011. 1

[29] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010. 1, 2, 3, 7

[30] H. Jégou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *Proc. CVPR*, 2007. 2

[31] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local images descriptors into compact codes. *IEEE PAMI*, 2012. 1

[32] H. Jégou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. In *Proc. CVPR*, 2014. 2, 10

[33] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proc. CVPR*, 2015. 9

[34] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *Proc. ICCV*, 2015. 2

[35] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *Proc. ECCV*, 2010. 1, 2, 5, 12

[36] D. Kotzias, M. Denil, P. Blunsom, and N. de Freitas. Deep multi-instance transfer learning. *CoRR*, abs/1411.3128, 2014. 6

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 1, 2, 6, 8, 12

[38] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation

applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. 1

[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[40] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3D point clouds. In *Proc. ECCV*, 2012. 1

[41] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. In *Proc. CVPR*, 2015. 2

[42] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 3, 7

[43] W. Maddern and S. Vidas. Towards robust night and day place recognition using visible and thermal imaging. In *Proc. Intl. Conf. on Robotics and Automation*, 2014. 1, 2

[44] A. Makadia. Feature tracking for wide-baseline image retrieval. In *Proc. ECCV*, 2010. 2

[45] C. McManus, W. Churchill, W. Maddern, A. Stewart, and P. Newman. Shady dealings: Robust, long-term visual localisation using illumination invariance. In *Proc. Intl. Conf. on Robotics and Automation*, 2014. 1, 2

[46] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-DOF localization on mobile devices. In *Proc. ECCV*, 2014. 1

[47] A. Mikulik, M. Perďoch, O. Chum, and J. Matas. Learning a fine vocabulary. In *Proc. ECCV*, 2010. 2

[48] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proc. CVPR*, 2014. 1, 2, 7

[49] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronnin, and C. Schmid. Local convolutional features with unsupervised training for image retrieval. In *Proc. ICCV*. 2

[50] F. Perronnin and D. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. CVPR*, 2007. 2

[51] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *Proc. CVPR*, 2010. 1, 2

[52] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007. 1, 2, 9

[53] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. CVPR*, 2008. 2, 9

[54] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *Proc. ECCV*, 2010. 2

[55] D. Qin, X. Chen, M. Guillaumin, and L. V. Gool. Quantized kernel learning for feature matching. In *NIPS*, 2014. 2

[56] D. Qin, Y. Chen, M. Guillaumin, and L. V. Gool. Learning to rank bag-of-word histograms for large-scale object retrieval. In *Proc. BMVC.*, 2014. 2

[57] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. In *Proc. CVPR*, 2011. 2

[58] D. Qin, C. Wengert, and L. V. Gool. Query adaptive similarity for large scale object retrieval. In *Proc. CVPR*, 2013. 2

[59] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014. 2, 5, 7

[60] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. A baseline for visual instance retrieval with deep convolutional networks. *CoRR*, abs/1412.6574v2, 2014. 10

[61] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. A baseline for visual instance retrieval with deep convolutional networks. In *Proc. ICLR*, 2015. 2, 3, 5, 7, 10

[62] T. Sattler, M. Havlena, F. Radenović, K. Schindler, and M. Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proc. ICCV*. 1, 2

[63] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2D–to–3D matching. In *Proc. ICCV*, 2011. 1, 2

[64] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *Proc. BMVC.*, 2012. 1, 2, 6

[65] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Proc. CVPR*, 2007. 1, 2

[66] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proc. CVPR*, 2015. 6

[67] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2004. 6

[68] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 2

[69] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, and F. Moreno-Noguer. Fracking deep convolutional image descriptors. *CoRR*, abs/1412.6537, 2014. 2

[70] K. Simonyan, A. Vedaldi, and A. Zisserman. Descriptor learning using convex optimisation. In *Proc. ECCV*, 2012. 2

[71] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Fisher networks for large-scale image classification. In *NIPS*, 2013. 5

[72] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 1, 6, 12

[73] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, volume 2, pages 1470–1477, 2003. 1, 3

[74] N. Sunderhauf, S. Shirazi, A. Jacobson, E. Pepperell, F. Dayoub, B. Upcroft, and M. Milford. Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free. In *Robotics: Science and Systems*, 2015. 1, 2

[75] V. Sydorov, M. Sakurada, and C. Lampert. Deep fisher kernels – end to end learning of the fisher kernel GMM parameters. In *Proc. CVPR*, 2014. 4, 8

[76] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2014. 1

[77] G. Tolias, Y. Avrithis, and H. Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *Proc. ICCV*, 2013. 2

[78] G. Tolias and H. Jégou. Visual query expansion with or without geometry: refining local descriptors by feature aggregation. *Pattern Recognition*, 2014. 2

[79] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *Proc.*

CVPR, 2015. 1, 2, 6, 7, 12, 14
[80] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual place recognition with repetitive structures. In *Proc. CVPR*, 2013. 1, 2, 5, 6, 12
[81] T. Turcot and D. G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshop on Emergent Issues in Large Amounts of Visual Data (WS-LAVD)*, 2009. 2
[82] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008. 1
[83] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS*, 2005. 6
[84] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *Proc. CVPR*, 2014. 6
[85] K. Q. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2006. 6
[86] S. Winder, G. Hua, and M. Brown. Picking the best DAISY. In *Proc. CVPR*, pages 178–185, 2009. 2
[87] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. ECCV*, 2014. 2, 7, 8
[88] J. Zepeda and P. Pérez. Exemplar SVMs as visual feature encoders. In *Proc. CVPR*, 2015. 2
[89] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 1, 7, 8, 12

A. Details of datasets

Table 4 shows the statistics of datasets used in this work, explained in section 5.1 of the main paper.

The newly collected Tokyo Time Machine (TokyoTM) database was generated from downloaded Time Machine panoramas, such that each panorama is represented by a set of 12 perspective images sampled evenly in different orientations [10, 24, 35, 79, 80]. For each query, the positive and negative sets are sampled from the database so that they have a time stamp at least one month away from the time stamp of the query. This is done for both training (training/val sets) and evaluation (testing set). All datasets are publicly available: Pitts250k from the authors of [80], Tokyo 24/7 from the authors of [79], while we will share TokyoTM and Pitts30k on request.

B. Google Street View Time Machine imagery

Figure 8 shows more examples of Google Street View Time Machine imagery used for training CNN-based representations in this paper (section 4 of the main paper).

C. Additional results

Figure 9 reports a complete set of results that did not fit into figure 5 of the main paper. Namely, it includes results on the Pitts30k-test and the complete breakdown of day versus sunset/night queries for the Tokyo 24/7 benchmark as done in [79]. Figure 10 shows additional visualizations of what has been learnt by our method. Please see section 5.3 of the main paper for the details of the visualization. Figures 11, 12 and 13 compare the top ranked images of our method versus the best baseline.

Dataset	Database size	Query Size
Pitts250k-train	91,464	7,824
Pitts250k-val	78,648	7,608
Pitts250k-test	83,952	8,280
Pitts30k-train	10,000	7,416
Pitts30k-val	10,000	7,608
Pitts30k-test	10,000	6,816
Tokyo Time Machine-train	49,104	7,277
Tokyo Time Machine-val	49,056	7,186
Tokyo 24/7 (-test)	75,984	315

Table 4. **Datasets.** Statistics of the datasets used in experiments. All train/val(idation)/test datasets are mutually disjoint geographically.

Training data	recall@1	recall@10
a. Off-the-shelf ImageNet [37]	45.3	68.8
b. Off-the-shelf Places205 [89]	54.0	76.6
c. Ours Pittsburgh	68.7	85.0
d. Ours TokyoTM	79.0	92.0

Table 5. **Generalization.** Testing place recognition on Tokyo Time Machine data (TokyoTM-val) using CNN image representations trained on different datasets. All tests are performed using the f_{max} model (see the main paper). Note that training on a completely unrelated city (c. Pittsburgh) results in some loss in performance (compared to training on the same city as in d.) but still performs much better than off-the-shelf models (a. and b.) trained for object/scene classification. Training on Pittsburgh was done on the Pitts250k-train dataset. Note that d. Ours TokyoTM was trained on the same city as the test set (Tokyo) but the the training area is geographically distinct from the test area.

D. Generalization

Here we investigate whether a representation trained on one city generalizes well to another completely unrelated city. A subset of the results is discussed in section 5.2 of the main paper. Here we include a complete set of results in Table 5.

E. Implementation details

We use two base architectures which are extended with Max pooling (f_{max}) and our NetVLAD (f_{VLAD}) layers: AlexNet [37] and VGG-16 [72]; both are cropped at the last convolutional layer (conv5), before ReLU. For Max we use raw conv5 descriptors (with no normalization) while for VLAD we add an additional descriptor-wise

(a) (b) (c) (d) (e) (f)

Figure 8. **Google Street View Time Machine examples.** Each column shows perspective images generated from panoramas from nearby locations, taken at different times. The goal of this work is to learn from this imagery an image representation that: has a degree of invariance to changes in viewpoint and illumination (a-f); tolerance to partial occlusions (c-f); suppresses confusing visual information such as clouds (a,c), vehicles (c-f) and people (c-f); and chooses to either ignore vegetation or learn a season-invariant vegetation representation (a-f).

L2-normalization layer before NetVLAD. We found this to work better for both architectures.

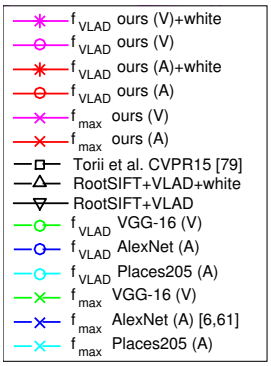
The number of clusters used in all VLAD / NetVLAD experiments is $K = 64$. The VLAD layer parameters are initialized to reproduce the conventional VLAD vectors (*c.f.* section 5.1 of the main paper) by clustering conv5 descriptors extracted from a subsample of the train set for each dataset. The α parameter used for initialization is chosen to be large, such that the soft assignment weights $\bar{a}_k(\mathbf{x}_i)$ are very sparse in order to mimic the conventional VLAD well. Specifically, α is computed so that the the ratio of the largest and the second largest soft assignment weight $\bar{a}_k(\mathbf{x}_i)$ is on average equal to 100.

We use the margin $m = 0.1$, learning rate 0.0001 (or 0.001 for the smaller Pitts30k dataset), which is halved every 5 epochs, momentum 0.9, weight decay 0.001, batch size of 4 tuples (a tuple contains many images, *c.f.* equation (7) of the main paper), and train for at most 30 epochs but convergence usually occurs much faster. The network which yields the best recall@5 on the validation set is used for testing.

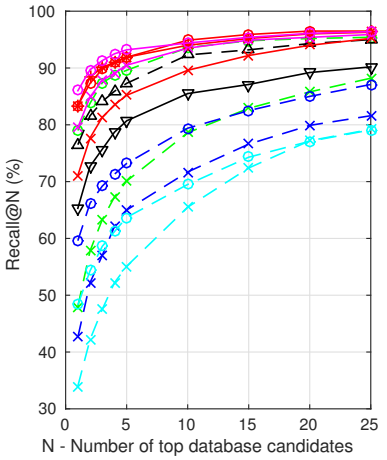
As the VGG-16 network is much deeper and more GPU-memory hungry than AlexNet, it was not possible to train it in its entirety. Instead, in the light of experiments in table 1 of the main paper, the VGG-16 network is only trained down to level conv5.

To create the training tuple for a query, we use all of its potential positives (images within 10 meters), and we perform randomized hard negative mining for the negatives (images further away than 25 meters). The mining is done by keeping the 10 hardest negatives from a pool of 1000 randomly sampled negatives and 10 hardest negatives from the previous epoch. We find that remembering previous hard negatives adds stability to the training process.

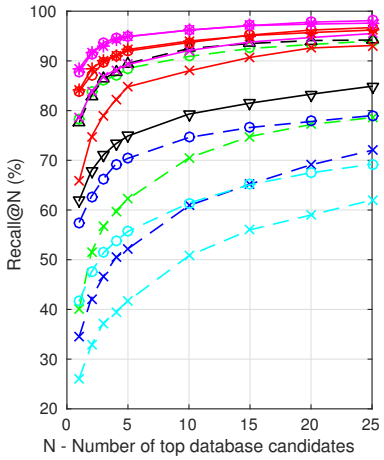
Naively implemented, the aforementioned training procedure would be too slow. Processing each training tuple would require a forward pass on more than 1010 full-resolution images. Instead, we compute image representations for the entire training query and database sets and cache them for a certain amount of time. The hard negative mining then uses these cached but slightly stale representations to obtain the 10 hardest examples and the forward and backward passes are only performed on these 10, compared to the original 1010, thus providing a huge computational saving. However, it is important to recompute the cached representations every once in a while. We have observed slow convergence if the cache is fixed for too long as the network learns quickly to be better than the fixed cache and then wastes time overfitting it. We found that recomputing the cached representations for hard negative mining every 500 to 1000 training queries yields a good trade-off between epoch duration, convergence speed and quality of the solution. As described earlier, we half the learning rate every 5 epochs – this causes the cached representations to change less rapidly, so we half the recomputation frequency every 5 epochs as well. All training and evaluation code, as well as our trained networks, will be released online at [1]. Additional tuning of parameters could further improve performance as we have still observed some amount of overfitting.



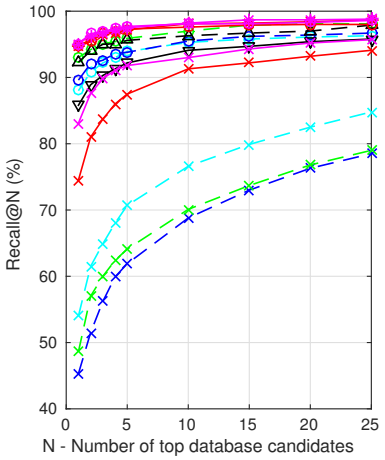
(a) Legend



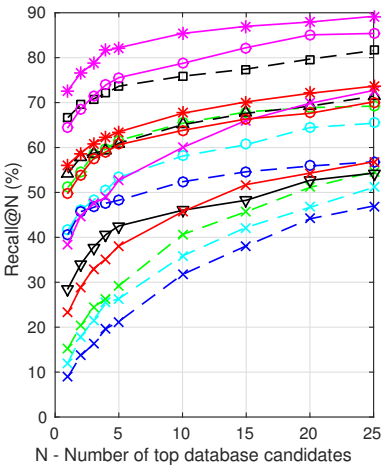
(b) Pitts30k-test



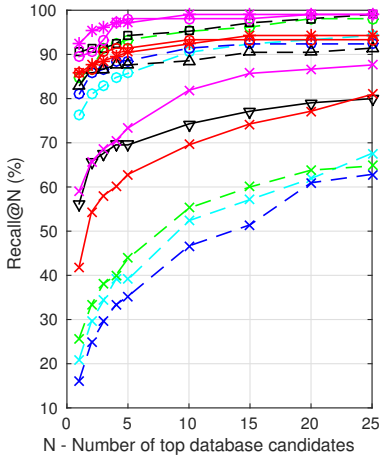
(c) Pitts250k-test



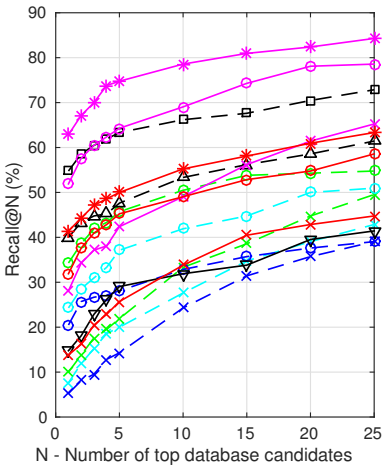
(d) TokyoTM-val



(e) Tokyo 24/7 all queries



(f) Tokyo 24/7 daytime



(g) Tokyo 24/7 sunset/night

Figure 9. **Comparison of our methods versus off-the-shelf networks and state-of-the-art.** The base CNN architecture is denoted in brackets: (A)lexNet and (V)GG-16. Trained representations (red and magenta for AlexNet and VGG-16) outperform by a large margin off-the-shelf ones (blue, cyan, green for AlexNet, Places205, VGG-16), f_{VLAD} (-o-) works better than f_{max} (-x-), and our f_{VLAD}+whitening (-$*$-) representation based on VGG-16 sets the state-of-the-art on all datasets. [79] only evaluated on Tokyo 24/7 as the method relies on depth data not available in other datasets.

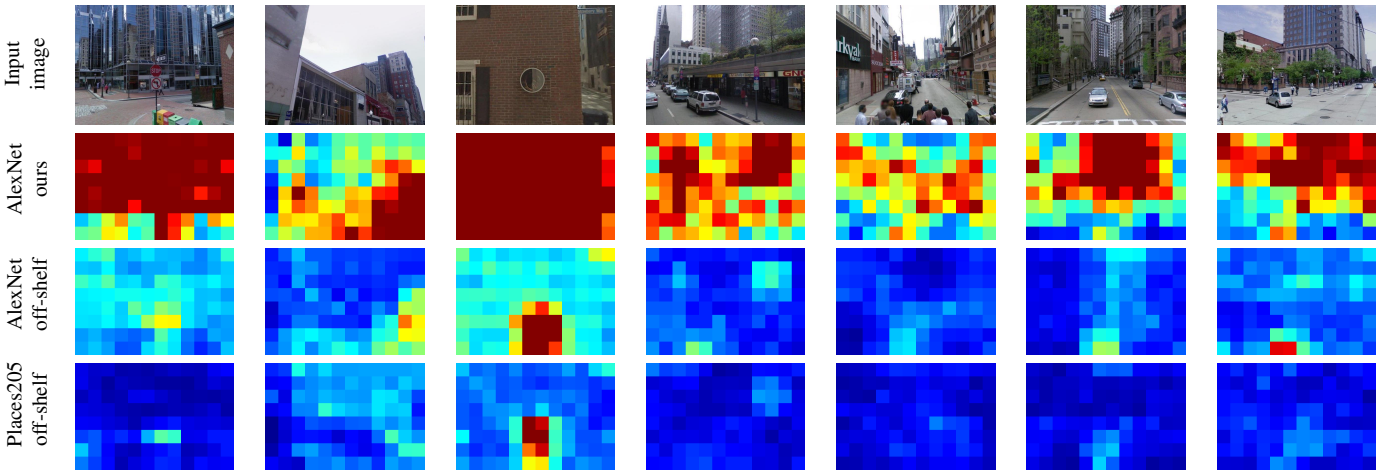


Figure 10. **What has been learnt?** Each column corresponds to one image (top row) and the emphasis various networks (under f_{max}) give to different patches. Each pixel in the heatmap corresponds to the change in representation when a large gray occluding square (100×100) is placed over the image in the same position (*c.f.* section 5.3); all heatmaps have the same colour scale. Note that the original image and the heatmaps are not in perfect alignment as nearby patches overlap 50% and patches touching an image edge are discarded to prevent border effects. All images are from Pitts250k-val that the network hasn't seen at training.

Figure 11. **Examples of retrieval results for challenging queries on Tokyo 24/7.** Each column corresponds to one test case: the query is shown in the first row, the top retrieved image using our best method (trained VGG-16 NetVLAD + whitening) in the second, and the top retrieved image using the best baseline (RootSIFT + VLAD + whitening) in the last row. The green and red borders correspond to positive and negative retrievals.

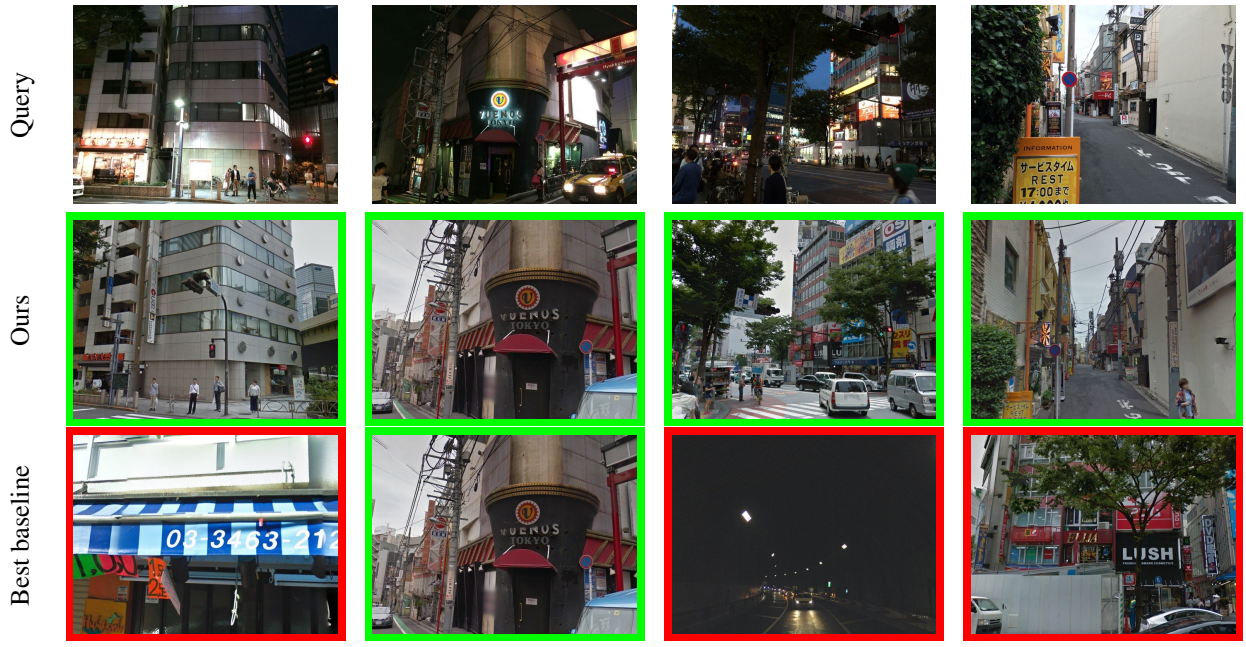


Figure 12. **Examples of retrieval results for challenging queries on Tokyo 24/7.** Each column corresponds to one test case: the query is shown in the first row, the top retrieved image using our best method (trained VGG-16 NetVLAD + whitening) in the second, and the top retrieved image using the best baseline (RootSIFT + VLAD + whitening) in the last row. The green and red borders correspond to positive and negative retrievals.

Figure 13. **Examples of retrieval results for challenging queries on Tokyo 24/7.** Each column corresponds to one test case: the query is shown in the first row, the top retrieved image using our best method (trained VGG-16 NetVLAD + whitening) in the second, and the top retrieved image using the best baseline (RootSIFT + VLAD + whitening) in the last row. The green and red borders correspond to positive and negative retrievals. The last column corresponds to a difficult query, which is hard for our method because of its darkness and indistinctiveness.