

Article

# ConvNet and LSH-Based Visual Localization Using Localized Sequence Matching <sup>†</sup>

**Yongliang Qiao** <sup>1,\*</sup>, **Cindy Cappelle** <sup>2</sup>, **Yassine Ruichek** <sup>2</sup> and **Tao Yang** <sup>2</sup>

<sup>1</sup> Australian Centre for Field Robotics (ACFR), Department of Aerospace, Mechanical and Mechatronic Engineering (AMME), The University of Sydney, Sydney, NSW 2006, Australia

<sup>2</sup> Connaissance et Intelligence Artificielle Distribuées (CIAD), University Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France; cindy.cappelle@utbm.fr (C.C.); yassine.ruichek@utbm.fr (Y.R.); tao.yang@utbm.fr (T.Y.)

\* Correspondence: yongliang.qiao@sydney.edu.au

† The Main Experiments and Manuscript Were Finished in UTBM, While the Final Submission Was Conducted in ACFR. Part of the Experiment Results of This Paper Has been Published in Industrial Electronics Society, IECON 42nd Annual Conference of the IEEE.

Received: 20 April 2019; Accepted: 23 May 2019; Published: 28 May 2019



**Abstract:** Convolutional Network (ConvNet), with its strong image representation ability, has achieved significant progress in the computer vision and robotic fields. In this paper, we propose a visual localization approach based on place recognition that combines the powerful ConvNet features and localized image sequence matching. The image distance matrix is constructed based on the cosine distance of extracted ConvNet features, and then a sequence search technique is applied on this distance matrix for the final visual recognition. To speed up the computational efficiency, the locality sensitive hashing (LSH) method is applied to achieve real-time performances with minimal accuracy degradation. We present extensive experiments on four real world data sets to evaluate each of the specific challenges in visual recognition. A comprehensive performance comparison of different ConvNet layers (each defining a level of features) considering both appearance and illumination changes is conducted. Compared with the traditional approaches based on hand-crafted features and single image matching, the proposed method shows good performances even in the presence of appearance and illumination changes.

**Keywords:** visual localization; place recognition; convolutional network; sequence matching; LSH; SLAM

---

## 1. Introduction

Visual-based vehicle localization in changing environments plays an important role in Simultaneous Localization and Mapping (SLAM) as well as the Advanced Driver Assistance Systems (ADAS) [1]. Unlike the LiDAR-based or GPS-based methods relying on expensive sensors, the vision-based localization approach using a low-cost camera is gaining popularity recently in the intelligent vehicle and robotics community [2,3]. In the vision-based localization approach, the current vehicle localization can be obtained by matching the image of the current location to an image of a previously visited position [4]. This is also known as place recognition, usually cast as an image retrieval task [5] where the current query image location is estimated using the locations of its retrieved image in a large geotagged image database. The core technique of appearance-based place recognition for visual localization is the representation of a place or location appropriately [6]. The huge appearance variations on the visual perception of a place caused by weather conditions or seasonal or illumination changes is a challenge for place-recognition-based visual localization in long-term driving.

At the early stages, place-recognition-based visual localization [7,8] mainly relies on the ad hoc features such as Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), or GIST. Through comparing these extracted features from each image (location), the re-visit location of a vehicle or robot is determined. These traditional feature extraction techniques have made progress in visual localization.

In recent years, with the deep learning development, an automatic and powerful image feature extractor—Convolutional Network (ConvNet)—achieved a state-of-the-art performance in the computer vision and machine learning communities [9,10]. The deep ConvNet with its strong image representation ability already achieved a high-level performance on visual recognition and classification tasks [11]. With training on a large-scale image data set, ConvNet obtained discriminative and human-interpretable feature representations, the learned features were robust, and the global features could be used for the task of visual localization without special training [12]. In long-term visual localization, two main problems need to be solved: false matching under changing environments (appearance or illumination) and the huge time cost for the high dimensional feature matching. Unlike the other deep-learning-based image recognition tasks using high dimensional features [11,13], we combine the robustness of sequence ConvNet features and the high-dimensional data reducing ability of Locality Sensitive Hashing (LSH) to develop an effective visual localization system for the long-term navigation of autonomous driving.

In this paper, a localized sequence-matching-based place recognition framework is developed. The proposed approach uses the ConvNet features and sequence matching to reduce the false recognition in long-term visual localization. In the whole visual recognition framework, ConvNet features are extracted based on a pretrained network first; then the extracted features are compared using the cosine distance. Finally, a localized sequence matching is conducted to retrieve the previous visited places based on the distance matrix. The main contributions of this paper are as follows: (1) the hierarchical nature of ConvNet features are exploited, and different ConvNet layers for place recognition under severe appearance and illumination variations are studied; (2) a comparison with state-of-the-art place recognition methods is performed on four data sets. The  $F_1$  (the harmonic average of the precision and recall) scores attained with the conv4 layer of ConvNet for the four different data sets are higher than 0.85, which are significantly better than those of Fast Appearance Based Mapping (FAB-MAP) and Sequence Simultaneous Localisation and Mapping (SeqSLAM); (3) for real-time visual localization, a speed-up method is achieved by approximating the cosine distance between features with a hamming distance over bit vectors obtained by Locality Sensitive Hashing (LSH), by using 4096 hash bits instead of the original feature permits to accelerate by 12 times the computation time, and by keeping 95% of the original place recognition performance.

The paper is organized as follows: Section 2 briefly reviews the existed visual recognition techniques and the progress of convolutional neural networks; Section 3 describes the proposed visual localization system; the experiment platform and used data sets are illustrated in Section 4; and the experimental results are demonstrated in Section 5. In Section 6, the conclusions and plans of future works are given.

## 2. Related Works

Many approaches related to place recognition have previously been proposed in the literature in the context of visual localization systems [14–16]. In this section, we briefly review the current state-of-the-art place recognition method related to visual localization and the application of convolutional neural networks for various visual recognition tasks.

### 2.1. Different Representations for Place Recognition

A keypoint of place-recognition-based localization approaches is the representation or description of a “place” (location) without the influence of lighting conditions or seasons changing [17,18]. This challenging question present in many computer vision and image searching applications already

leads up to numerous answers. In terms of place representation, the majority of methods can be classified into two categories: approaches based on a selective extraction of interesting or notable parts on the image (local features) and approaches based on the whole scene description (global features).

Local features, such as SIFT and SURF, have been widely used in appearance-based visual localization. The typical example is FAB-MAP [19]; it matches the appearance of the current location to a past place by converting the image into a bag-of-words representation built on local features such as SIFT or SURF. Beyond the FAB-MAP approach, SeqSLAM [20] using the sum of absolute differences (SAD) between the contrast-enhanced images [21] to measure image similarity. Then, a local-best matching over the predefined constant velocity path (i.e., the alignment between the query sequence and database sequences) is conducted to realize the place recognition under lighting and atmospheric variations. In addition, Badino et al. [22] present an outdoor visual localization approach based on a descriptor called Whole Image SURF (WI-SURF). Other local features like Local Binary Pattern (LBP) [23], Local Difference Binary (LDB) [24], and Binary Robust Independent Elementary Features (BRIEF) [25] can be also used in place recognition. However, an image description using binary descriptors may contain hundreds of local features, and features comparison is time-consuming.

In terms of global features, some researches describe the place to recognize in a holistic manner, using the whole-image (or global descriptors). A popular whole-image descriptor is GIST, which has been used for place recognition on a number of occasions [26]. Kosecka et al. [27] propose a place recognition strategy using gradient orientation histograms as image descriptors. Although global features are very fast to compute, they are not robust to effects such as lighting changes or perspective transformations [28].

Place recognition methods based on the above hand-crafted features are prone to being affected by a change in illumination or appearances. Nowadays, it is rapidly becoming apparent that, in recognition tasks, hand-crafted features are being outperformed by learned features [29]. It will be interesting and promising for visual localization to use automatically learned features thanks to the convolutional networks.

## 2.2. Convolutional Networks

Convolutional Network (ConvNet) is one of the popular deep neural networks and was firstly proposed by LeCun et al. [30] in 1989. ConvNet features are learned automatically from data sets through multi-layer supervised networks. ConvNets permit significant performance improvements on object classification or recognition to be achieved and outperform traditional hand-crafted features-based approaches [31].

Yan et al. [9] conducted a comprehensive performance comparison of the utility of features from all 21 layers for place recognition. In Reference [32], the AlexNet ConvNet model was trained on the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) for object recognition. Sünderhauf et al. [33] presented a novel place recognition system that was built on state-of-the-art object detection methods and convolutional visual features. The astonishing power of convolutional neural network features was used to identify matching landmark proposals between images to perform place recognition over extreme appearance and viewpoint variations. The experiment results have also revealed further insights: Mid-level ConvNet features appear to be highly suitable as descriptors for landmarks of various sizes in a place recognition context.

In addition, the availability of pretrained network models makes ConvNets easy to experiment about for different tasks. It therefore appears very promising to analyze these features and to experimentally investigate their feasibility for the task of visual recognition. The software packages Overfeat [34], Caffe [35], and MatConvNet [36] provide network architectures pretrained for a variety of recognition tasks. Especially, MatConvNet, an important ConvNet MATLAB toolbox designed with an emphasis on simplicity and flexibility, allows for the fast prototyping of new ConvNet architectures and supports efficient computation on CPUs and GPUs [36].

### 3. Proposed Approach

The proposed visual localization approach can be divided into off-line and online parts. In the off-line part, a set of GPS-tagged training images  $I^{train} = \{I_i^{train}\}_{i=1}^{N^{train}}$  is firstly acquired, where  $N^{train}$  is the number of training images. Then, the pretrained caffe-alex network (trained using the ILSVRC2012 data set) is used to extract features from training images [13]. The extracted ConvNet features from training database are noted  $F^{train} = \{f_i^{train}\}_{i=1}^{N^{train}}$ , where  $f_i^{train}$  is the feature extracted from the training image  $I_i^{train}$ .

For the online phase, the current testing image  $I_T^{test}$  is input into the caffe-alex network and the ConvNet feature  $f_T^{test}$  of the current testing image is computed. Then,  $f_T^{test}$  is compared with the training image feature set  $\{f_i^{train}\}_{i=1}^{N^{train}}$  using the cosine distance.

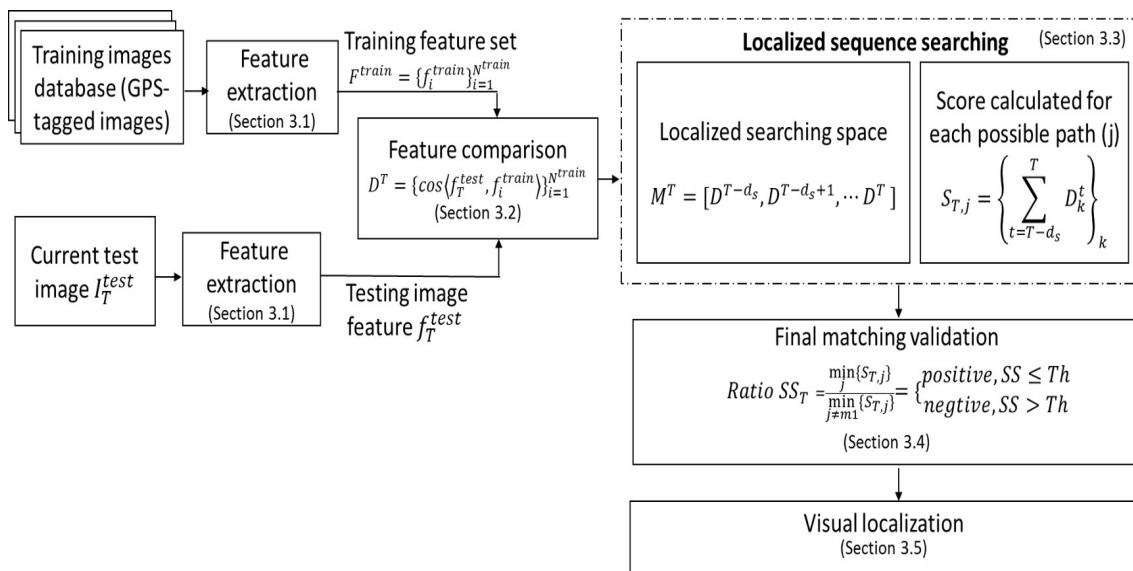
In terms of localized sequence matching, given a testing sequence of length  $d_s$  (a sequence composed of images indexed from  $T - d_s + 1$  to  $T$ , where  $T$  is the index of the current image), some possible training sequence candidates are firstly determined from the training database through the ratio between the testing and training trajectory speeds. For each possible sequence candidate, a score  $S$  is calculated by summing all the cosine distances along each sequence. The sequence candidate that provides the minimum score can be considered as the most similar one to the testing sequence. In fact, the two best sequences (according to the matching score) are conserved to further validate the final matching result.

Following, the best matching, the candidate will be validated through a distance ratio  $SS$  (see Section 3.4). This distance ratio  $SS$  between the two minimum computed distances (corresponding to the two best candidates) is considered to validate the training sequence that finally best matches to the current testing sequence. If the ratio  $SS$  is below or equal to a threshold  $Th$ , the first best sequence candidate (with the lower matching score) is confirmed and regarded as positive; otherwise, it is considered as a negative one (in this case, no matching result is conserved). When a sequence candidate is confirmed as positive, the position can be obtained from the matched GPS-tagged training images (see Section 3.5).

As illustrated in Figure 1, there are four important components in our visual localization approach:

- **ConvNet features extraction** (detailed in Section 3.1): ConvNet features  $F^{train}$  are extracted from all training database images by off-line processing, and  $f_T^{test}$  is extracted from the current testing image by online processing using the pretrained caffe-alex network. These learned features are robust to both appearance and illumination changes and represent each location (place) profoundly. The extracted ConvNet features will be compared in the next step.
- **Feature comparison** (detailed in Section 3.2): The cosine distances are computed between the feature  $f_T^{test}$  of the current testing image and the features  $\{f_i^{train}\}_{i=1}^{N^{train}}$  of all the images of the training database. All these distances form a vector  $D^T$ . Based on this, localized sequence matching is conducted in the next step.
- **Localized sequence matching** (detailed in Section 3.3): To achieve an efficient place recognition, localized sequence matching is used instead of single image matching. Considering the testing sequence composed of the last  $d_s$  testing images (indexed from  $T - d_s + 1$  to  $T$ ), localized sequence matching is conducted in the matrix  $M^T = [D^{T-d_s+1}, D^{T-d_s+2}, \dots, D^T]$ . According to the speed ratio between the testing and training sequences, some possible training sequence candidates in the training database can be firstly determined. A score  $S$  is calculated by summing all the testing image to training image cosine distances along each possible training sequence. The sequence that provides the minimum score can be considered the most similar one to the testing sequence. The two best sequence matching scores are conserved for further matching validation.
- **Final Matching Validation** (detailed in Section 3.4): The ratio  $SS$  between the two best sequence matching scores is used to verify the best sequence candidate. If the ratio  $SS$  is below or equal to a threshold  $Th$ , the first candidate (with the lower matching score) is confirmed and regarded as positive matching; otherwise, it is considered a negative one (in this case, no matching is conserved).

Several advantages of our approach can be highlighted: (1) The system uses an off-the-shelf pretrained convolutional network to extract features which makes feature extraction more convenient; (2) ConvNet features as auto-learned features are more stable and powerful. By using these robust features as descriptors for place representation, we inherit their robustness against appearance and weather changing; (3) using a localized sequence matching allows us to search in a small range rather than in the whole training database. This makes place recognition more robust and efficient.

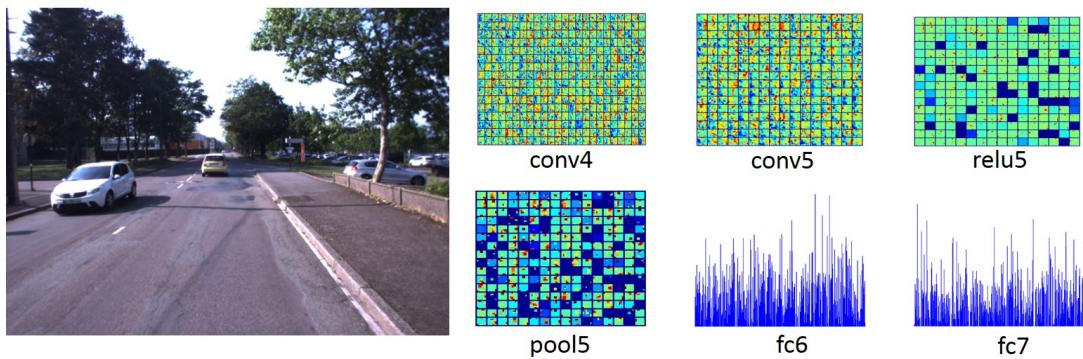


**Figure 1.** A detailed block diagram of the proposed visual localization method: Feature extraction uses a pretrained network, feature comparison uses the cosine distance, and localized sequence searching is conducted based on the potential path.

### 3.1. ConvNet Features Extraction

In this study, a pretrained caffe-alex [35] ConvNet model and MatConvNet toolbox are deployed to extract features. The caffe-alex ConvNet model is a 21-layer network; each layer output is a deep learnt representation of the image (ConvNet feature). The low layers retain a high spatial resolution for a precise localization with low-level visual information. While high layers capture more semantic information and less fine-grained spatial details. The network is able to process images of any size equal to or greater than  $227 \times 227$  pixels (the original caffe-alex network was trained on  $227 \times 227$  images). Place recognition is then performed by comparing the ConvNet features extracted from the current testing image  $I_T^{test}$  with the ConvNet features extracted from all the images  $\{I_i^{train}\}_{i=1}^{N^{train}}$  of the training database.

Considering that middle layers take the advantage of the low-level and semantic information, our approach exploits feature information of these middle layers to handle large appearance changes and then alleviate false recognition. The used layers and their dimensionality are listed in Table 1. The corresponding ConvNet features generated by convolutional Networks for an example of input image are illustrated in Figure 2. It can be seen that the conv4, conv5, and relu5 layers provide more image spatial information while the pool5, fc6, and fc7 layers bring more semantic information.



**Figure 2.** An example of scene and extracted features from different layers of the caffe-alex network. Features obtained from different ConvNet layers can serve as holistic image descriptors for place recognition.

**Table 1.** The layers from the caffe-alex ConvNet model used in our evaluation and their output dimensionality (height × width × feature map number).

| Layer | Dimensions                | Layer | Dimensions               |
|-------|---------------------------|-------|--------------------------|
| conv4 | $13 \times 13 \times 384$ | fc6   | $1 \times 1 \times 4096$ |
| conv5 | $13 \times 13 \times 256$ | fc7   | $1 \times 1 \times 4096$ |
| relu5 | $13 \times 13 \times 256$ |       |                          |
| pool5 | $6 \times 6 \times 256$   |       |                          |

### 3.2. Feature Comparison

The cosine distance is widely used to measure the feature vector similarity in the computer vision field; therefore, in our work, feature comparison is performed based on the cosine distance between the extracted ConvNet features. Each testing image feature is compared with all the images features of the training database. For that, the cosine distances between the feature  $f_T^{test}$  of the current testing image and the features  $\{f_i^{train}\}_{i=1}^{N^{train}}$  of all the images of the training database are computed as follows:

$$d_{T,i} = \cos\langle f_T^{test}, f_i^{train} \rangle = \frac{f_T^{test} \cdot f_i^{train}}{\|f_T^{test}\| \|f_i^{train}\|}; i = 1, 2, \dots, N^{train} \quad (1)$$

Then, these  $N^{train}$  distances are concatenated to form a  $D^T$  vector:

$$D^T = [\cos\langle f_T^{test}, f_1^{train} \rangle, \cos\langle f_T^{test}, f_2^{train} \rangle, \dots, \cos\langle f_T^{test}, f_{N^{train}}^{train} \rangle] \quad (2)$$

where  $N^{train}$  is the total number of images in the training database.  $D^T$  is the vector that contains the cosine distance between the testing image  $I_T^{test}$  and all the training images.

### 3.3. Localized Sequence Matching

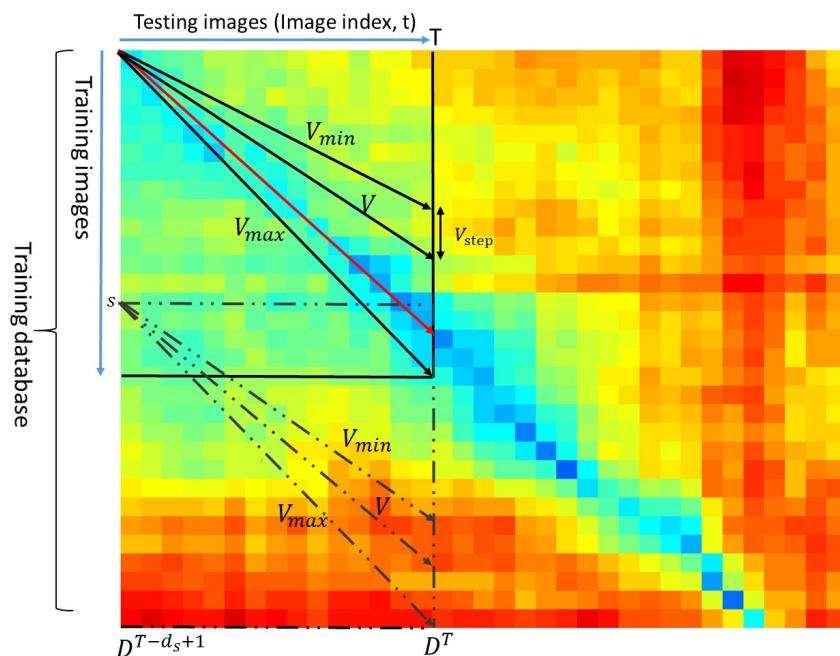
Assume that the vehicle repeatedly travels in a route with a negligible relative acceleration or deceleration. For a given testing sequence, composed of  $d_s$  images, indexed from  $T - d_s + 1$  to  $T$ , where  $T$  is the index of the current testing image, we search the sequence (from the training database) that corresponds to the current testing sequence. Rather than searching in the whole training database, the searching procedure is performed by considering possible training sequence candidates that are determined by the speed ratio between the training and testing sequences. This procedure is qualified as localized sequence searching.

At each time step, i.e., for each new testing image  $I_T^{test}$ , localized sequence searching is performed through a matrix  $M^T$  constructed by cosine distance vectors  $D^t (T - d_s + 1 \leq t \leq T)$  over the test sequence, composed of the  $d_s$  previous images (including the current testing image):

$$M^T = [D^{T-d_s+1}, D^{T-d_s+2}, \dots, D^T] \quad (3)$$

where  $d_s$  is the testing sequence length (in terms of images number) that determines how far back the search goes. As defined previously,  $D^t (T - d_s + 1 \leq t \leq T)$  is the cosine distance column vector for the current testing image  $I_t^{test}$ . It contains distances between the testing image feature  $f_t^{test}$  and all training image features  $\{f_i^{train}\}_{i=1}^{N_{train}}$ .

For testing sequence  $T$  (composed of  $d_s$  images, indexed from  $T - d_s + 1$  to  $T$ ), due to the linear relationship restriction of the testing and training driving speed, the corresponding training sequence candidates can be firstly confirmed using the speed ratio  $V (V_{min} \leq V \leq V_{max})$ . As shown in Figure 3, the possible paths representing different speed ratios can be projected onto each element in the matrix  $M^T$ . Thus, the lowest-cost path which has the minimum distance score  $S$  is deemed to be the best match, shown as the red line in Figure 3.



**Figure 3.** The search algorithm finds the lowest-cost straight line within the searching matrix  $M^T$ . These lines are the set of potential paths through the matrix. The red line is the lowest-cost path which aligns the testing sequence and the training sequence. Each element represents the cosine distance between two images.

In Figure 3, each element of the matrix  $M^T$  is the cosine distance between a testing image and a training image. A blue color in the matrix  $M^T$  indicates a small distance value while a red color means a large distance value. Searching ranges are constrained into the space between the minimum speed ratio  $V_{min}$  and the maximum speed ratio  $V_{max}$ . Each possible path (dark line) in the space indicates a possible match between the testing (query) sequence and the training sequence. The lowest-cost path (red line) is regarded as the best matching.

A score  $S$  is calculated for each path based on the distance values corresponding to the matrix components through which the straight-line passes from frame number  $T - d_s + 1$  to the current frame  $T$ :

$$S = \sum_{t=T-d_s+1}^T D_{k(t)}^t \quad (4)$$

where  $k(t)$  is the index of the column vector  $D^t$  by which the path (line) passes through:

$$k(t) = s + V(t - (T - d_s + 1)) \quad (5)$$

where  $s$  is the training image index from which the path is originated. The initial value of  $s$  is 0 and then is increased by 1 at each step.  $V$  is the speed ratio varying between  $V_{min}$  and  $V_{max}$  with a step value  $V_{step}$ . The score  $S$  (sum of distance values along path (line)) is used to identify the best matching candidate (the one that has the lowest score) for each testing sequence.

### 3.4. Final Matching Validation

Given the current testing image number  $T$ , the corresponding testing sequence (images indexed from  $T - d_s + 1$  to  $T$ ) can be constructed. Using the localized sequence matching method, the best two sequence candidates who have smaller scores are conserved for further validation. Suppose  $S_{m1}$  and  $S_{m2}$  are respectively the first and second minimum scores of the top two training sequence candidates to the testing sequence obtained by the following equation:

$$\begin{cases} S_{T,m1} = \min_j \{S_{T,j}\} \\ S_{T,m2} = \min_{j(j \neq m1)} \{S_{T,j}\} \end{cases} \quad (6)$$

where  $j$  is the index of the training sequence candidates. In order to validate the best sequence matching, a ratio  $SS_T$  is calculated as follows:

$$SS_T = \frac{S_{T,m1}}{S_{T,m2}} \quad (7)$$

The value of ratio  $SS$  is between 0 and 1. A threshold  $Th$  is then applied to the ratio  $SS_T$  to determine if the sequence pair  $(T, m1)$  is matched or not. If the ratio  $SS_T$  is not larger than the threshold  $Th$ , the training sequence corresponding to  $m1$  is matched to the current testing sequence; this is also called positive matching. Otherwise, no matching is considered (negative matching).

### 3.5. Visual Localization

After a matching result is successfully validated, the vehicle can localize itself through the matched training image position. Since the training images are tagged with GPS information, the vehicle can get its position information through the training image matched with the current testing image. This is also a topological level localization—it simply identifies the most likely location. Therefore, this is not a very accurate localization because the training and testing trajectory cannot be exactly the same.

### 3.6. Algorithm of Proposed ConvNet-Based Visual Localization

Algorithms 1 and 2 illustrate the proposed method for visual localization using localized sequence matching. It includes ConvNet feature extraction and comparison, localized sequence matching, matching validation, and visual localization steps. Algorithm 1 shows how to conduct a ConvNet feature extraction and comparison, while Algorithm 2 gives localized sequence matching methods and the final visual localization results.

**Algorithm 1** ConvNet feature extraction and comparison.

---

**Inputs:**

- $\{I_i^{train}\}_{i=1}^{N^{train}}$  {training images database};
- $\{I_i^{test}\}_{i=1}^{N^{test}}$  {testing images database};
- $N^{train}, N^{test}$  {training and testing images numbers};

**Outputs:**

$D$  {Cosine distance};

**Algorithm:**

```

for  $i \leftarrow 1$  to  $N^{test}$  do
    for  $j \leftarrow 1$  to  $N^{train}$  do
         $f_j^{train} \leftarrow$  Feature extraction for training images;
         $f_j^{test} \leftarrow$  Feature extraction for testing images;
         $d_{i,j} \leftarrow \cos(f_j^{test}, f_j^{train})$ ; // Cosine distance [Section 3.2].
    end for
     $D^i \leftarrow [d_{i,1}, d_{i,2}, \dots, d_{i,N^{train}}]$ ; Column vector  $D^i \in \mathbb{R}^{N^{train} \times 1}$  that contains the cosine distance between the testing image  $I_i^{test}$  and all the training images [Section 3.2].
end for

```

---

**Algorithm 2** Localized sequence matching and visual localization.

---

**Inputs:**

$\{D\}$  {Cosine distance};  
 $N^{train}, N^{test}$  {training and testing images numbers};  
 $V_{max}, V_{min}$  {maximum and minimum speed ratios};  
 $V_{step}$  {Vehicle speed step-size}  
 $d_s$  {Sequence length} ;

**Outputs:**

$S$  {Path-line (sequence candidate) score};

```

for  $T \leftarrow d_s$  to  $N^{test}$  do
     $M^T \leftarrow [D^{T-d_s+1}, D^{T-d_s+2}, \dots, D^T]$ ; // Local searching matrix.
     $j \leftarrow 1$ ; // Path number (sequence candidates number) initialization.
    for  $s \leftarrow 0$  to  $(N^{train} - V_{max} \times d_s)$  do
        for  $V \leftarrow V_{min}:V_{step}:V_{max}$  do
             $S_{T,j} \leftarrow 0$ ;
            for  $t \leftarrow (T - d_s + 1)$  to  $T$  do
                 $k(t) \leftarrow s + V(t - (T - d_s + 1))$ ; //  $k$  is a line index in the column vector  $D^t$ ;  $s$  is the training image number where the path originated in.
                 $S_{T,j} \leftarrow S_{T,j} + D_{k(t)}^t$ ; // Score  $S$  is calculated for each possible path.
            end for
             $j \leftarrow j + 1$ ; Sequence candidate number updating.
        end for
    end for
     $SS_T = \frac{\min_j \{S_{T,j}\}}{\min_{j(j \neq m1)} \{S_{T,j}\}}$ ;  $m1$  is the index of minimum score.
    if  $SS_T \leq Th$ 
        Matching validation is positive;
        Vehicle position  $\leftarrow$  The matched training image position
    if  $SS_T > Th$ 
        Matching validation is negative;
        Vehicle position  $\leftarrow$  NaN (no position results)
    end for

```

---

## 4. Experimental Setup

### 4.1. Experimental Platform

To evaluate the effectiveness of our approach, the proposed approach tests two changing data sets: one is our own and the other is an open public data set. Our data were acquired by an experimental GEM (Global Electric Motorcars) vehicle equipped with a stereoscopic Bumblebee XB3 camera system (16 Hz, 1280 × 960 image size), a RTK-GPS receiver (10 Hz), and two SICK LMS221 laser range finders, as shown in Figure 4.



**Figure 4.** The experimental vehicle equipped with sensors (camera and RTK-GPS).

### 4.2. Data Sets and Ground Truth

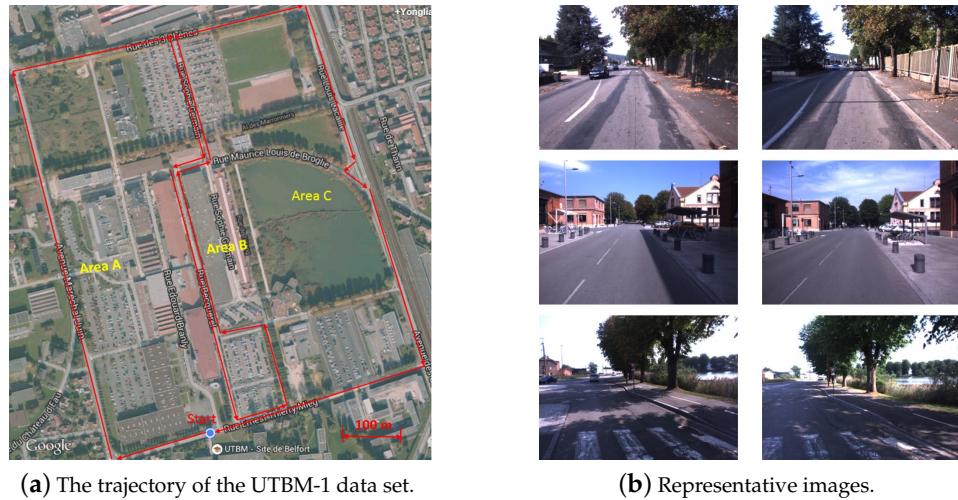
Four data sets with different characteristics (as described in Table 2) will be used to evaluate our method.

**Table 2.** Descriptions of the main characteristics of the data sets employed in the experiments.

| Data Set    | Length     | No. Images                  | Description                                      |
|-------------|------------|-----------------------------|--|
| UTBM-1      | 2 × 4.0 KM | training: 848; testing: 819 | minor variations in appearance and illumination  |
| UTBM-2      | 2 × 2.3 KM | training: 540; testing: 520 | medium variations in appearance and illumination |
| Nordland    | 4 × 728 KM | 4 × 3568                    | severe variations in appearance                  |
| City Center | 2 × 2.0 KM | 2 × 1237                    | medium variations in viewpoint                   |

#### (1) UTBM-1 data set

In the UTBM-1 data set, the experimental traversed about 4 km in an urban area. As illustrated in Figure 5a, the driving trajectory crossed three typical areas: urban city (area A), factory district (area B) and a natural scene place (area C). Some representative examples of the UTBM-1 data set are shown in Figure 5b. From this figure, the changing of shadow, vegetation, and field of view between the testing and training images can be also seen. The training and testing data were collected respectively on 11 September 2014 and 5 September 2014. Among all the acquired images (at about 16 Hz), only a subset of images were selected to perform matching between the training and testing data sets (848 images for training trajectory and 819 images for testing trajectory). The average distance interval between two selected frames was around 3.5 m. To tag the reference images (training database), each image was associated with its GPS position obtained by an RTK-GPS receiver.



(a) The trajectory of the UTBM-1 data set.

(b) Representative images.

**Figure 5.** The trajectory of the UTBM-1 data set and its representative images: (a) The trajectory acrosses forest, city, and parking areas. (b) Three representative examples of appearance and shadow variations. The images in each row are taken in the same place at different times (interval time of one week).

### (2) UTBM-2 data set

The data set UTBM-2 was collected on 5 September 2014 when the vehicle traversed on a 2.3 km long route in the urban city of Belfort. The trajectory and some image examples can be seen in Figure 6. The two traversals of this data set were conducted in the morning and afternoon. As shown in Figure 6b, there was a huge illumination variation between the images of training and testing. A total of 1060 images were used for the twice traveling (540 and 520 images for two traversals respectively).



(a) The trajectory of the UTBM-2 data set.

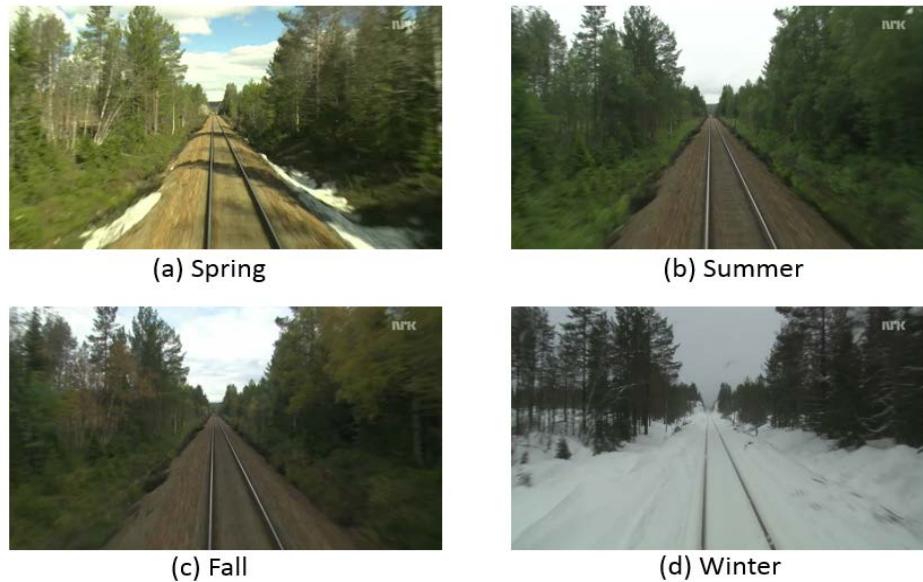
(b) Representative images.

**Figure 6.** The trajectory of the UTBM-2 data set and its representative images: (a) The trajectory acrosses forest, city, and parking areas. (b) Two representative examples of illumination variations. The images in each row were taken in the same place at different times (morning vs. afternoon).

### (3) Nordland data set

The Nordland data set has four video footages of a 728-km long train ride taken in northern Norway in four seasons [37]. As demonstrated in Figure 7, there was a huge appearance variation between the four seasons due to seasonal changing. The different landscape (plants and mountains) and local weather conditions (i.e., sunshine, clouds, rain, and snowfall) were experienced on the long trip. The original videos were recorded at 25 fps with a resolution of  $1920 \times 1080$ . GPS readings were recorded in conjunction with the video at 1 Hz. The full-HD recordings have been time-synchronized such that the position of the train in an arbitrary frame from one video correspond to the same frame

in any of the other three videos. In our experiment, frames were extracted from the original videos at 0.1 fps.



**Figure 7.** Four representative examples of the Nordland data set (each image corresponds to a different season).

#### (4) City Center data set

The City Center data set was collected by Mobile Robotics Group of the University of Oxford [38]. The robot traveled twice in one day around a loop with a total path length of 2 km; a total of 2474 images were collected while traveling. This data set was collected on a windy day with bright sunshine, which makes the abundant foliage and shadow features unstable, as can be observed in Figure 8.

For all four data sets, the ground truth was constructed by manually finding pairs of frame correspondences based on the GPS position. A match is considered as a true positive when it is within 1~3 frames of the ground truth (depending on the frame rate of the recorded data set), otherwise the match is considered a false positive.



**Figure 8.** City Center data set [38]: twice traveling. The left column shows the training images, and the right column shows the testing images.

#### 4.3. Performance Evaluation

Precision-recall characteristics are widely used to evaluate image retrieval abilities. Therefore, our evaluation methodology is based on precision-recall curves and  $F_1$  scores. The final curve is computed by varying the threshold  $Th$  in a linear distribution between 0 and 1 and by calculating the corresponding values of precision and recall.

In our experiments, the training image number is larger than or equal to the testing images number; thus, each testing image has a ground-truth match. Therefore, among the positives, there are only true positives (correct results among successfully validated image matching candidates) and false positives (wrong results among successfully validated image matching candidates). The sum of the true positives and false positives is the total retrieved images number.

More specifically, precision is the ratio of true-positives over the retrieved image numbers (the number of all the successfully validated image matching candidates), and recall is the ratio of true-positives over the total testing images. One hundred threshold values are processed to obtain well-defined curves. A perfect system would return a result where both precision and recall have a value of one. Based on the precision and recall,  $F_1$  score can be defined as

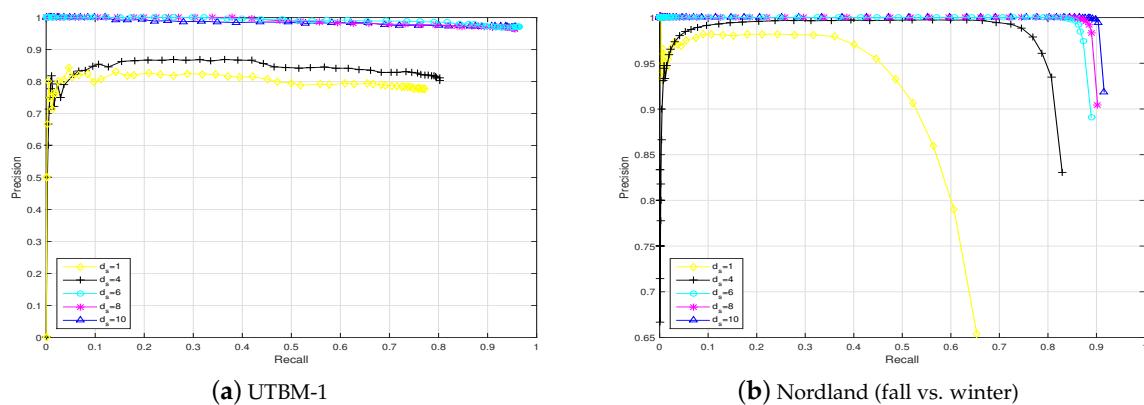
$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (8)$$

## 5. Experimental Results

### 5.1. Performance Comparison between Single Images and Sequences Based Approach

Traditionally, visual localization has been performed by considering places as single images. However, other more recent proposals, such as SeqSLAM, changed this concept and introduced the idea of recognizing places as sequences of images.

In this section, the place recognition performances based on sequences of images and single images are compared. In Figure 9, the results obtained for the UTBM-1 and Nordland data sets are presented. Attending to the precision-recall curves depicted in Figure 9, the influence of the sequence length ( $d_s$ ) is decisive to improving the performance of visual localization in life-long conditions. It can be clearly found that the approach using sequences allows for better results than those of a single image (almost no recall at 100% precision) to be achieved in long-term visual localization.



**Figure 9.** Two examples of a performance comparison of our proposal depending on the image sequence length ( $d_s$ ) in the challenging UTBM-1 and Nordland data sets (fall vs. winter). The feature used here is the conv4 layer.

Furthermore, there is a limit near a length of 8 for the UTBM-1 data set and a length of 6 for the Nordland data set from which the results are not greatly enhanced. Based on this sequence length

comparison and the driving speed, a sequence length of  $d_s = 8$  was chosen for data sets UTBM-1 and UTBM-2 in the rest of the experiments and results. For the City Center data set, the sequence length was set to 3, and that for the Nordland data set was 6. For all data sets, the speed ratio limits were  $V_{max} = 1.1$  and  $V_{min} = 0.9$ , and a step size of  $V_{step} = 0.04$  was set according to the experiment tests.

Figure 10 shows frame match examples on the Nordland data set (fall vs. winter). Despite the large appearance variations between different seasons, the proposed ConvNet-based visual localization using sequence matching attained better recognition results than those obtained using a single image.



**Figure 10.** Frame match examples from the Nordland (fall vs. winter) data sets. The top row shows a query sequence, and the middle and third rows show the frames recalled by  $d_s = 1$  (single image) and  $d_s = 6$ , respectively. Visual recognition based on sequence matching achieves a better performance than that of a single image.

## 5.2. Comparison of ConvNet Features Layer-By-Layer

This section provides a thorough investigation of the utility of different layers in the ConvNet hierarchy for place recognition and evaluates their individual robustness against the two main challenges in visual place recognition: appearance and illumination changes.

### 5.2.1. Appearance Change Robustness

(1) UTBM-1 data set: The interval time of training and testing data collection was one week. As illustrated in Figure 11 (top), the appearance between test and training images has minor changes and the viewpoint has medium variations. The precision-recall curves of place recognition are shown in Figure 11 (bottom). It can be seen that a recall obtained for the conv4 layer at a totally correct level is around 40%. While the performances of layers fc6 and fc7 are poor.

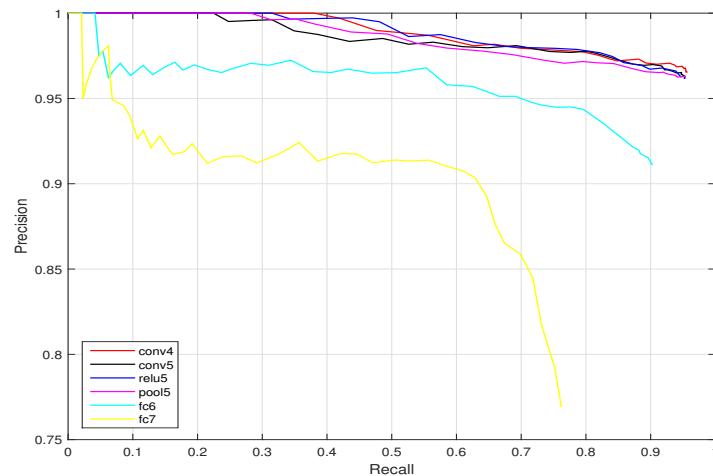
(2) City Center data set: This data set was acquired along public roads near the oxford city center with many dynamic objects such as traffic and pedestrians. The precision-recall curves are shown in Figure 12. Except that the recall at 100% precision of the layer fc7 is around 70%, the performance of the other layers (conv4, conv5, relu5, pool5, and fc6) reaches above a 75% recall at a totally correct level. The conv4 layer is the best one, achieving the highest recall level.

(3) Nordland data set: It is probably the longest (3000 km) that can be currently used for a life-long visual topological localization evaluation. It contains four videos with very strong seasonal appearance changes as depicted in Figure 7. The precision-recall curves of different cases (season for training vs. season for testing) are reported in Figure 13.

It can be seen that in the case of summer vs. fall, the performances obtained from the six layers (conv4, conv5, relu5, pool5, fc6, and fc7) are excellent (around 80% recall at the 100% precision level). For the other cases, conv4, conv5, relu5, and pool5 are more robust than the higher layers fc6 and fc7 considering appearance changes.



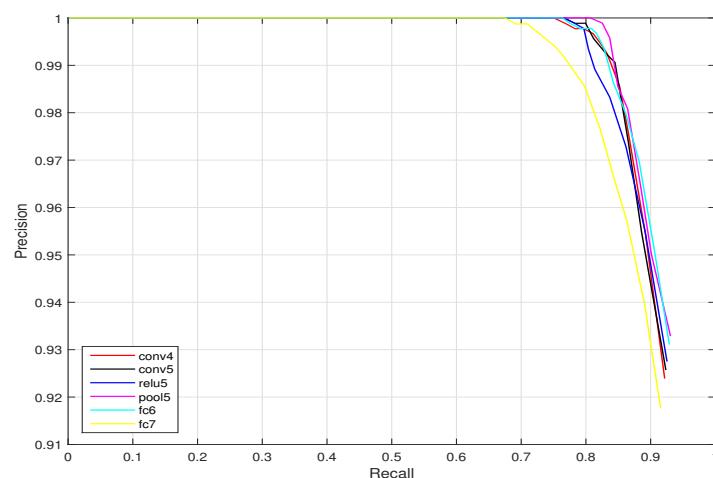
UTBM-1 data set: An example of training and testing images (interval time of one week).



**Figure 11.** Precision-recall curves for the UTBM-1 data set (the trajectory acrosses forest, city, and parking areas) ( $d_s = 8$ ).

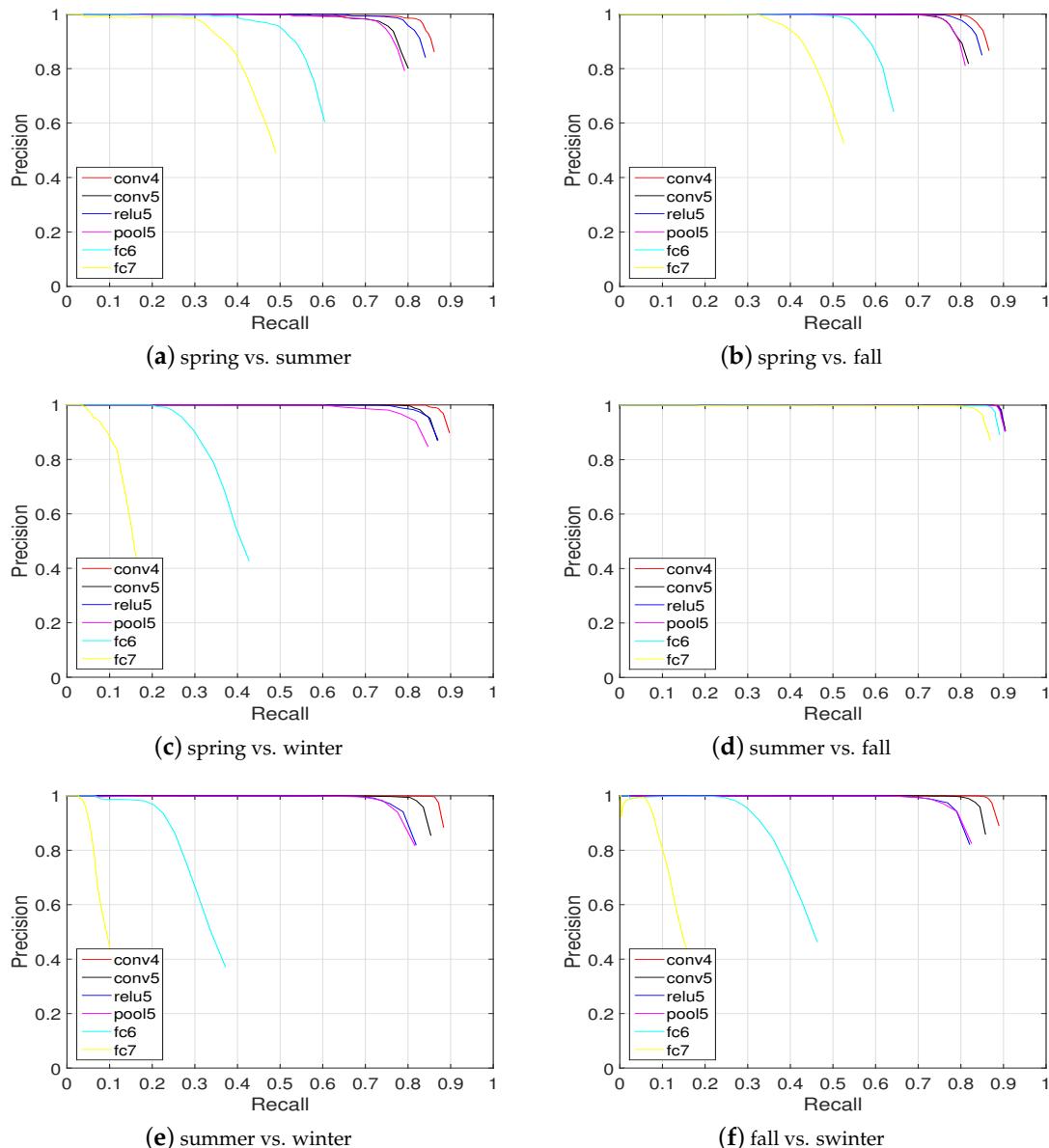


City Center data set [38]: twice traveling.



**Figure 12.** Precision-recall curves for the City Center data set (the trajectory acrosses city and parking areas) ( $d_s = 3$ ).

Figures 11–13 show the resulting precision-recall curves for appearance changing situations. The mid-level features from layers conv4 and relu6 are more robust against appearance changes than features from the other layers. Especially in the Nordland data set, the conv4 layer feature achieves above an 85% recall at the 100% correct level of all the tested cases. While the higher layers (i.e., fc6 and fc7) in the feature hierarchy lack robustness and exhibit an inferior place recognition performance.



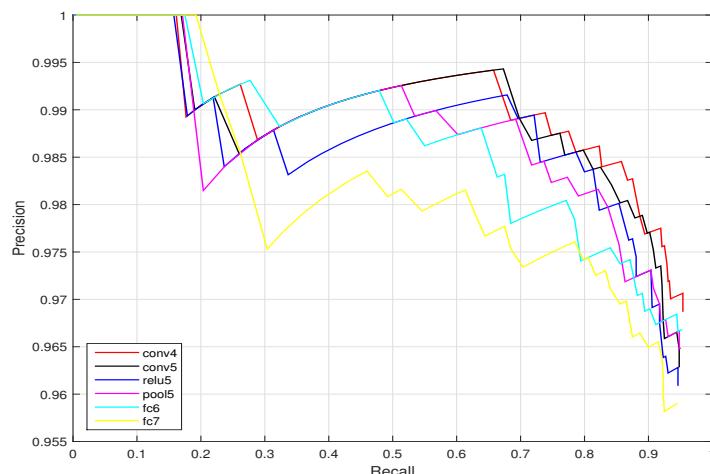
**Figure 13.** Place recognition across seasons on the Nordland data set. It can be seen that conv4 and conv5 perform better than the others, while fc6 and fc7 are the worst ( $d_s = 6$ ).

### 5.2.2. Illumination Change Robustness

Since illumination is another important influence factor in visual recognition, we investigate the ConvNet feature performances on the UTBM-2 data set that considers an illumination variation between morning and afternoon. The precision-recall curves are presented in Figure 14. As can be seen in Figure 14, the recall of ConvNet features from layer conv4 achieved is around 40%, which performs better than the other layer features (conv5, relu5, pool5, fc6, and fc7) to deal with the severe illumination changes. Since pretrained networks model were trained with good quality images, some layers do not show strong robust abilities in illumination variance situations.



UTBM-2 data set: morning vs. afternoon



**Figure 14.** Precision-recall curves for the UTBM-2 data set considering different ConvNet layers ( $d_s = 8$ ).

Table 3 shows  $F_1$  scores obtained for different layers and other state-of-the-art methods like SeqSLAM and FAB-MAP. For a SeqSLAM comparison, the OpenSeqSLAM code [20] was used and the same sequence lengths were taken as settled above. While the other parameters are set to default values as reported in Reference [20]. For a FAB-MAP comparison, the OpenFABMAP code [39] was used.

By comparing our approach to the SeqSLAM method under extreme appearance changes in Table 3, it can be found that the obtained  $F_1$  scores (the harmonic average of the precision and recall) based on ConvNet features extracted by layer conv4 are above 0.85, which exceeds the SeqSLAM and FAB-MAP performances. In addition, it can be noticed that the performance of the ConvNet feature extracted from the conv4 layer is better than the other layers, especially the higher layers (i.e., fc6 and fc7). It confirmed again that the middle layers such as conv4 contain more useful image information due to advantages of the spatial information from low layers and semantic information from high layers being taken.

**Table 3.**  $F_1$ -Scores considering different Caffe-AlexNet layers and other state-of-the-art methods (SeqSLAM and FAB-MAP). The † means the  $F_1$  score is smaller than 0.01, and ‡ means experiment fails. For each data set, the highest  $F_1$ -Score is bold.

| Dataset     | Caffe-Alex Layers |               |        |               |        |        | SeqSLAM | FAB-MAP |
|-------------|-------------------|---------------|--------|---------------|--------|--------|---------|---------|
|             | conv4             | conv5         | relu5  | pool5         | fc6    | fc7    |         |         |
| Norland     | spring vs. summer | <b>0.8967</b> | 0.8427 | 0.8734        | 0.8354 | 0.6722 | 0.5455  | 0.7222  |
|             | spring vs. fall   | <b>0.8984</b> | 0.8572 | 0.8821        | 0.8579 | 0.7098 | 0.5859  | 0.7015  |
|             | spring vs. winter | <b>0.9255</b> | 0.8987 | 0.8983        | 0.8750 | 0.4795 | 0.2387  | 0.6685  |
|             | summer vs. fall   | <b>0.9396</b> | 0.9381 | 0.9388        | 0.9375 | 0.9286 | 0.9047  | 0.6960  |
|             | summer vs. winter | <b>0.9245</b> | 0.8935 | 0.8581        | 0.8497 | 0.4142 | 0.1817  | 0.5117  |
|             | fall vs. winter   | <b>0.9288</b> | 0.8922 | 0.8598        | 0.8599 | 0.5119 | 0.2337  | 0.5293  |
| UTBM-1      | <b>0.9607</b>     | 0.9576        | 0.9576 | 0.9583        | 0.9607 | 0.7762 | 0.7222  | 0.2356  |
| UTBM-2      | <b>0.9622</b>     | 0.9564        | 0.9544 | 0.9574        | 0.9593 | 0.9516 | 0.7180  | 0.4813  |
| City Center | 0.9288            | 0.9246        | 0.9264 | <b>0.9317</b> | 0.9299 | 0.9166 | †       | 0.5326  |

### 5.3. Local Sensitive Hashing for Real-Time Place Recognition

In contrast to typical computer vision benchmarks where the recognition accuracy is the most important performance metric, visual localization for vehicles or robots always needs agile algorithms for real-time application [40]. In the above studies, conv4 has shown its strong ability in place recognition. However, computing the cosine distance between many 64,896 dimensional conv4s is an expensive operation since Locality Sensitive Hashing (LSH) is arguably the most popular feature compression method and widely used in the field of information retrieval and computer vision [41]. For a fast place recognition, the LSH method is used, which maps the conv4 feature  $f_{conv4}$  to a low-dimensional binary vector:

$$H(K) = \text{sign}(w^\top f_{conv4} + b) \quad (9)$$

where  $w$  is a  $K$  dimension data-independent random matrix, which satisfies a standard Gaussian distribution [41] and  $b$  is a random intercept. In our experiment, the conv4 feature  $f_{conv4}$  is normalized with a zero mean; then, an approximately balanced partition is obtained with  $b = 0$ . Thus, the high dimension feature is converted into a low  $K$  dimension binary bits. The binary bit vectors can then be compared using the hamming distance more efficiently.

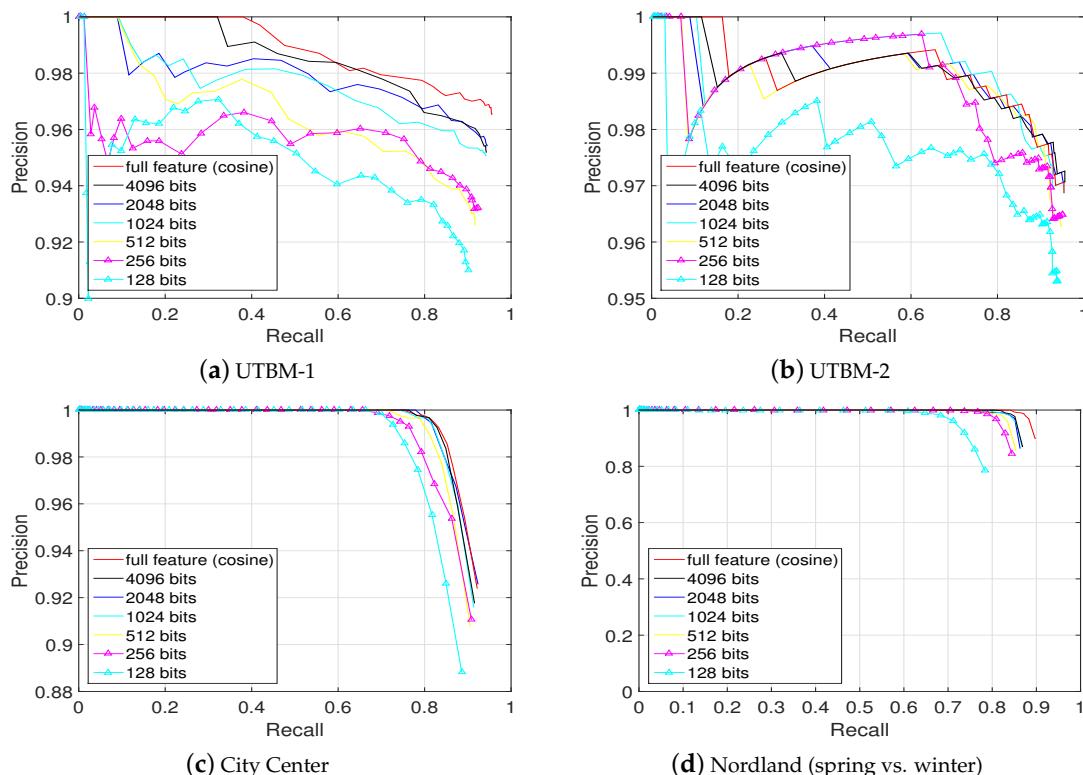
In Figure 15, the place recognition performance achieved with the hashed conv4 feature vectors of different hash bit lengths ( $2^8 \dots 2^{12}$  bits) on the four data sets is compared. Hashing the original 64,896 dimensional vectors into 4096 hash bits corresponds to a data compression of 63.1%. In addition, the 4096-hash-bit representation retains approximately 95% of the original place recognition performance. It can be seen that, from Figure 15, when the length of hash bits is decreasing, the place recognition performance is also descending.

Table 4 shows the  $F_1$  scores of different hash bit lengths achieved in four data sets. The average times per matching are also presented. The experiments are conducted on a laptop machine with intel i7-4700MQ CPU and 32 Gb RAM.

As shown in Table 4, the average time per matching using 4096 hash bits is 0.0291 s which corresponds almost to a speed-up factor of 12 compared to using the cosine distance over the original conv4 feature requiring 0.3259 s per matching. Compared with the full feature matching, using the 4096 hash bits representing the original full feature permits the matching to be faster and easier. There is no doubt that, for larger scale data sets, the speed-up advantages can be more significant.

**Table 4.**  $F_1$ -scores and matching time comparison of different lengths of hash bits. For each data set, the highest  $F_1$ -Score is bold.

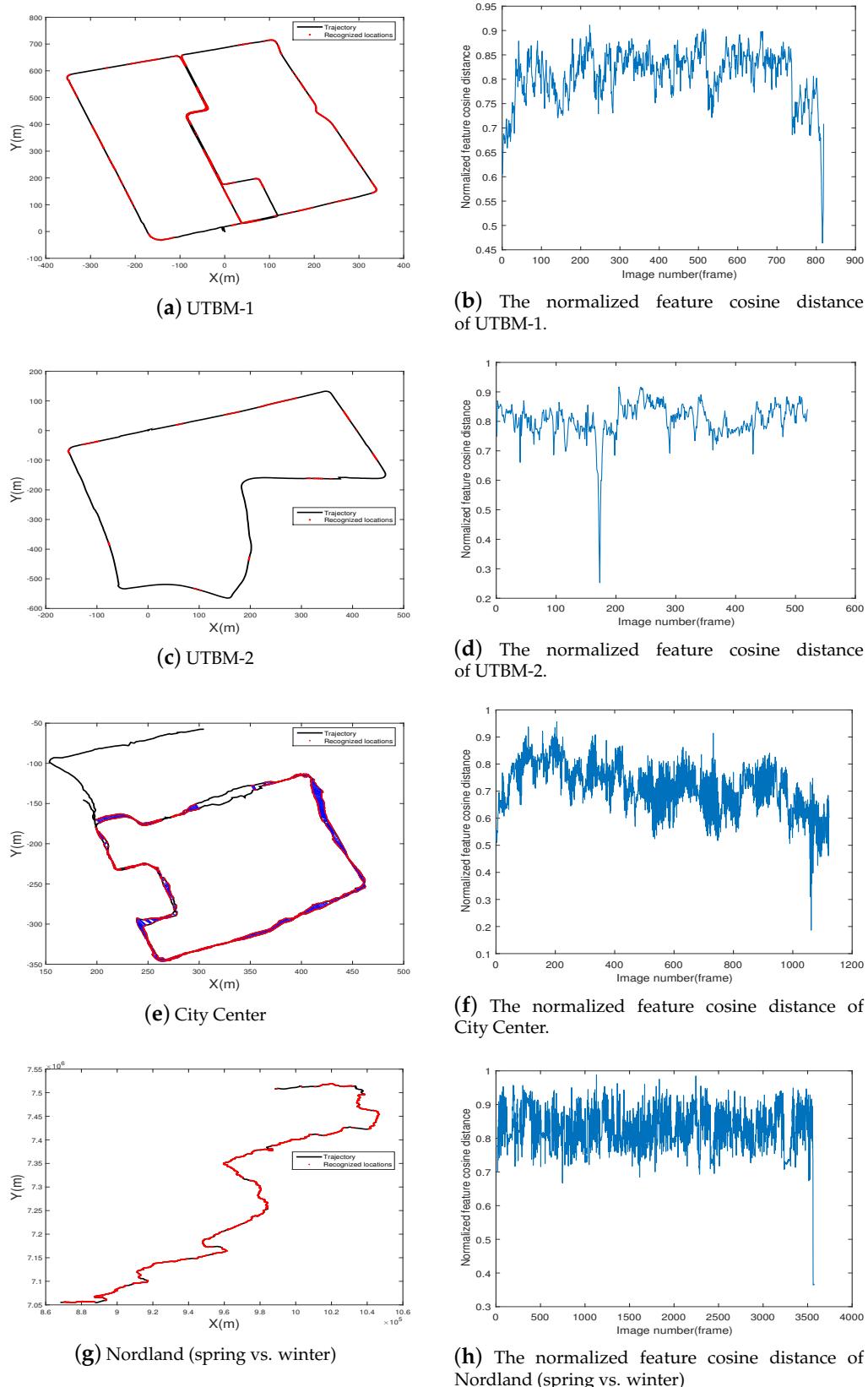
| Method               | $F_1$ Scores  |               |               |                                 | Average Time Per Matching<br>(All Data Sets) |
|----------------------|---------------|---------------|---------------|---------------------------------|--|
|                      | UTBM-1        | UTBM-2        | City Center   | Nordland<br>(Spring vs. Winter) |  |
| 256 bits             | 0.9411        | 0.9574        | 0.9094        | 0.8817                          | 0.0135 s                                     |
| 512 bits             | 0.9478        | 0.9554        | 0.9084        | 0.8944                          | 0.0147 s                                     |
| 1024 bits            | 0.9460        | 0.9612        | 0.9162        | 0.9046                          | 0.0170 s                                     |
| 2048 bits            | 0.9521        | 0.9632        | 0.9246        | 0.9064                          | 0.0209 s                                     |
| 4096 bits            | 0.9521        | <b>0.9641</b> | 0.9166        | 0.9099                          | 0.0291 s                                     |
| Full feature (conv4) | <b>0.9607</b> | 0.9622        | <b>0.9228</b> | <b>0.9255</b>                   | 0.3259 s                                     |



**Figure 15.** Precision-recall curves of different hash bit lengths. The cosine distance over the full feature vector of 64,896 dimensions (red) can be closely approximated by the hamming distance over bit vectors of length 4096 (dark) without losing much performance. This corresponds to a compression of 63.1%.

#### 5.4. Visual Localization Results

For visual localization based on place recognition, the recognition rate at a high precision level is a key indicator in reflecting whether the system is robust enough to determine the position under a changing environment. A correct place recognition means a successful visual localization, while an incorrect place recognition could cause a huge localization error. Therefore, the higher the recognition rate at a 100% precision is, the more robust the visual localization system is. Figure 16 shows the final place-recognition-based visual localization results for the different data sets at a precision level of 100%. Regardless of the appearance and illumination changes, the proposed method can still localize the vehicle in most places.



**Figure 16.** The visual localization results in the four data sets. The used feature is 4096 hash bits of the conv4 layer. In the left column, two images from the same location (on the basis of appearance alone) are marked with red points and joined with a blue line. In the right column are the corresponding normalized feature cosine distances.

In Table 5, recall ratios at different precision levels are given. Using 4096 hash bits at 100% precision, the proposed approach achieves above a 75% recall on the City Center data set and above 72.88% on the more challenging Nordland data set (Spring vs. Winter), while on the UTBM-1 and UTBM-2 data sets, the recall are 32.88% and 11.54% respectively.

**Table 5.** The recall results at 100% precision. The † means the accuracy is smaller than 10%.

| Dataset     | Recall Results at 100% Precision |                |         |         |
|-------------|----------------------------------|----------------|---------|---------|
|             | Full Feature (conv4)             | 4096 Hash Bits | FAB-MAP | SeqSLAM |
| Norland     | spring vs. summer                | 57.09          | 69.02   | † 45.71 |
|             | spring vvs. fall                 | 64.66          | 67.26   | † 33.91 |
|             | spring vs. winter                | 76.77          | 72.88   | † 35.53 |
|             | summer vs. fall                  | 86.88          | 87.67   | † 47.89 |
|             | summer vs. winter                | 60.47          | 28.26   | † 22.84 |
|             | fall vs. winter                  | 82.16          | 79.65   | † 15.82 |
| UTBM-1      | 37.97                            | 32.88          | †       | 20.16   |
| UTBM-2      | 16.35                            | 11.54          | †       | 8.53    |
| City Center | 75.04                            | 76.29          | 31.78   | 52.63   |

## 6. Conclusions and Future Works

In this paper, a visual vehicle localization approach based on ConvNet features and localized sequence matching is proposed. The approach takes advantages of the ConvNet image representation ability and localized sequence matching, which make place recognition fast and accurate. We also compared the proposed approach with state-of-the-art methods (SeqSLAM and FAB-MAP) on four typical data sets that consider big challenges in visual place recognition: appearance and illumination changes. The experimental results showed that ConvNet feature conv4 can achieve a good performance with above a 0.89  $F_1$  score. In addition, for satisfying real-time constraints, the speed-up approach based on the LSH method was used to compress the high dimension of ConvNet features. By using the 4096 hashing bits representation to replace the original conv4 feature, each matching process was almost 12 times faster. The proposed visual localization approach allowed the vehicle to localize itself in changing environments.

In future work, we will extend our understanding of ConvNet features and explore how to use transfer learning to service the place recognition. Additionally, we will also considering how to train a specific end-to-end CNN for a life-long visual localization under changing conditions.

**Author Contributions:** Y.Q. performed the experiments, analyzed the data, and wrote the paper; C.C., Y.R., and T.Y. participated in the paper preparation and revision.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

|         |  |
|---------|--|
| ConvNet | Convolutional Network                          |
| LSH     | locality sensitive hashing                     |
| FAB-MAP | Fast Appearance Based Mapping                  |
| SeqSLAM | Sequence Simultaneous Localisation and Mapping |

## References

1. Rivera-Rubio, J.; Alexiou, I.; Bharath, A.A. Appearance-based indoor localization: A comparison of patch descriptor performance. *Pattern Recognit. Lett.* **2015**, *66*, 109–117. [[CrossRef](#)]

2. Lin, S.; Cheng, R.; Wang, K.; Yang, K. Visual localizer: Outdoor localization based on convnet descriptor and global optimization for visually impaired pedestrians. *Sensors* **2018**, *18*, 2476. [[CrossRef](#)] [[PubMed](#)]
3. Qiao, Y.; Cappelle, C.; Ruichek, Y. Visual localization across seasons using sequence matching based on multi-feature combination. *Sensors* **2017**, *17*, 2442. [[CrossRef](#)] [[PubMed](#)]
4. Herranz, L.; Jiang, S.; Li, X. Scene recognition with CNNs: Objects, scales and data set bias. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 571–579.
5. Yuan, B.; Tu, J.; Zhao, R.W.; Zheng, Y.; Jiang, Y.G. Learning part-based mid-level representation for visual recognition. *Neurocomputing* **2018**, *275*, 2126–2136. [[CrossRef](#)]
6. Li, Q.; Li, K.; You, X.; Bu, S.; Liu, Z. Place recognition based on deep feature and adaptive weighting of similarity matrix. *Neurocomputing* **2016**, *199*, 114–127. [[CrossRef](#)]
7. Garcia-Fidalgo, E.; Ortiz, A. Vision-based topological mapping and localization methods: A survey. *Robot. Auton. Syst.* **2015**, *64*, 1–20. [[CrossRef](#)]
8. Ouerghi, S.; Boutteau, R.; Savatier, X.; Tlili, F. Visual odometry and place recognition fusion for vehicle position tracking in urban environments. *Sensors* **2018**, *18*, 939. [[CrossRef](#)] [[PubMed](#)]
9. Chen, Y.; Shen, Y.; Liu, X.; Zhong, B. 3D object tracking via image sets and depth-based occlusion detection. *Signal Process.* **2015**, *112*, 146–153. [[CrossRef](#)]
10. Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN features off-the-shelf: An astounding baseline for recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, Columbus, OH, USA, 24–27 June 2014; pp. 806–813.
11. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 685–694.
12. Zhu, J.; Ai, Y.; Tian, B.; Cao, D.; Scherer, S. Visual Place Recognition in Long-term and Large-scale Environment based on CNN Feature. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Suzhou, China, 26–30 June 2018; pp. 1679–1685.
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*; Pereira, F., Burges, C., Bottou, L., Weinberger, K., Eds.; The MIT Press: Cambridge, MA, USA, 2012; pp. 1097–1105.
14. Li, L.; Goh, W.; Lim, J.H.; Pan, S.J. Extended Spectral Regression for efficient scene recognition. *Pattern Recognit.* **2014**, *47*, 2940–2951. [[CrossRef](#)]
15. Valiente, D.; Gil, A.; Payá, L.; Sebastián, J.; Reinoso, Ó. Robust visual localization with dynamic uncertainty management in omnidirectional SLAM. *Appl. Sci.* **2017**, *7*, 1294. [[CrossRef](#)]
16. Valiente, D.; Gil, A.; Fernández, L.; Reinoso, Ó. A comparison of EKF and SGD applied to a view-based SLAM approach with omnidirectional images. *Robot. Auton. Syst.* **2014**, *62*, 108–119. [[CrossRef](#)]
17. Song, X.; Jiang, S.; Herranz, L.; Kong, Y.; Zheng, K. Category co-occurrence modeling for large scale scene recognition. *Pattern Recognit.* **2016**, *59*, 98–111. [[CrossRef](#)]
18. Duan, Q.; Akram, T.; Duan, P.; Wang, X. Visual saliency detection using information contents weighting. *Optik* **2016**, *127*, 7418–7430. [[CrossRef](#)]
19. Cummins, M.; Newman, P. Appearance-only SLAM at large scale with FAB-MAP 2.0. *Int. J. Robot. Res.* **2011**, *30*, 1100–1123. [[CrossRef](#)]
20. Milford, M.; Wyeth, G. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, 14–18 May 2012; pp. 1643–1649.
21. Neubert, P.; Sunderhauf, N.; Protzel, P. Appearance change prediction for long-term navigation across seasons. In Proceedings of the European Conference on Mobile Robots (ECMR), Barcelona, Spain, 25–27 September 2013; pp. 198–203.
22. Badino, H.; Huber, D.; Kanade, T. Real-time topometric localization. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, 14–18 May 2012; pp. 1635–1642.
23. Qiao, Y.; Cappelle, C.; Ruichek, Y. Place Recognition Based Visual Localization Using LBP Feature and SVM. In Proceedings of the Mexican International Conference on Artificial Intelligence, Morelos, Mexico, 25–31 October 2015; Springer: New York, NY, USA, 2015; pp. 393–404.

24. Arroyo, R.; Alcantarilla, P.; Bergasa, L.; Yebes, J.; Bronte, S. Fast and effective visual place recognition using binary codes and disparity information. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago, IL, USA, 14–18 September 2014; pp. 3089–3094.
25. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary robust independent elementary features. In Proceedings of the European Conference on Computer Vision, Crete, Greece, 5–11 September 2010; Springer: New York, NY, USA, 2010; pp. 778–792.
26. Liu, Y.; Zhang, H. Visual loop closure detection with a compact image descriptor. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 1051–1056.
27. Kosecka, J.; Zhou, L.; Barber, P.; Duric, Z. Qualitative image based localization in indoors environments. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 16–22 June 2003; Volume 2, pp. II-3-II-8.
28. Chen, Z.; Lam, O.; Jacobson, A.; Milford, M. Convolutional Neural Network-based Place Recognition. *arXiv* **2014**, arXiv:1411.1509v1.
29. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
30. LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]
31. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5297–5307.
32. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
33. Sünderhauf, N.; Shirazi, S.; Dayoub, F.; Upcroft, B.; Milford, M. On the performance of ConvNet features for place recognition. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–3 October 2015; pp. 4297–4304.
34. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In Proceedings of the 2014 CBLS International Conference on Learning Representations (ICLR 2014), Banff, AB, Canada, 14–16 April 2014.
35. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.B.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
36. Vedaldi, A.; Lenc, K. MatConvNet: Convolutional Neural Networks for MATLAB. In Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, Brisbane, Australia, 26–30 October 2015; pp. 689–692.
37. Sünderhauf, N.; Neubert, P.; Protzel, P. Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Workshop on Long-Term Autonomy, Karlsruhe, Germany, 6–10 May 2013.
38. Cummins, M.; Newman, P. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *Int. J. Robot. Res.* **2008**, *27*, 647–665. [[CrossRef](#)]
39. Glover, A.; Maddern, W.; Warren, M.; Reid, S.; Milford, M.; Wyeth, G. OpenFABMAP: An open source toolbox for appearance-based loop closure detection. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, 14–18 May 2012; pp. 4730–4735.
40. Jacobson, A.; Chen, Z.; Milford, M. Autonomous Multisensor Calibration and Closed-loop Fusion for SLAM. *J. Field Robot.* **2015**, *32*, 85–122. [[CrossRef](#)]
41. Datar, M.; Immorlica, N.; Indyk, P.; Mirrokni, V.S. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the Twentieth Annual Symposium on Computational Geometry, Brooklyn, NY, USA, 9–11 June 2004; ACM: New York, NY, USA, 2014; pp. 253–262.

