Fusion and Binarization of CNN Features for Robust Topological Localization across Seasons

Roberto Arroyo[1], Pablo F. Alcantarilla[2], Luis M. Bergasa[1] and Eduardo Romera[1]

***Abstract*— The extreme variability in the appearance of a place across the four seasons of the year is one of the most challenging problems in life-long visual topological localization for mobile robotic systems and intelligent vehicles. Traditional solutions to this problem are based on the description of images using hand-crafted features, which have been shown to offer moderate invariance against seasonal changes. In this paper, we present a new proposal focused on automatically learned descriptors, which are processed by means of a technique recently popularized in the computer vision community: Convolutional Neural Networks (CNNs). The novelty of our approach relies on fusing the image information from multiple convolutional layers at several levels and granularities. In addition, we compress the redundant data of CNN features into a tractable number of bits for efficient and robust place recognition. The final descriptor is reduced by applying simple compression and binarization techniques for fast matching using the Hamming distance. An exhaustive experimental evaluation confirms the improved performance of our proposal (CNN-VTL) with respect to state-of-the-art methods over varied long-term datasets recorded across seasons.**

I. Introduction

Where am I? This is one of the most important questions that any mobile robot or autonomous vehicle must solve, with the aim of determining its location along the time and facilitating a life-long navigation. The great advances reached in computer vision in the last few years have allowed to perform some of these complex localization tasks by means of cheap image sensors, such as cameras. Unfortunately, the performance of the major part of the proposed solutions commonly decreases when long periods of time are considered. The main reason is related to the strong appearance changes that a place suffers in long-term situations due to dynamic elements, illumination or weather. More specifically, the variations on the visual perception of a place across the four seasons of the year are currently one of the most challenging problems in camera-based localization, as can be observed in the image examples depicted in Fig. 1.

Currently, several new concepts provided in computer vision areas such as deep learning can help to robustly solve the previously mentioned dilemmas associated with seasonal visual localization. According to this, Convolutional Neural Networks (CNNs) are one of the most commonly employed techniques in deep learning over the last years [1]–[13].

*This work is funded by the UAH through a FPI grant, the Spanish MINECO through the project Smart Driving Applications (TEC2012-37104) and the CAM through the project RoboCity2030-III-CM (P2013/MiT2748).

[1]Department of Electronics, University of Alcalá (UAH), Alcalá de Henares, 28871, Madrid, Spain. {roberto.arroyo, bergasa, eduardo.romera}@depeca.uah.es

[2]iRobot Corporation, 10 Greycoat Place, Victoria, London, UK. palcantarilla@irobot.com

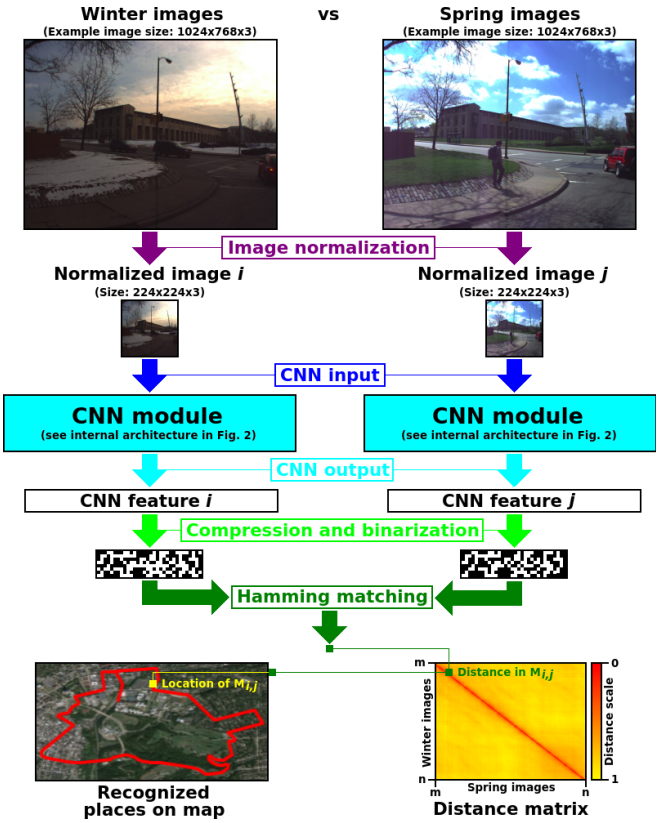


Fig. 1. Global system architecture of our approach for visual topological localization. It must be noted that the internal architecture of the CNN module is extensively described in Fig. 2. In this case, the sample images correspond to the CMU-CVG Visual Localization dataset, where the extreme changes that a place suffers across the seasons of the year can be perceived.

The popularization of CNNs is due to the great descriptive power that they support in image recognition, which makes them suitable for an immense variety of problems studied by the computer vision community. Our goal is to also take advantage of these deep learning concepts for the benefit of the robotics community, especially in perception robotic systems that depends on cameras, such as the designed for the automation of visual localization in long-term scenarios.

In this paper, we contribute a new proposal that exploits the advantages of powerful feature representations via CNNs in order to perform a robust topological vision-based localization across the seasons of the year, as introduced in the graphical explanation of our approach given in Fig. 1. In this topic, some of the most representative state-of-the-art works [14]–[20] have been typically based on describing locations by means of some traditional hand-crafted features, such as SURF [21], BRIEF [22] or more recently LDB [23].

Inspired by the success of how image representations learned with CNNs on large-scale annotated datasets can be transferred to other recognition tasks [12], we consider the possibility of using pre-trained CNN features for the identification of places in our visual localization approach.

Nevertheless, the main inconvenience of using CNNs is that they usually are expensive in terms of computational costs and memory resources, which sometimes is a problem for real-time performance. For this reason, we present an efficient CNN model for computing our features that provides not only a high precision in visual topological localization across the seasons, but also a reduced consumption of resources and processing costs. With the aim of achieving this efficiency maintaining the effectiveness of our approach, we provide several innovative proposals that suppose a contribution regarding to the current state of the art (more extensively described in Section II). These main contributions are the following:

- An improved CNN architecture based on some of the ideas of [1] and [3]. Our model is adapted and reduced to the requirements of our visual localization system. The CNN is pre-trained in a different dataset [2] with respect to the used in tests to demonstrate how transferable are the learned features [13] (see Section III-A).
- A novel fusion of the features obtained by the convolutional layers that improves the performance. The redundancy of this fused features is subsequently decreased by applying feature compression (see Section III-B).
- A binarization of the final reduced features with the aim of improving the matching of locations by computing an efficient Hamming distance (see Section III-C).
- A wide set of results comparing our method against the main state-of-the-art algorithms in three large datasets with seasonal changes [15], [17], [24] (see Section IV).
- A discussion about the most relevant conclusions of our work and future research lines (see Section V).

II. Related Work

A. Convolutional Neural Networks (CNNs)

Nowadays, CNNs have revolutionized the computer vision community, mainly due to the innovative work presented by [1], which defined one of the most relevant CNN architectures: AlexNet. It obtained impressive results in image classification over the challenging ImageNet dataset [2]. After that, other works tested the power of CNNs by designing new refined architectures based on AlexNet and widely comparing them against other visual recognition methods [3]. In this sense, the benefits of CNNs have been exhibited in a different range of typical computer vision problems such as semantic segmentation [4] or optical flow [5]. The popularization and extension of these deep learning algorithms in varied contexts has been also possible thanks to the useful open toolboxes provided by some authors to the community, such as Caffe [6] or MatConvNet [7]. Besides, other contributions helped researchers to better understand the complex division into layers behind CNNs and to visualize their features [8].

The application of CNNs to learn robust visual descriptors has been studied in works such as [9], where features are processed for global image description. Moreover, other proposals extract CNN features over a set of detected points [10], like traditional hand-crafted local descriptors such as SURF. Apart from general image recognition, there are also approaches where deep descriptors are trained over a specific database of places [11], with the aim of categorizing concrete place scenes such as forests, coasts, rooms, etc. On the other hand, some works analyze how transferable is the knowledgement acquired in training by deep neural networks [12], [13], which demonstrates that learned features appear not to be specific to a particular dataset or problem, they can be generalizable to several datasets and problems.

B. Life-long Visual Topological Localization

The major part of the state-of-the-art proposals for visual topological localization are focused on description methods that compute conventional hand-crafted features from single images. Probably, the most popular algorithm in this line is FAB-MAP [14], which uses vector-based descriptors like SURF jointly with bags-of-words. Global vector-based description methods like WI-SURF are also applied in topometric localization [15]. Besides, the irruption of BRIEF and similar binary features motivated place recognition proposals based on a global binary description efficiently matched by means of the Hamming distance, such as BRIEF-Gist [16].

Unfortunately, the previously mentioned approaches typically diminish their precision over long periods of time. Due to this, algorithms such as SeqSLAM [17] improved the accuracy in long-term scenarios using sequences of images to define places instead of single images, jointly with a customized descriptor based on image difference vectors computed over pixels. More recently, new proposals obtained a remarkable performance focusing on the challenge of life-long topological localization across seasons, such as the three versions of ABLE contributed for different types of cameras: panoramic (ABLE-P [18]), stereo (ABLE-S [19]) and monocular (ABLE-M [20]). This approach presents a visual description of locations based on sequences of illumination invariant images represented by binary codes, which are extracted from a global LDB descriptor that is fast matched using an approximated nearest neighbor search.

However, methods based on CNN features can be a promising alternative for a more precise place recognition across seasons. Recent papers exhibited studies about possible utilities of pre-trained CNN features in visual topological localization [25], [26]. On the other hand, the approach defined in [27] performs end-to-end learning of a CNN for identifying places. Other interesting models based on deep learning are focused on pose regression for relocalization in small-scale environments [28], in contrast to works where images are matched under substantial appearance changes exploiting GPS metric priors [29]. These recent proposals have motivated our current work, with the aim of providing a more robust and efficient life-long visual localization system based on improved and simplified CNN features.

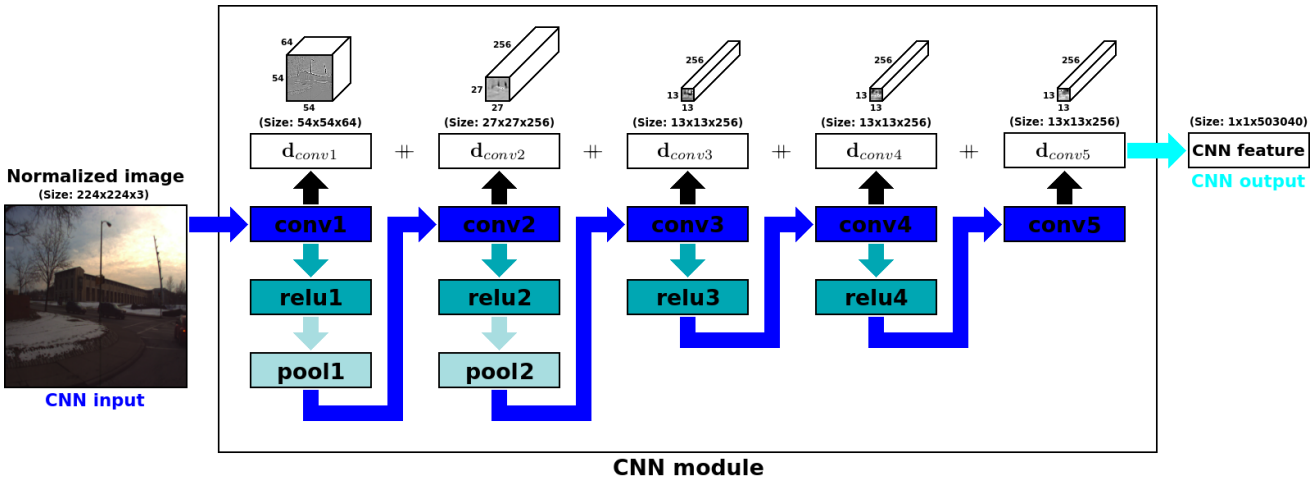


Fig. 2. CNN-VTL architecture. The description process takes as input an image normalized to a size of 224x224x3 and returns as output a CNN feature. Internally, our CNN is divided into different layers that capture image information at several levels and granularities, where five convolutions are computed. The features obtained for each convolution are fused to form the final feature, which will be compressed and binarized in subsequent stages of our system.

III. Our Visual Topological Localization (CNN-VTL)

A. CNN architecture

The architecture designed for our CNN model follows some concepts of the VGG-F presented in [3], which takes as reference a similar structure to the one suggested by AlexNet, but including some improvements for a faster processing. To achieve a more efficient performance, VGG-F reduced the number of convolutional layers originally proposed by AlexNet to eight learnable layers, five of which are convolutional, and the last three are fully-connected. In the results provided by [3], it is corroborated how this simplification does not have a significant impact in the effectiveness for image recognition, but it greatly decrease the computational costs. Inspired by these experiments, we implement a much more simplified approach. We also take into account the study about CNNs performance in place recognition carried out by [26], which demonstrates that fully-connected layers are not so effective as the convolutional ones in this task, as certified by our own observations. For this reason, our final model eliminates fully-connected layers and is mainly based on five convolutions. According to all the previous considerations, our Convolutional Neural Network for Visual Topological Localization (CNN-VTL) is graphically described in Fig. 2.

Following the ideas exposed in [12], [13], our CNN-VTL architecture is based on a pre-trained model over the ImageNet dataset, in order to confirm the generalization of the automatically learned features. This will demonstrate that the description power acquired by the CNN features is transferable to the specific datasets used in the tests of our visual topological localization across seasons. Besides, this evaluation is much fairer than processing training images recorded in environments similar to the evaluated in the datasets where our tests are registered.

The architecture designed in CNN-VTL is modeled thanks to some of the functionalities provided by the MatConvNet toolbox, which allows to create a great variety of different wraps to define the layers in the network. As can be seen in Fig. 2, our architecture is formed by three main types of layers: convolutions, ReLUs and pools. The mechanisms for computing the information of each layer inside of our CNN-VTL are not trivial. For this reason, now we must provide a more detailed explanation about how the different layers work in our specific model:

1) *Convolutional layers ($conv_n$)*: convolutions are the basic layer in any CNN and they are usually the main level in all the blocks. In the case of our CNN-VTL, five convolutional layers are on the top of the five blocks that form the proposed architecture. The derivatives associated with convolutions are solved with techniques of backpropagation. Each convolutional layer receives an input map ($\mathbf{x} \in \mathbb{R}^{H \times W \times D}$) and a bank of filters with multiple dimensions ($\mathbf{f} \in \mathbb{R}^{H' \times W' \times D \times D''}$), returning the subsequent output ($\mathbf{y} \in \mathbb{R}^{H'' \times W'' \times D''}$). It must be noted that in our implementation any bias input is processed. Taking into account the inputs and the output, the internal computation of our convolutions is represented by Eq. 1:

$$y_{i''j''k''} = \sum_{i'=1}^{H'} \sum_{j'=1}^{W'} \sum_{k'=1}^{D} f_{i'j'k} \times x_{i''+i'-1,j''+j'-1,k',k''} \tag{1}$$

2) *ReLU layers ($relu_n$)*: the Rectified Linear Unit (ReLU) is an activation function used by our CNN model. In CNN-VTL, a ReLU layer is located after all the convolutional layers, except in the case of the last convolution ($conv_5$), where it is unnecessary according to our feature generation proposal. We choose the ReLU activation function because it is very easily and efficiently computed, in contrast to other more complex functions, such as sigmoidals. The activation in ReLU is simply thresholded at zero, as exposed in Eq. 2:

$$y_{ijk} = \max(0, x_{ijk}) \tag{2}$$

3) *Spatial pooling layers ($pool_n$)*: pools are an important layer in the proposed architecture. Spatial pooling decreases the amount of parameters and computational costs over our CNN using a non-linear downsampling, which also allows to control problems derived from overfitting. CNN-VTL contains two pools in the lower part of its two first blocks connected to $conv_2$ and $conv_3$, which are sufficient to achieve the desired reduction of data. In our case, a max pooling operator is implemented. It processes the maximum response of each feature channel in a $H' \times W'$ patch. The internal application of this layer is formulated in Eq. 3:

$$y_{i''j''k} = \max_{1 \leq i' \leq H', 1 \leq j' \leq W'} x_{i''+i'-1, j''+j'-1, k} \quad (3)$$

B. Description of locations

The model defined by the layers of CNN-VTL allows to carry out a strong visual description of places at different image levels and granularities for the proposed visual topological localization across seasons. The visual features automatically learned by our network can be used now by applying the designed convolutions.

In order to form a more robust final descriptor ($\mathbf{d}_{cnn}$) from our CNN-VTL, we concatenate ($+\!\!+$) the vectorized features obtained by the n different convolutional layers ($\mathbf{d}_{conv_n}$):

$$\mathbf{d}_{cnn} = \mathbf{d}_{conv_1} +\!\!+ \mathbf{d}_{conv_2} +\!\!+ \mathbf{d}_{conv_3} +\!\!+ \mathbf{d}_{conv_4} +\!\!+ \mathbf{d}_{conv_5} \quad (4)$$

The goal of the strategy formulated in Eq. 4 is to conserve the multiresolution information provided by each convolution, which acts as a local and translation invariant operator, as stated in [7]. In works such as [25], only the features generated by the third convolutional layer ($\mathbf{d}_{conv_3}$) are considered in the place description process, which produces a loss of invariance. Due to this, [25] needs to use a complex and expensive algorithm to obtain several region landmarks per image and compute their respective CNN features in order to maintain the robustness when revisited locations have important changes on the field of view. On the other hand, our method can directly use global images thanks to the invariance procured by the fusion of the convolutional outputs, which is a more efficient approach.

The different features contained in $\mathbf{d}_{cnn}$ are initially returned by the CNN in a float format. For this reason, with the aim of facilitating a subsequent binarization, we cast these features into a normalized 8-bit integer format ($\mathbf{d}^{int}_{cnn}$) by following Eq. 5, where $min = 0$ and $max = 255$:

$$\mathbf{d}^{int}_{cnn} = (\mathbf{d}_{cnn} - \min(\mathbf{d}_{cnn})) \frac{max - min}{\max(\mathbf{d}_{cnn}) - \min(\mathbf{d}_{cnn})} + min \quad (5)$$

The length of the descriptor (l_{cnn}) acquired after the fusion of the five convolutional outputs can be calculated as exposed in Eq. 6, where h_{conv_n}, w_{conv_n}, d_{conv_n} are the height, width and dimensions of each convolution, respectively:

$$l_{cnn} = \sum_{n=1}^{5} h_{conv_n} \times w_{conv_n} \times d_{conv_n} \quad (6)$$

If we solve Eq. 6 using the output sizes of the convolutions applied in our CNN-VTL architecture (see Fig. 2), we obtain $l_{cnn} = 503040$ bytes. This length can be excessive for efficiently performing the subsequent features matching. Due to this, we apply reductions to this size in order to analyze how they affect to the accuracy of our place recognition method. This is motivated by works such as [30], which evidences that a handful of bits is sufficient for conducting an effective vision-based navigation. Besides, in very recent studies [31], several traditional hand-crafted binary descriptors are tested to conclude that a remarkable precision can be achieved using a small fraction of the total number of bits from the whole descriptor. We demonstrate in the experiments presented in Section IV-B that the features extracted from CNNs have a similar behavior to the observed in [31].

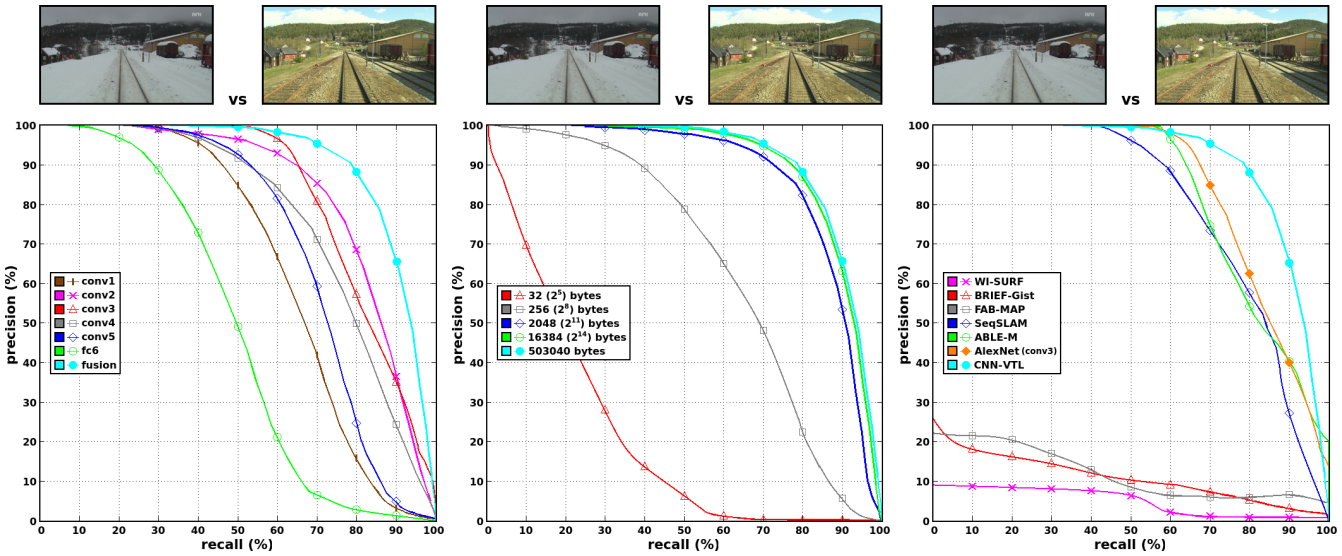
With the aim of reducing the size of our CNN descriptors without losing a great accuracy, the redundant features can be omitted to compress the final length. In works such as [22] or [23], methods based on a random selection of features have demonstrated to be an efficient and effective alternative with respect to more complex algorithms. In fact, the evaluation presented in the binary description performed by LDB in [23] yielded surprisingly favorable results, where the precision of a random feature selection is close to the one achieved using more refined methods, such as entropy-based.

We also implement a similar random selection of features in order to compress our CNN descriptor in an easy and efficient way. This technique randomly chooses a specific set of features and applies the same selection in all the following descriptions to match the same correlative features. A proportional number of features is randomly selected for each layer to preserve as possible the multiresolution provided by our fusion of convolutional features at different granularities. Our compression proposal is supported by the satisfactory results exposed in Fig. 3 (b) and Table I.

C. Matching of locations

The bottleneck of our system is in the matching of descriptors for identifying locations, because the number of images to be matched is increased in each iteration, while the description costs are constant along the time. Apart from features compression, other techniques can be applied for reducing the computational costs of matching tasks. One of them is the usage of the Hamming distance for obtaining the similarity between features, which is more efficient than the L_2 norm or the cosine distance used in works such as [25]. This efficiency is due to the simplicity of its calculation, which consists on an elementary XOR operation ($\oplus$) and a basic sum of bits, as formulated in Eq. 7. According to this, our CNN descriptors are binarized, because this is the main condition to correctly use the Hamming distance. This binarization is a trivial operation after the conversion of our features to an 8-bit integer format (see Eq. 5). Finally, a distance matrix (M) is computed by matching all the binary features ($\mathbf{d}^{bin}_{cnn}$) using the Hamming distance:

$$M_{i,j} = \text{bitsum}(\mathbf{d}^{bin}_{cnn_i} \oplus \mathbf{d}^{bin}_{cnn_j}) \quad (7)$$



(a) Winter vs Spring. CNN layers comparison. (b) Winter vs Spring. Compression comparison. (c) Winter vs Spring. State of the art comparison.

Fig. 3. Precision-recall curves comparing results about CNN-VTL in some of the most challenging sequences of the Nordland dataset (Winter vs Spring).

IV. Evaluation in Localization across Seasons

A. Datasets and Evaluation Methodology

With the aim of demonstrating the capability of our CNN-VTL method, we carry out several evaluations using three publicly available datasets, where several image sequences are recorded for a same route across the seasons of the year: the Nordland dataset [24], the CMU-CVG Visual Localization dataset [15] and the Alderley dataset [17]. These tests allow us to analyze the long-term behavior of our proposal over a distance of more than 3000 km and in the different conditions associated with each dataset.

We compare the performance of our solution against some of the main state-of-the-art works. For evaluating WI-SURF and BRIEF-Gist, we use implementations of them based on the SURF and BRIEF descriptors provided by the OpenCV library [32]. FAB-MAP is tested using the OpenFABMAP toolbox [33]. The experiments for SeqSLAM are performed with OpenSeqSLAM [24]. ABLE-M evaluations are computed thanks to the source code developed by authors [20]. Additionally, we implement the approach defined in [26] based on the $conv_3$ features obtained from an AlexNet pre-trained in MatConvNet over the ImageNet dataset.

The results presented in this work for our proposal and the state-of-the-art methods are compared by means of the objective evaluation methodology more detailed in [19], which is mainly based on precision-recall curves obtained from the distance matrices processed in each test.

B. Results in the Nordland Dataset

The Nordland dataset comprises a train trip of 10 hours across Norway, which is registered four times, once in each season. Image sequences are synchronized and field of view is always the same due to the invariant camera position. In this paper, we present results obtained between the routes recorded in winter and spring, which is one of the most challenging evaluations because of the extreme variability in visual appearance between both seasons in this dataset.

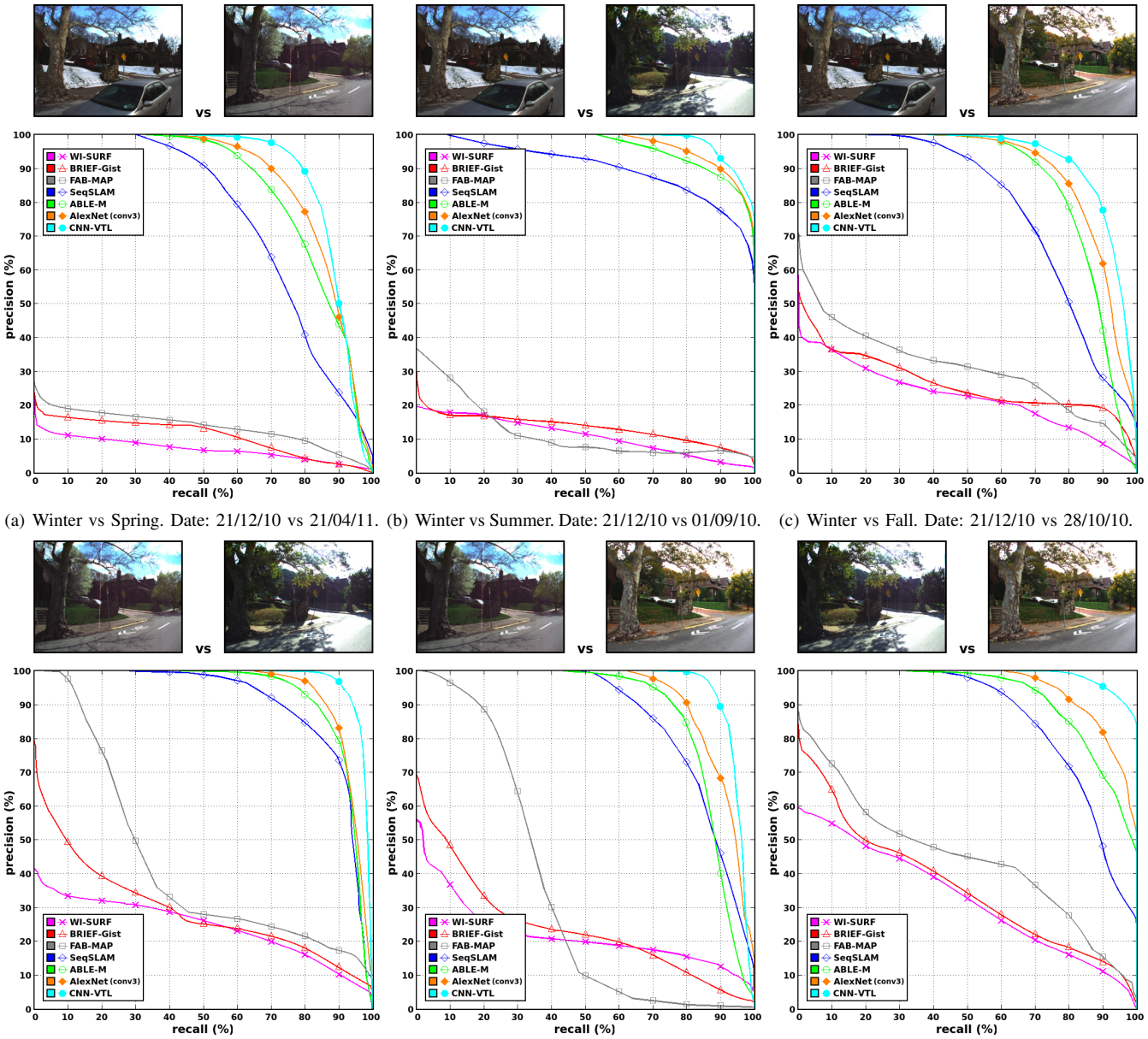
In Fig. 3 (a), it is corroborated how our approach focused on using a fusion of features from convolutional layers works much better than the features extracted from individual layers, which are proposed in works such as [25] or [26]. In this case, we also include a test for an added fully-connected layer (fc_6) to confirm its worse behavior in our problem.

Precision-recall curves depicted in Fig. 3 (b) validate our approach for reducing the amount of redundant information in our CNN features. Here, it is demonstrated that features can be highly compressed maintaining a remarkable performance. Table I presents a more detailed study, where it can be seen how our initial CNN features can be reduced to 2048 bytes (a compression of 99.59%), losing only about a 2% of precision. In higher reductions, (256 or 32 bytes) the loss of accuracy is much more critical. Table I also details the average speedups achieved in matching when features are compressed, which is proportional to the magnitude of the reduction.

The results in Fig. 3 (c) show that our CNN-VTL proposal obtains a successful performance compared to state-of-the-art algorithms based on traditional hand-crafted features. It must be also noted that the precision yielded by CNN-VTL is superior to the achieved in the curve computed for the CNN-based method defined in [26] (AlexNet $conv_3$).

TABLE I
Study about the performance of compressed features.

Size in bytes	F_1-score	Percentage of compression	Average speedup for matching
503040	0.899	0 %	None
16384	0.894	96.74 %	30x
2048	0.872	99.59 %	245x
256	0.651	99.94 %	1965x
32	0.216	99.99 %	15720x



(a) Winter vs Spring. Date: 21/12/10 vs 21/04/11. (b) Winter vs Summer. Date: 21/12/10 vs 01/09/10. (c) Winter vs Fall. Date: 21/12/10 vs 28/10/10.

(d) Spring vs Summer. Date: 21/04/11 vs 01/09/10. (e) Spring vs Fall. Date: 21/04/11 vs 28/10/10. (f) Summer vs Fall. Date: 01/09/10 vs 28/10/10.

Fig. 4. Precision-recall curves comparing our CNN-VTL against state-of-the-art algorithms in the CMU-CVG Visual Localization dataset.

C. Results in the CMU-CVG Visual Localization Dataset

This dataset contains several sequences of images acquired by a car in different months of the year around a same route in Pittsburgh (USA). In this case, apart from seasonal variations, there are changes on the camera field of view between the images recorded for a same place.

Now, we process results for sequences corresponding to the four seasons of the year in the six possible combinations (winter vs spring, winter vs summer, winter vs fall spring vs summer, spring vs fall, summer vs fall). These evaluations are depicted by the precision-recall curves showed in Fig. 4, where it can be observed how the different seasons affect to performance in life-long visual topological localization. In general terms, the sequence captured in winter is the most problematic in our tests, due to the extreme changes that a place suffers in this season: snow, less vegetation or different illumination, among others.

The results presented in Fig. 4 compare again the precision of CNN-VTL against state-of-the-art algorithms, but now in an environment where changes on the field of view have a negative effect in the accuracy of the major part of the methods. Our proposal obtains a better performance in this situation, because our fusion of CNN convolutional features at different levels and granularities provides a higher local and translation invariance with respect to approaches based on individual layers, such as [26]. Apart from this, precision-recall curves computed in Fig. 4 yield much worse results for algorithms based on traditional hand-crafted features using single images (WI-SURF, BRIEF-Gist, FAB-MAP) than using sequences of images (SeqSLAM, ABLE-M).

Moreover, Fig. 5 shows a distance matrix computed over two sequences of the dataset by our CNN-VTL. It evidences the reliable performance of our method, which is only unable to match a low amount of images when a truck occludes the camera view. Fig. 6 presents other complex situations where our method correctly detects a revisited place. In these cases, geometric change detection [34] could be applied to detect the specific variations in the structure of the matched place.

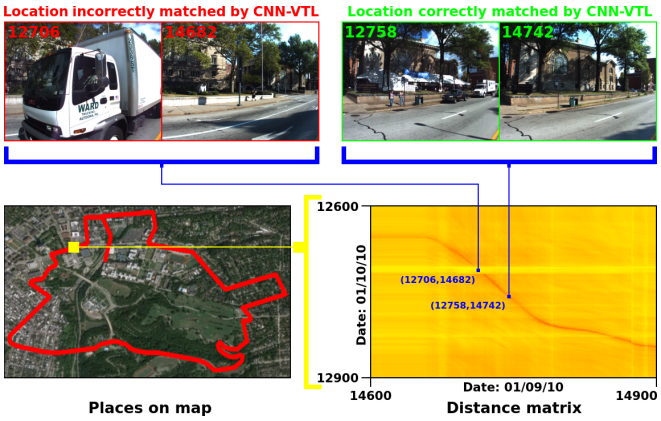


Fig. 5. A representative example of the distance matrix obtained by our CNN-VTL in a part of the map between sequences recorded at 01/09/10 and 01/10/10 in the CMU-CVG Visual Localization Dataset. In almost all the cases, it can be observed that locations are correctly matched (see red line in distance matrix), except in a low amount of frames where a truck completely occludes the camera view (see frames 12706 and 14682). Other complex situations are correctly matched, such as locations with new buildings that change the initial appearance of a place (see frames 12758 and 14742).

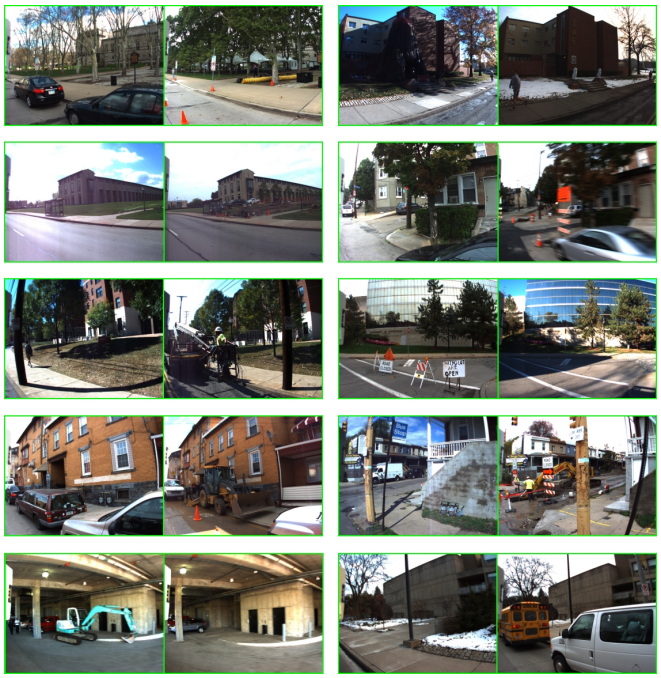


Fig. 6. More image pairs of complex locations correctly matched by our CNN-VTL in different sequences of the CMU-CVG Visual Localization Dataset. The images show difficult cases with extreme changes over the past appearance of a place: new buildings, constructions, dynamic elements, important changes on the field of view or partial occlusions, among others.

D. Results in the Alderley Dataset

The Alderley dataset comprises two sequences of images acquired in Brisbane (Australia). One of them is recorded in a stormy winter night and the other one in a sunny summer day. For this reason, apart from the typical seasonal changes previously studied, we can now perform evaluations under extremely variable illumination conditions. The precision-recall curves exposed in Fig. 7 show an acceptable accuracy for our CNN-VTL method in this challenging case.

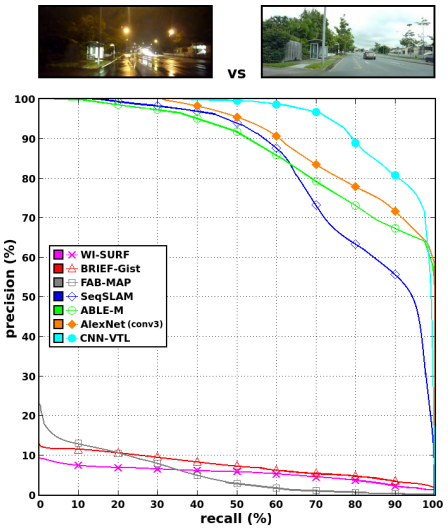


Fig. 7. Precision-recall curves comparing our CNN-VTL against state-of-the-art algorithms in the Alderley dataset (Winter night vs Summer day).

V. Final Discussion

A. Conclusions

Along this paper, our novel approach for life-long visual topological localization using convolutional neural networks (CNN-VTL) has extensively demonstrated its contribution to the robotics and computer vision communities. The proposed method has a valuable applicability in several fields related to tasks such as camera-based place recognition or loop closure detection, which are usually indispensable in any SLAM or visual odometry system.

Our proposal is validated in challenging conditions derived from the extreme changes that the visual appearance of a place suffers across the four seasons of the year. A wide set of results in varied long-term scenarios corroborates the remarkable performance of our CNN-VTL compared against the main state-of-the-art algorithms based on hand-crafted descriptors, such as WI-SURF, BRIEF-Gist, FAB-MAP, SeqSLAM or ABLE-M. Moreover, we have also evidenced that our method reports a better precision than other very recent approaches which have studied the application of CNNs for place recognition in robotics [26]. This is mainly due to our improved CNN architecture and to the fusion of the features acquired in several convolutional layers, that provides an enhanced local and translation invariance with respect to [26], which is mainly based on CNN features from a $conv_3$ layer computed by a pre-trained AlexNet.

In addition, we have contributed an efficient model with the aim of decreasing the costs associated with CNN descriptors. We exposed how our compression of features can reduce the redundancy of our descriptors in a 99.59%, while precision is only decreased in about a 2%, achieving a speedup in matching near to 245x in this case. Besides, our binarization of features allows to use the Hamming distance, that also represents a speedup to match locations.

B. Future research lines

Although the performance of our CNN-VTL can be considered quite satisfactory with respect to the main works in visual topological localization, there are some interesting future directions to follow such as:

- Test the application of sequences of images instead of single images in CNN-VTL, similarly to [17] and [20].
- Study the compression of CNN features by means of more refined techniques, such as Local Sensitive Hashing (LSH) or Principal Components Analysis (PCA).
- Perform end-to-end training of a CNN architecture such as the one described in [27] and analyze its generalization properties to different domains.
- Evaluate the usage of change detection methods [34] for updating the information about revisited places.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1106–1114.

[2] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference (BMVC)*, 2014.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[5] A. Dosovitskiy, P. Fischery, E. Ilg, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 2758–2766.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM International Conference on Multimedia (ACMMM)*, 2014, pp. 675–678.

[7] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *ACM International Conference on Multimedia (ACMMM)*, 2015, pp. 689–692.

[8] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision (ECCV)*, vol. 8689, 2014, pp. 818–833.

[9] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Workshops at the IEEE Conference on Computer Vision and Pattern Recognition (W-CVPR)*, 2014, pp. 806–813.

[10] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 118–126.

[11] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 487–495.

[12] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1717–1724.

[13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 3320–3328.

[14] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *International Journal of Robotics Research (IJRR)*, vol. 27, no. 6, pp. 647–665, 2008.

[15] H. Badino, D. F. Huber, and T. Kanade, "Real-time topometric localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1635–1642.

[16] N. Sünderhauf and P. Protzel, "BRIEF-Gist - Closing the loop by simple means," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1234–1241.

[17] M. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1643–1649.

[18] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes, and S. Gámez, "Bidirectional loop closure detection on panoramas for visual navigation," in *IEEE Intelligent Vehicles Symp. (IV)*, 2014, pp. 1378–1383.

[19] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes, and S. Bronte, "Fast and effective visual place recognition using binary codes and disparity information," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 3089–3094.

[20] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, "Towards life-long visual localization using an efficient matching of binary sequences from images," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6328–6335.

[21] H. Bay, T. Tuytelaars, and L. van Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision (ECCV)*, vol. 3951, 2006, pp. 404–417.

[22] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *European Conference on Computer Vision (ECCV)*, 2010, pp. 778–792.

[23] X. Yang and K. T. Cheng, "Local difference binary for ultrafast and distinctive feature description," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 36, no. 1, pp. 188–194, 2014.

[24] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons," in *Workshop on Long-Term Autonomy at the IEEE International Conference on Robotics and Automation (W-ICRA)*, 2013.

[25] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free," in *Robotics Science and Systems Conference (RSS)*, 2015.

[26] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of ConvNet features for place recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4297–4304.

[27] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5297–5307.

[28] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 2938–2946.

[29] O. Vysotska, T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, "Efficient and effective matching of image sequences under substantial appearance changes exploiting GPS priors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2774–2779.

[30] M. Milford, "Visual route recognition with a handful of bits," in *Robotics Science and Systems Conference (RSS)*, 2012.

[31] P. F. Alcantarilla and B. Stenger, "How many bits do I need for matching local binary descriptors?" in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2190–2197.

[32] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools (DDJ)*, vol. 25, no. 11, pp. 122–125, 2000. [Online]. Available: http://opencv.org

[33] A. J. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. F. Wyeth, "OpenFABMAP: An open source toolbox for appearance-based loop closure detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 4730–4735.

[34] P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi, "Street-view change detection with deconvolutional networks," in *Robotics: Science and Systems Conference (RSS)*, 2016.