Chapter 3

Trainable image descriptors for place recognition

Abstract

Visual place recognition is the task of automatically recognizing a previously visited location through its appearance, and plays a key role in mobile robotics and autonomous driving applications. The difficulty of recognizing a revisited location increases with appearance variations caused by weather, illumination or point of view changes. In this paper we present a convolutional neural network (CNN) embedding to perform place recognition, even under severe appearance changes. The network maps images to a low dimensional space where images from nearby locations map to points close to each other, despite differences in visual appearance caused by the aforementioned phenomena. In order for the network to learn the desired invariances, we train it with triplets of images selected from datasets which present a challenging variability in visual appearance. Our proposal is validated through extensive experimentation that reveals better performance than state-of-the-art methods. Importantly, though the training phase is computationally demanding, its online application is very efficient.

3.1 Introduction

Visual place recognition is the process of identifying images that belong to the same location, and is still an open problem in computer vision. In fact, it is a challenging task when the images differ in appearance due to perspective, illumination or weather changes, or even the presence of objects which are not part of the static scene (i.e., cars on the street). Moreover, it is crucial in robotics as part of vision-based localization and mapping (SLAM) systems.
3. Trainable image descriptors for place recognition

and Tardós, 2014) and has also found use in indoor localization applications (Song and Park, 2015; Tomasi and Anedda, 2013; Werner et al., 2011). Traditionally, this problem has been addressed by employing bags of visual words (BoW) (Sivic and Zisserman, 2003; Nistér and Stewénius, 2006), which have proven to work quickly and effectively for many applications, but they have several drawbacks: They usually generate histograms obtained from keypoint descriptors, such as SIFT (Lowe, 2004), SURF (Bay et al., 2006), or BRIEF (Calonder et al., 2010), which describe the local appearance of individual patches, limiting their descriptive power with respect to holistic methods, as observed by Milford and Wyeth (2012). Thus, their performance in challenging environments strongly depends on the invariance of those descriptors to perceptual changes.

On the other hand, Convolutional Neural Networks (CNNs) are gaining importance in most classification tasks (Krizhevsky et al., 2012). When used as generic feature generators, they often outperform the state-of-art algorithms, even for other tasks such as visual instance retrieval (Razavian, Azizpour, Sullivan and Carlsson, 2014). Although their use in place recognition was initially limited to the exploitation of generic features extracted from the internal layers of networks trained for other tasks (Chen et al., 2014; Sunderhauf, Shirazi, Dayoub, Upcroft, Milford, Sunderhauf, Dayoub, Sareh, Ben and Michael, 2015), models specifically trained for the task of place recognition (Lopez-Antequera et al. 2017; Arandjelovic et al., 2016) have achieved state-of-art performance.

In this paper, we propose an approach to place recognition capable of detecting revisited places under changes in weather (Figure 3.1), point of view (Figure 3.2), or illumination (Figure 3.4). For this purpose, we have used a triplet-based learning scheme for training a CNN to embed images in a low dimensional space where small euclidean distances are representative of place similarity, allowing for efficient image summarization. We demonstrate that place recognition can be better resolved by discriminatively training a network for such a problem, as opposed to using hand-designed local features or feature vectors extracted from generic networks. Moreover, we claim and demonstrate that place recognition can be performed with a smaller and faster network than those employed for object recognition. Our network produces a single descriptor vector of 128 elements for each image, reducing storage requirements for long term operation, and hence it is suitable to run on portable computers, robots or smartphones in real time. We test our proposal on challenging datasets, outperforming several state-of-the-art methods used for loop closure and image retrieval, namely:

- DBoW2 (Mur-Artal et al., 2015)
- Raw features extracted from internal representations of AlexNet (Krizhevsky)
3.2. Related Work

Figure 3.1: Frames extracted from the Nordland dataset (Sunderhauf et al., 2013) taken at the same location in the four seasons. Our proposal can recognize the same location under such appearance changes.

et al., 2012), VggNet (Simonyan and Zisserman, 2014), and NetVLAD (Arandjelovic et al., 2016) as by Pepperell et al., (2014)

• Pooled features extracted by single and quadruple cell max pooling from of AlexNet, VggNet, as by Razavian, Sullivan, Maki and Carlsson (2014)

3.2 Related Work

Visual place recognition has been object of research in robotics as a key part of the localization and mapping systems. One of the first works, which introduced the use of Bag of Words (BoW) in this context, was FAB-MAP (Cummins and Newman, 2008), where a probabilistic approach to place recognition based on the appearance of each location was proposed. Later, (Galvez-Lopez and Tardos, 2012) introduced DBoW2, reducing in more than an order of magnitude the time employed by the feature extraction process, by using binary descriptors. The use of BRIEF, which is not rotation or scale invariant, limited the recognition task to scenes taken from the same viewpoint in planar trajectories. An improved version of this algorithm has recently been published by Mur-Artal et al., (2015), where the authors build a urban dictionary based on the ORB descriptor (Rublee et al., 2011) which yields a better
performance in popular datasets.

A common problem with previous techniques is their poor behavior in place recognition under strong appearance changes caused by different illumination or weather conditions and poorly textured environments. SeqSLAM [Milford and Wyeth, 2012] works on sequences instead of estimating the best single location. The authors propose a post-processing technique that improves precision and recall by aggregating evidence over image sequences. This aggregation over sequences is useful for any method of image description, including ours. However, the method presents several drawbacks derived from their image description, based on simply calculating the aggregated pixel differences of downscaled and contrast-normalized images: it only works with local and consistent sequences, which makes it impractical for applications that work with isolated images. It also may fail under big changes of viewpoint and rotation, since its viewpoint invariance is only provided by extreme downscaling of the input images.

Recently, another group of techniques have shown promising results, motivated by the outstanding performance achieved by CNNs as generic feature generators in several tasks [Razavian, Azizpour, Sullivan and Carlsson, 2014]. In this context, a recent work is [Sunderhauf, Shirazi, Dayoub, Upcroft, Milford, Sunderhauf, Dayoub, Sareh, Ben and Michael, 2015], where the authors use a pre-trained network named OverFeat [Sermanet et al., 2013]. They study the use of the intermediate representations learned by the CNN as image features valuable for place recognition even under challenging appearance changes. In a similar way, the work of [Razavian, Sullivan, Maki and Carlsson, 2014] achieves state-of-the-art results in image retrieval by performing max pooling over the whole feature maps extracted using generic convolutional nets.

In [Sunderhauf, Shirazi, Jacobson, Dayoub, Pepperell, Upcroft and Milford, 2015], authors presented an approach that uses generic CNN representations of independent landmarks within an image to increase invariance to occlusion and viewpoint changes. Global scores are computed after comparing all of the detected landmarks in each image pair, achieving good localization results, without specific training. Another line of work [Neubert and Protzel, 2015, 2016] proposed to combine local region detectors with CNN-based descriptors to harness the robustness against appearance changes of CNN descriptors with the robustness to viewpoint changes of local landmarks. In [Neubert and Protzel, 2016], the authors extended their previous work by also proposing a novel region detector, namely the multi-scale superpixel grid (SP-Grid). These methods achieve good results without any specific training, at the expense of relatively large computation times, as features must be extracted for each region independently instead of the whole image at once, and feature comparisons are performed several times: once per region pair.
Instead of relying on models trained for object recognition, we advocate methods that perform specific training to achieve the desired invariances in the extracted representations. This enables developing efficient models of the right size for the task at hand. Our proposal, first made public by Lopez-Antequera et al. (2017), adopts this philosophy, and relies on triplets of images with ground-truth location to train the network to achieve the desired invariances. Later, a similar approach was presented by Arandjelovic et al. (2016), where the authors also propose a new layer based on the vector of locally aggregated descriptors (VLAD) to enhance viewpoint invariance. Their network is also trained with triplets, although selected focusing on the appearance variation in the images instead of the viewpoint variations that we seek in our datasets.

3.3 Methodology

We propose to solve the Place Recognition problem by training a Convolutional Neural Network to embed images in a low dimensional space where small Euclidean distances are representative of close-by locations. For that, we discriminatively train the network using datasets which present close-by locations under different illumination, point of view or weather conditions, using a triplet-based training scheme. Triplet training schemes have already been used in conjunction with neural networks as a nonlinear dimensionality reduction technique for applications such as content-based image retrieval (Wang et al., 2014), face recognition (Schroff et al., 2015), object recognition (Wohlhart and Lepetit, 2015) and more recently for producing local image descriptors (Kumar et al., 2015; Balntas et al., 2016). In this case, we select triplets formed by two images from the same location (similar pair) and a third one from a different location (dissimilar pair). Notice that the similar pair is formed by images that, while being from the same location, present differences in appearance (different weather, illumination or perspective). In the following sections we describe the architecture of the network, the cost function, the datasets used during training and the training procedure itself.

3.3.1 Architecture of the CNN

We build our network upon an existing architecture trained on the Large Scale Visual Recognition Challenge (ILSVRC) 2012 dataset, as the datasets available for place recognition are relatively small. We take the reference CaffeNet network (Jia et al., 2014), which mirrors the architecture of Krizhevsky et al. (Krizhevsky et al., 2012), known as AlexNet. More recent works (Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2015) exceed AlexNet’s classification performance in
3. Trainable image descriptors for place recognition

(a) Query location

(b) Same location

(c) Different location

Figure 3.2: Training triplet extracted from the KITTI dataset (Geiger et al., 2012), where large viewpoint changes are present. Images (a) and (b) form the similar pair (same location), while images (a) and (c) form the dissimilar pair (different location).

the ILSVRC, however, these models are much larger in memory and execution time, and we argue that the level of abstraction required for place recognition is lower than other tasks such as image classification, allowing for smaller models. In fact, we only keep the first four convolutional layers from AlexNet, replacing the rest with one fully connected layer producing our descriptor (see Figure 3.3). Since we discard all the fully connected layers, we are not constrained to the original input size of $227 \times 227$ pixels and instead we work with a smaller input of $160 \times 120$ pixels, in the interest of reducing the execution time and memory footprint of the network while directly using an input with a 4:3 aspect ratio, as found in most consumer-grade cameras and in most of the images employed for training our network.
3.3 Methodology

3.3.2 Triplet similarity embedding

We train our network through a triplet embedding scheme, where the same network is used to produce three descriptors $h(I_i)$, $h(I_j)$, $h(I_k)$ from three input images $I_i$, $I_j$, $I_k$ forming the similar $\{h(I_i), h(I_j)\}$ and dissimilar $\{h(I_i), h(I_k)\}$ descriptor pairs. All three descriptors are then used as input to a triplet hinge loss function:

$$C(I_i, I_j, I_k) = \max \left\{ 0, 1 - \frac{\|h(I_i) - h(I_k)\|_2}{\beta + \|h(I_i) - h(I_j)\|_2} \right\}. \quad (3.1)$$

which is zero when the distance of the dissimilar pair is larger than the distance of the similar pair by at least a margin $\beta$. Triplets not satisfying this condition will produce non-zero costs that the training process will attempt to reduce by updating the weights of the CNN accordingly through stochastic gradient descent. This cost function resembles the one proposed by Wohlhart and Lepetit (2015) for object recognition, although we exclude the pairwise term as we found it was unnecessary for the task. The pairwise term penalizes the separation of descriptors extracted from images of the same location, pulling them together during training. We argue that this is not required for place recognition tasks, as the triplet term already provides separability by the margin $\beta$. A concern when removing the pairwise term is that distances in the descriptor space could grow indefinitely during training, however, we found that the magnitude of the descriptors stabilizes during training as the margin in the triplet term is satisfied, even when L2 regularization is not performed.
3.3.3 Triplet selection

To achieve the desired invariances in the descriptors produced by the network, triplets must be chosen as to provide relevant visual cues (see Figure 3.2 for an example). We train the network using a mixture of triplets from several datasets to improve invariance to lighting, weather and point of view changes. In the following we describe the sources and rules for obtaining these triplets, followed by the applied training procedure.

**KITTI Dataset (Geiger et al., 2012)**

The odometry benchmark from the KITTI dataset comprises 11 training sequences with accurate ground truth of the trajectory, and 10 test sequences without ground truth for evaluation. Both the training and the test sequences are stereo frames extracted from urban environments in daylight conditions. We select triplets from this dataset in order to increase the robustness of the network to changes in viewpoint. The similar pair is chosen so that the images are taken from poses separated by less than 5 meters and rotated by, at most, 30 degrees. The dissimilar image is selected so that it is at least 8 meters away and rotated by more than 30 degrees. Figure 3.2 depicts a triplet extracted from the KITTI dataset. We use sequences 1 to 10 for training the network, reserving sequence 11 for testing.

**Alderley Dataset (Milford and Wyeth, 2012)**

This dataset is formed by two 8 km sequences along the suburb of Alderley in Brisbane (Australia). The first sequence was recorded during a clear morning, while the second one was collected in a stormy night with low visibility (see Figure 3.4). In order to achieve robustness to these changes in appearance, we provide the network with triplets that combine images from both sequences. Although the dataset does not include accurate ground truth location, all of the frames in the first sequence are manually matched to the second one. We select the similar pair by selecting matching images from both sequences, and the dissimilar pair by randomly selecting an image from any of the two sequences which is at least separated by 1.000 frames, roughly 500m. We use the first 10.000 frames from the day sequence and their matches from the night sequence for training the network, reserving the rest for testing.

**Nordland Dataset**

The Nordland dataset, extracted from the documentary “Nordlandsbanen - Minutt for Minutt” consists of a 728 km long train journey connecting the cities of Trond-
3.3. Methodology

heim and Bodø, in Norway. The journey was recorded once in each season, presenting challenging appearance changes due to the changing landscape and weather, as Figure 3.1 shows. The recordings were manually aligned so that frames with the same numeral are from the same location, we generate triplets by providing two images from the same place in different seasons, and an image from another location in any season, separated by at least 100m using the provided ground truth. We separate the dataset in two parts, keeping the first 30.000 frames for training and the last 5.700 frames for testing.

![Daylight sequence](image1)

(a) Daylight sequence

![Stormy night sequence](image2)

(b) Stormy night sequence

Figure 3.4: Frames extracted from the Alderley dataset [Milford and Wyeth 2012], where the same trajectory is recorded twice, presenting drastic illumination changes.
3.3.4 Training

The network is trained using the Caffe library \cite{Jia2014}, modified to include the previously described cost function. The weights of the four convolutional layers are fine-tuned from the CaffeNet reference network, whereas the weights for the final fully connected layer are initialized by sampling $\mathcal{N}(0, 0.01)$. The learning rate is set to $10^{-6}$ for the pre-trained convolutional layers and to $10^{-5}$ for the new fully connected layer. The margin $\beta$ is set to 1 and the L2 regularization constant $\lambda$ to 0.0005. We found momentum and dropout to have little impact on the results. These hyperparameter settings were selected through quasi-random grid sampling, first on learning rate and regularization, and then on momentum and dropout. Our final network is trained for 40,000 iterations, with a minibatch size of 30 for a total of 1.2 million triplets.

3.4 Experimental Evaluation

We perform a series of experiments over the aforementioned datasets, where we compare our proposal with the following state-of-the-art techniques in place recognition and image retrieval:

- Feature histogram descriptors from DBoW2 \cite{Mur-Artal2015}
- From AlexNet \cite{Krizhevsky2012} and VggNet-19 \cite{Simonyan2014}, we take the activations of the last two convolutional layers, which were found to be best for place recognition and instance retrieval \cite{Chen2014, Razavian2014}, and compare them as:
  - Raw descriptors \cite{Chen2014}
  - 1x1 and 2x2 max-pooled descriptors \cite{Razavian2014}
  - PCA-reduced descriptor (128 dimensions).

The dimensions of each descriptor are summarized in Table 3.1. For a fair comparison, we employ the optimal scoring method for similarity between descriptors for each algorithm and layer. For our method, the dissimilarity score is, by design, the L2 norm of the descriptors of each pair of images, i.e., $\| h(I_i) - h(I_j) \|_2$. In the case of DBoW2, the similarity score is calculated from the L1 distance of histograms generated from tf-idf weighted visual words as part of the authors’ implementation. Finally, for all of the other CNN-based methods, we compared the descriptors from each of the internal layers of the networks by using all L1, L2 and cosine distances.
as dissimilarity scores, always obtaining better results with L2 distance. Therefore, for the sake of clarity, we ignore the results obtained when using L1 and cosine distances as the dissimilarity score for the competing CNN methods in our figures. The implementations we have used are the official distribution of ORB-SLAM (Mur-Artal et al., 2015), and the Caffe (Jia et al., 2014) releases of both AlexNet (Krizhevsky et al., 2012) and VggNet (Simonyan and Zisserman, 2014). The resolution of the input images are 160 x 120 pixels for our proposal, 224 x 224 pixels for VggNet, 227 x 227 pixels for AlexNet, and the native resolution of each dataset for DBoW2.

To test our method and perform the comparisons, we take the standard approach of collecting scores in a similarity matrix, where the rows and columns correspond to the database and query sequences. Since there is no ground-truth value for the similarity of any two images, we take the common approach of evaluating the performance of the methods in a classification framework. We test on synchronized sequences without any loop closures, where the ground-truth pair is the diagonal of the similarity matrix (that is, when the query and database images belong to the same location). Then, we select the $k$ best scoring matches (the $k$ images from the database sequence that score best when compared to the query image), and count a match as an inlier (true positive match) if it is close enough to the ground truth (the diagonal) with a tolerance of $d$ frames. Finally, we generate precision-recall curves by varying that tolerance, and measuring the ratio of inliers for all the considered methods.
We test only on synchronized sequences which do not present any loop closures, since in those cases the ground truth pair is the diagonal of the similarity matrix. For every row (query frame) we select the $k$ best scoring matches. We count a match as an inlier if it is close enough to the diagonal with tolerance $d$. The ratio of inliers is the figure of merit with which we compare all the methods.

### 3.4.1 Results

Two comparison curves are plotted for each dataset, by picking 5 and 10 best matching frames ($k = 5$ and $k = 10$). All of the datasets have been evaluated by ranging $d$ from 0 to 20 frames and plotting the inlier (true positive) ratio.

**KITTI Dataset**

Figure 3.5 depicts the similarity matrix obtained with our method on the sequence KITTI-11 by using the images from the left and right cameras of the stereo rig as the query and database sequences, respectively. It presents a thick diagonal, which suggests that our approach is robust to changes in the point of view. Nevertheless, it is quite difficult to extract quantitative conclusions by observing the similarity matrices. Figure 3.6 shows the performance of the compared methods in this sequence, where we can observe that all methods score well, as the appearance changes due to the relative position of the cameras in the stereo rig is minimal.

**Nordland Dataset**

For these experiments, we have employed the last hour of the dataset, which was not used for training, and removed the segments which include either tunnels or stations. Figure 3.7 depicts the performance curves of the approaches by comparing the most challenging sequence pair, summer and winter. We observe that our proposal outperforms the rest, being the PCA reductions of the last two convolutional layers of VggNet the second and third best methods.

**Alderley Dataset**

We compare using the last 5k frames from the day sequence and their matches from the night sequence of the Alderley dataset. Figure 3.8 shows the performance of our proposal against the compared methods, with a better ratio of inliers in all cases. However, it can be noticed that a low ratio is obtained by all of the approaches, since it is a highly challenging dataset. Hence, the use of a post-processing technique on the similarity matrix based on sequentiality (Milford 2013; Pepperell et al. 2014).
Figure 3.6: Performance curves on the KITTI-11 sequence, when comparing the left and the right cameras.
Figure 3.7: Performance curves on a subset of the Nordland dataset, when using the summer sequence as database and the winter sequence as query inputs.
3.4. Experimental Evaluation

Table 3.1: Dimensionality of the features and compute time.

<table>
<thead>
<tr>
<th>Network(layer)</th>
<th>Raw</th>
<th>1x1 pool</th>
<th>2x2 pool</th>
<th>PCA</th>
<th>t (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our proposal</td>
<td>128</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.83</td>
</tr>
<tr>
<td>Alexnet(conv4)</td>
<td>64896</td>
<td>384</td>
<td>1536</td>
<td>128</td>
<td>0.29</td>
</tr>
<tr>
<td>Alexnet(conv5)</td>
<td>43264</td>
<td>256</td>
<td>1024</td>
<td>128</td>
<td>0.31</td>
</tr>
<tr>
<td>VggNet16(conv5-2)</td>
<td>100352</td>
<td>512</td>
<td>2048</td>
<td>128</td>
<td>25.6</td>
</tr>
<tr>
<td>VggNet16(conv5-3)</td>
<td>100352</td>
<td>512</td>
<td>2048</td>
<td>128</td>
<td>25.8</td>
</tr>
<tr>
<td>VggNet19(conv5-3)</td>
<td>100352</td>
<td>512</td>
<td>2048</td>
<td>128</td>
<td>30.5</td>
</tr>
<tr>
<td>VggNet19(conv5-4)</td>
<td>100352</td>
<td>512</td>
<td>2048</td>
<td>128</td>
<td>30.6</td>
</tr>
<tr>
<td>NetVLAD(CPU)</td>
<td>4092</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1000*</td>
</tr>
<tr>
<td>Protzel2015(CPU)</td>
<td>100k-300k</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5000**</td>
</tr>
<tr>
<td>DBoW2(CPU)</td>
<td>200-500</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4-22</td>
</tr>
</tbody>
</table>

* Although we didn’t run NetVLAD on a GPU, we expect it to be slightly slower than Alexnet(conv5).
** Extraction times only. Comparison times are relatively expensive for this method.

would be necessary to obtain a system with a reasonable performance in similar scenarios.

3.4.2 Computational performance

Finally, we examine the computational performance in several aspects, which are presented in Table 3.1. Our tests run on a Core i7-3770, while our GPU tests also rely on a GeForce GTX 790. We measure the time required to process a single image. In CNN-based methods, the value includes loading the image and performing a forward pass to obtain the feature vector. In the case of DBoW2 [Mur-Artal and Tardos, 2014], we measure the time required to compute the bag-of-words histogram. Since the input image resolution for DBoW2 is variable depending on the dataset, we show the minimum and maximum average times from all the sequences. The results indicate that DBoW2 is less demanding than CNN-based methods, and that our network is faster than all the other CNN-based approaches. The size of the descriptor is relevant for long term operation and for the cost of performing any operation with them, such as building the similarity matrix. The length of the word histogram of DBoW2 is variable in the official implementation, and can be as long as the dictionary size (32k elements). In our experiments, the length of the DBoW2 histogram varied from 200 to 500 elements. On this matter, our method outperforms the rest with a smaller, fixed length descriptor of 128 elements, matched only by the PCA reductions.
Figure 3.8: Performance curves on the final part of the Alderley dataset. We have employed the day sequence as database, and the challenging night sequence as query.
3.5 Conclusions

We have trained a convolutional neural network to perform place recognition under challenging appearance changes due to weather, seasons, time of day and point of view. The network embeds images in a 128-dimensional space where samples from similar locations are separated by small Euclidean distances. The network was trained using triplets of images from datasets where accurate ground-truth location is available and where the aforementioned changes in appearance are present, in order to allow the network representation to be invariant to these changes.

The proposed network outperforms all generic methods when conditions are challenging. It also outperforms the other trained method [Arandjelovic et al., 2016], although the latter has not been exposed to the domain of images which we use for testing.

Our descriptor is faster and more compact than all of the competing approaches, making our solution suitable for real-time applications on portable computers or compact robots where long-term operation is required.