Cross-scene loop-closure detection with continual learning for visual simultaneous localization and mapping

Shilang Chen¹, Junjun Wu¹, Qinghua Lu¹, Yanran Wang² and Zeqin Lin¹

Abstract
Humans maintain good memory and recognition capability of previous environments when they are learning about new ones. Thus humans are able to continually learn and increase their experience. It is also obvious importance for autonomous mobile robot. The simultaneous localization and mapping system plays an important role in localization and navigation of robot. The loop-closure detection method is an indispensable part of the relocation and map construction, which is critical to correct mappoint errors of simultaneous localization and mapping. Existing visual loop-closure detection methods based on deep learning are not capable of continual learning in terms of cross-scene environment, which bring a great limitation to the application scope. In this article, we propose a novel end-to-end loop-closure detection method based on continual learning, which can effectively suppress the decline of the memory capability of simultaneous localization and mapping system by introducing firstly the orthogonal projection operator into the loop-closure detection to overcome the catastrophic forgetting problem of mobile robot in large-scale and multi-scene environments. Based on the three scenes from public data sets, the experimental results show that the proposed method has a strong capability of continual learning in the cross-scene environment where existing state-of-the-art methods fail.

Keywords
Mobile robot, visual SLAM, loop-closure detection, continual learning

Introduction
Simultaneous localization and mapping (SLAM) is a key technology for autonomous mobile robots¹,² Visual-based SLAM has become a research hot spot because of the rich information acquired by visual sensors. The amount of training sample data learned by the visual SLAM system is always limited. So, the SLAM system encounters many problems, when the autonomous mobile robot works in a changeable real-world environment.³,⁴ For example, when the visual SLAM system trained in scene A works in scene B, the catastrophic forgetting problem of the neural network causes a greater suppression of the memory capability of the visual SLAM system. Meanwhile, the visual SLAM lose the memory of the original scene when it learns...
map construction in the new environment. With the larger scene spanning, the longer running time, and the more new sample data, the memory capability of the existing visual SLAM system declines more severely. This cause the robot to fail to effectively and incrementally complete the map construction and need to relearn when it encounters cross-scene.\(^5\)

When the visual SLAM faces the cross-scene environment, how to make it have human-like continual learning capability is the key to the robot’s practical application. As a key module of visual SLAM, loop-closure detection (LCD) plays a crucial role in improving the human-like learning capability, if it can continually learn new knowledge from new scenes without forgetting the memory of previous scenes. Therefore, it is of great significance to study the method of visual LCD with cross-scene learning capability.

At present, most of the LCD methods mainly use image descriptors to visually describe the environment and then complete LCD by matching the current image with the key-frames of the map. Compared with the LCD methods based on handcrafted descriptors,\(^6\)–\(^10\) the methods based on convolutional neural network (CNN) have significant advantages and also have received great attention.\(^11\)–\(^15\) However, all LCD methods based on CNN have the problem of catastrophic forgetting, that is, they almost gradually forget the previously acquired content after learning new knowledge, which makes LCD methods lack the persistent adaptation to the environment and continual learning like human beings.\(^16\)

Existing LCD methods need to change the training data and retrain the model according to different application scenes. Although the model trained in the new scene can work well in the new scene, it partially forgets the old scene. As a result, the more new scenes are learned, the worse the capability of the model to recognize the old scene will be, or even completely unable to recognize the old scene. Figure 1 shows the state-of-the-art method of LCD NetVLAD\(^17\) gradually lost its memory for old scenes after continually learning (a) Oxford night, (b) Pittsburgh, and (c) Oxford day.

In this article, a new cross-scene LCD method for visual SLAM is proposed to solve the problems of insufficient continual learning capability, loss of previously learned experience, and memory decline in terms of cross-scene learning. The proposed method learns image features by improving the learning strategy and CNN structure, then uses the improved NetVLAD to generate image descriptors with image feature aggregation, and finally adopts an efficient index structure to ensure the efficiency of the online LCD method. The most significant difference with the existing methods is that our method firstly adopts the parameter learning mechanism based on orthogonal weight modification, which effectively suppresses the memory decline of LCD.

The main contributions of this article include

1. We integrate the orthogonal weight modification theory into the CNN for the first time and propose a new deep learning mechanism. The proposed method enables deep neural networks to obtain the ability to learn image features across scenes. The improvement of this basic learning ability has greatly improved the image matching performance of the robot in different scenes, which is of great significance to tasks such as LCD and visual place recognition.

2. We propose a novel end-to-end LCD method, which has continual learning capability and generates more robust image descriptors. This method enables the nonlinear dynamic SLAM system to have the ability to build incremental maps across scene, which greatly improves the intelligence level of the SLAM system.

The rest of the article is organized as follows. We discuss the related work in the second section and introduce the proposed LCD method in the third section. The experiment is elaborated in detail in the fourth section and fifth section. In the sixth section, the work of the article is summarized and the future work is expected.

**Related work**

In this section, we briefly review the representative methods of LCD for visual SLAM and introduce the relationship between these methods and our work. Building a high-precision environmental map for visual SLAM is the most important and fundamental capability for its robust perception of the surrounding environment, and LCD is the key to correct mapping.\(^1\) The LCD method can reduce the
probability of incorrect representation of node of map and error accumulation brought by the front-end of visual SLAM, so as to obtain a globally consistent map. LCD is to determine whether the robot has returned to a location that has been visited before, and it is, in essence, the same as global localization and place recognition. So next we introduce these three aspects together.

Visual LCD can be roughly divided into two kinds of algorithms: those based on shallow features and those based on deep features. LCD based on shallow features mainly relies on handcrafted features which are mainly designed by expert experience. The GIST is a representative work based on the shallow features, which can describe the macroscopic feature of the whole image scene. This method does not require any form of pretraining and only inputs an image to get the global feature vector. Therefore, this method is listed as one of the representative methods for the comparison experiment in our article. With the development of the methods in recent years, many local-based invariance features have appeared, such as SIFT, ORB, and SURF. These methods can maintain a relatively stable matching accuracy in the face of camera rotation, translation, scale changes, and so on. They have also achieved certain effects in the SLAM system. But these local feature-based methods not only need to construct bag-of-words and they are also weak when facing more complex changing environment such as light changes.

To address these challenges, researchers find that methods based on deep learning can find stable feature regions in a large number of scene samples and show stronger robustness when the appearance of environment changes dramatically due to changes in environmental light and seasons. Therefore, researchers try to introduce deep learning into the field of visual SLAM for detecting loop-closure, and many representative achievements appeared. The method proposed by Lopez-Antequera et al. uses a training model to map images in a low-dimensional space, so that images with similar scenes can be mapped close to each other. Yin et al. introduce MDFL, a multi-domain feature learning method, to achieve end-to-end LCD. Camara et al. integrate geometric verification and continuous frame time relationships to achieve performance improvements. Those abovementioned methods are robust in specific environment but perform poorly in terms of efficiency. Therefore, Chancán et al. propose a LCD method combining FlyNet and CANN neural network models. This method combines the compact pattern recognition capability of the FlyNet model with the powerful time filtering capability of CANN and greatly improves the efficiency. Khaliq et al. propose a lightweight visual LCD method that can achieve high performance at a lower computational cost, and the efficiency is increased by 12 times compared with state-of-the-art methods. Although the above methods can solve some problems of LCD, they do not perform well in the face of long-term scene changes.

To cope with the challenge of long-term scene changes, researchers have proposed four different types of LCD methods. The first type is based on probability statistics and usually relies on a grid map model. It requires repeated observations of the environment to continuously update and maintain the map model and is only suitable for a small range of application scenes. The second type is based on “sampling” or “memory,” which divides the landmarks into temporary memory and permanent memory. These methods need to visit the same scene as much as possible to cover the changes of scene conditions, and it is easy to cause information explosion in a large range of scenes. The third type is to build a stable semantic information model and use semantic information for matching. However, the current semantic segmentation precision is poor, and the effect of direct application in LCD is not very satisfactory. The fourth type is a learning-based method that uses long-term interaction and learning between mobile robots and the environment. It can improve the adaptability of loop-clousre detection to environment, which is the focus of current research. However, these learning methods based on CNNs suffer catastrophic forgetting problem as the number of scenes increase. The existing visual LCD methods still lack research in this aspect, which is the motivation of the research in this article.

Cross-scene training with continual learning

In this section, we introduce the principle and implementation process of the cross-scene LCD method in detail. In a typical LCD, given a query frame, \( I^q \), and a set of \( M \) database frames, \( I^d = \{ I^d_m, m = 1, 2, \ldots, M \} \), which have been built in the map, the goal is to find the most similar keyframes, \( I^d_k \) in \( I^d \). In general, such a process consists of two key steps: image descriptor generation and loop-closure candidate selection. Among them, the generation of the descriptor needs to be obtained through a large amount of image data training. In our method, continual learning images of different scenes is the key to cross-scene training. Therefore, in the following content, we also pay attention to how to derive the formula for continual learning. As shown online phase in Figure 2, all images should be described in the form of image descriptors (including \( I^d \) and \( I^q \)), and the loop-closure candidate can be searched from \( I^d \). Next, we introduce the entire pipeline and focus on the three core parts: cross-scene training, image descriptor generation, and loop-closure candidate selection.

Overview

Figure 2 clearly shows the pipeline of the proposed cross-scene LCD method. The inspiration of this article comes from the fact that the capability to overcome catastrophic
Forgetting problem is the key to human-like cross-scene learning in LCD, and the latest continual learning work provides us with an idea. Unlike most methods that only use pretrained models, the pipeline designed in this article includes online and offline phases to make our method have stronger learning capability and adaptability. The model training in the offline learning phase is the basis and the main innovation of this article. To this end, this article proposes a novel cross-scene model training method (given in Algorithm 1) and a new network structure (as shown in Figure 4). By using the trained model, the image descriptors can be generated end-to-end.

In the pipeline of Figure 2, some of the links can be considered as part of the subsection “Loop-closure candidate selection.” In the offline phase, we extract the convolution features of all images in the database and implement feature vectorization and dimensionality reduction. Then the inverted index can be built to link the feature vectors to the database images. In the online phase, the convolution features of the query image are firstly extracted, then vectorized, and dimensioned. Lastly, the Top-N candidate images from the database are retrieved to utilize the inverted index and provided to the loop-closure verification link to determine the final loop-closure location.

Cross-scene learning

In fact, mobile robots are always working in a variety of different scenes or environments with changing conditions. For example, an SLAM system that has learned about the office environment in a bright day may need to continue to learn the map of the outdoor environment in a dark evening. However, at present, SLAM system based on deep learning does not have the ability to accumulate experience incrementally. This is mainly because they can only adapt to the autonomous positioning and map building in the environment after training in a specific working scene. When the robot enters into a new working environment, it cannot directly add learning content to the previously

---

**Figure 2.** Overview of our proposed pipeline. It is mainly composed of an online phase and an offline phase. The entire process is described in subsection “Overview” and the details of the two phases in the pipeline are shown in subsections “Image descriptor generation” and “Loop-closure candidate selection.”
trained experience model but need to retrain the model. As the volume of data increases, the existing methods are learning new knowledge while increasing the experience, it leads to the loss of accumulated experience or memory decline, which greatly limits the incremental map building ability of mobile robot for different scenes, thus we propose a new cross-scene training method to overcome those disadvantages.

Algorithm 1 describes the proposed method, which can improve the performance of LCD in terms of continual learning for multiple scenes. Firstly, the K-Means clustering algorithm is used to calculate the clustering center, and the obtained clustering center is used to initialize the NetV-LAD layer. Since the high acquisition cost of label data, and easy acquisition of GPS data, we use GPS as ground-truth to realize weakly supervised learning, and the triplet loss function is also used to calculate the loss for back-propagation (BP). In the process of BP, the schematic of our cross-scene method is shown in Figure 3. Traditional methods, such as stochastic gradient descent (SGD), most likely search outside rather than the overlapping area. Thus, we define the orthogonal projector $P_{l+1}$ in the input space of layer $l + 1$ in continual learning, it allows the standard BP to be projected in the overlapping area. Below we derive in detail the process of how to calculate the $P_{l+1}$ matrix.

Consider a neural network of $L + 1$ layers, indexed by $l = 0, 1, \cdots, L$. $W_l$ represents the connections between the $(l - 1)$th and $l$th layer. $x_l(k)$ denote the output of the $l$th layer of the network, where $k$ is the index of sub-task. According to character of Deriving Projection Matrices, it can be easy to derive Orthogonal Projection Matrix of $x_l(k)$ is

$$P_{l+1}(k) = I - (x_l^T(k)x_l(k) + \alpha I)^{-1}x_l^T(k)x_l(k)$$

in which $\alpha$ is the relatively minor constant.

For the case of $k = 0, 1, P_{l+1}(0)$ and $P_{l+1}(1)$ can be written as

$$P_{l+1}(0) = I - (G_l(0) + \alpha I)^{-1}G_l(0)$$

$$P_{l+1}(1) = I - (G_l(1) + \alpha I)^{-1}G_l(1)$$

in which

$$G_l(0) = x_l^T(0)x_l(0) + \alpha I$$

$$G_l(1) = \begin{bmatrix} x_l(0) \\ x_l(1) \end{bmatrix}^T \begin{bmatrix} x_l(0) \\ x_l(1) \end{bmatrix} + \alpha I$$

It can be easily derived that

$$G_l(1) = G_l(0) + x_l^T(1)x_l(1)$$
The convolutional neural network structure diagram proposed by this article. This network can extract the global image descriptor end-to-end, which provides convenience for the subsequent construction of index structure and keyframe query.

$P_{t+1}(1) = P_{t+1}(0) - G_t^{-1}x_i^T(1)x_i(1)P_{t+1}(0)$

According to equations (6) and (7), the relationship of $k$th scene and $k+1$th scene can be written as

$G_i(k+1) = G_i(k) + x_i^T(k+1)x_i(k+1)$

According to the Woodbury matrix identity

$G_i^{-1}(k+1) = G_i^{-1}(k) - G_i^{-1}(k)x_i(k+1)x_i(k+1)G_i^{-1}(k) / (I + x_i(k+1)G_i^{-1}(k)x_i(k+1))$

Thus we can get the orthogonal projection matrix $P$

$P_{t+1}(k+1) = P_{t+1}(k) + G_i^{-1}(k+1)x_i^T(k+1)x_i(k+1)$

Traditional method such as SGD update the weight matrix by

$W_i(k+1) = W_i(k) + l_e\Delta W_i^{BP}$

where $l_e$ denotes the learning rate.

In our method, the weight matrix $(k > 1)$ in convolutional layer is updated by

$W_i(k+1) = W_i(k) + l_eP_t(k)\Delta W_i^{BP}$

**Image descriptor generation**

As shown in the NetVLAD module in Figure 4, the original NetVLAD method doesn’t take into account the effects of some low-quality features such as ambiguity. These low-quality features don’t contribute much to the recognition and have side effects, and the weight of this low-quality information to the final aggregation should be reduced. The simplest method is to find out the low-quality images in the preprocessing stage and reduce their contribution weight, but this can’t achieve the goal of intelligence. In this article, the method of end-to-end automatic training is implemented so that the network itself can optimize the identification and reduce the weight of this part of the sample. We increase the number of clustering centers to $K + G$ based on NetVLAD, but the added cluster centers did not participate in contribution weights when building the aggregation feature matrix. Therefore, the dimensions of output are still $D_P \times K$, where $\{w_k\}$ and $\{b_k\}$ both have $K + G$ elements, and $c_k$ only $K$ elements. The aggregation descriptor is calculated by equation (13)

$V(j, k) = \sum_{i=1}^{N} \sum_{k=1}^{K+G} e^{w_k x_{i} + b_k} (x_i(j) - c_k(j))$
improved the network structure based on AlexNet\(^4\). AlexNet is oriented to image classification task, using five convolutional layers (among which three convolutional layers are connected to the max-pooling layer), three fully connected layers, containing a total of 630 million connections, 60 million parameters, and 650,000 neurons. Due to the limited computational resource of mobile robots, the algorithm should ensure lightweight and high real-time performance. However, the fully connected layer has a huge number of parameters and considerable computational complexity, so we remove all the fully connected layers from the network structure. The max-pooling layer can reduce the size of the model and improve the computing speed, but it loses a lot of information that is irreversible. We remove the final max-pooling layer from the network structure, which can enhance the robustness of the network model. In this way, the output of the convolutional layer can be conveniently used as the input of the NetVLAD module.

### Loop-closure candidate selection

The ultimate goal of the proposed online phase is to search the most similar N keyframes for current observed image from the map, which are the candidate loop-closure keyframes. The process can be described as: given \(x\) which is \(D\)-dimensional vector (current observed image) and set \(\Gamma = y_1, y_2, \ldots, y_k\) (map), \(N\) nearest neighbors with the shortest distance from \(x\) should be found. Taking the Euclid distance as an example, it can be expressed as

\[
L = N - \arg\min_{i=0:N} \| x - y_i \| \quad (14)
\]

The simplest way is to compare the query frame with all the images in the map one by one and then select the \(N\) nearest candidates. The time complexity of constructing distance matrix and finding \(N\) nearest neighbors in the minimum heap algorithm for distance matrix are \(O(D \times N^2)\) and \(O((k - n)\log N)\), respectively. If \(K = 20000000\), \(N = 1000\), and \(D = 1000\), the order of operation time complexity of distance matrix is constructed to reach \(10^{17}\), and the time complexity of finding \(N\) nearest neighbors from distance matrix reaches \(10^8\), which is unacceptable for LCD algorithm with strong real-time requirement.

But this is not the focus of this article, so our experiment directly adopts the product quantization inverted method\(^4\) to ensure experimental efficiency of the whole method for validating effectiveness of the proposed cross-scene method with continual learning. Algorithm 2 describes the process of obtaining the loop-closure candidate, and the most similar N keyframes can be searched according to the current observed image. In line 3 of Algorithm 2, it is the process of inverted indexing of all images in the map, while line 4 is the process of a real-time query of the observed images in the index structure.

### Experimental setup

We perform a number of comparative experiments to evaluate the performance of the proposed LCD method in the cross-scene environment. The operating system of the experimental environment is Ubuntu 18.04, and the graphics card type is Nvidia RTX 2080Ti. In this section, the data sets related to the experiment are firstly described, then the adopted evaluation protocols and methods are introduced, and finally the comparison methods are listed.

#### Data sets

Oxford RobotCar data set\(^45\) and Pittsburgh data set\(^46\) have been used as standard testing data sets in many papers. In this article, three challenging scenes are selected to evaluate the performance of the proposed LCD method. In our experiments, two scenes are taken from the Oxford RobotCar data set, in which the RobotCar-Day scene is collected from the day images, the RobotCar-Night scene is from the night images, and the third scene is collected from the Pittsburgh. The learning data include three parts: training set, validation set, and test set.

1. **Training set**: Training set is the scene keyframes that have been collected during the running of the autonomous mobile robot. It is used for model fitting, which is crucial for the generation of the final model.

---

### Algorithm 2. Loop-closure candidate selection.

**Input**: database images \(I_d\), query image \(I_q\), number of top candidates to select \(N\)

**Output**: best matched \(I_d^*\)

1. \(F_{model,d} \leftarrow \text{impro}_\text{model}(I_d)\) # Using the improved CNN to get the deep features
2. \(F_{model,q} \leftarrow \text{impro}_\text{model}(I_q)\)
3. \(F_d \leftarrow \text{impro}_\text{netvlad}(F_{model,d})\) # Converting the model feature to improved netvlad vector by Eq. (13)
4. \(F_q \leftarrow \text{impro}_\text{netvlad}(F_{model,q})\)
5. \(F_d, \text{index} \leftarrow \text{Inverted}_\text{index}(F_d)\) # Indexing the database images by Product Quantization
6. \(I_d^* \leftarrow \text{Online}_\text{search}(F_q, F_d, \text{index}, N)\) # Selecting the top \(N\) candidates with query image
7. \(\text{return } I_d^*\)
Table 1. Detailed composition of the training set, validation set, and test set in the RobotCar-Day scene.

| Acquisition time       | Camera   | Frame number |
|------------------------|----------|--------------|
| training set           | query    | 2014/11/28-12:07:13 Stereo-Centre 7070 |
| database               |          | 2014/12/09-13:21:02 Stereo-Centre 11376 |
| validation set         | query    | 2014/11/28-12:07:13 Mono-Left 3450 |
| database               |          | 2014/12/09-13:21:02 Mono-Left 6899 |
| test set               | query    | 2014/11/28-12:07:13 Mono-Left 3449 |
| database               |          | 2014/12/09-13:21:02 Mono-Left 6898 |

Table 2. Detailed composition of the training set, validation set, and test set in the RobotCar-Night scene.

| Acquisition time       | Camera   | Frame number |
|------------------------|----------|--------------|
| training set           | query    | 2014/12/16-18:44:24 Stereo-Left 6517 |
| database               |          | 2014/12/10-18:10:50 Stereo-Centre 12772 |
| validation set         | query    | 2014/12/16-18:44:24 Mono-Rear 3450 |
| database               |          | 2014/12/10-18:10:50 Mono-Left 6899 |
| test set               | query    | 2014/12/16-18:44:24 Mono-Rear 3449 |
| database               |          | 2014/12/10-18:10:50 Mono-Rear 6898 |

2. Validation set: Validation set is a sample set left alone during model training, which is used to adjust the hyperparameters and to preliminarily evaluate the capability of the model.

3. Test set: Test set is used to evaluate the generalization capability of the final model. Based on the test set, we can evaluate the performance of our LCD algorithm.

In detail, the training data set, validation data set, and test data set of RobotCar-Day scene data set and RobotCar-Night scene data set, and Pittsburgh scene data set are following as Table 1, Table 2, and the reference, respectively. In the following, the three scene data sets of RobotCar-Day scene data set, RobotCar-Night scene data set, and Pittsburgh scene data set are introduced.

1. RobotCar-Day: The RobotCar-Day scene selects images collected by different cameras at different times. In the sequence collected by the stereo central camera, we select one frame every five and finally got 7070 images as the query sequence. In the sequence collected by the stereo left camera, we take every 3 frames to get 11,376 frames as the database sequence. Both the test set and the validation set are collected at different times by monocular left camera. For more detailed images, acquisition information is shown in Table 1. Taking into account the errors of the GPS data, we use the GPS data corrected by the inertial measurement unit (IMU) as the ground-truth of localization and set the localization accuracy to 25 m as the success of the query.

2. RobotCar-Night: Similar to RobotCar-Day, RobotCar-Night uses different cameras to capture images at different times. All images in the training set are collected by a stereo camera. In the sequence collected by the stereo left camera, we select one frame every five to finally get 6517 images as the query set. In the sequence collected by the stereo central camera, we select one frame every three to finally get 12,772 frames as the database. Both the test set and the validation set are collected at different times with the rear camera, and the specific acquisition information is shown in Table 2. In RobotCar-Night, GPS data corrected by the IMU is also used as the ground-truth. We set the localization accuracy within 25 m as a loop-closure candidate.

For the different experiment aims, it should be noted that the test data sets of the following specific experiments are changed. For subsection “Evaluation of overall performance,” the test set of every learning scene is test data sets combined with all of the scenes. For subsection “Evaluation of cross-scene ability,” the every learning scene of test data set is the initial scene.

Evaluation index

Our experiment adopted three typical indexes to evaluate the performance of the algorithm, which are Precision-recall curve, Recall@N, and Average searching time per query. Precision-recall curve is used to evaluate the overall performance of LCD. Recall@N curve is used to show the tendency to continually learn multiple scenes. Average searching time is used to test the real time of the algorithm

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (15)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (16)
\]

1. Precision-recall curve: Precision-recall curve is a commonly used criterion in the LCD community. The classification result of loop-closure algorithm is shown in Table 3, and the calculation methods of precision and recall refer to equations (15) and (16). Different precision and recall can be obtained by setting different thresholds in the algorithm, and the precision-recall curve can clearly reflect the advantages and disadvantages of comparative algorithms.

2. Recall@N curve: Recall@N curve is a standard index to evaluate the LCD algorithm. After the
algorithm continually learns each scene in our experiments, we test the recognition performance on the initial scene in the experiments. To show the variation trend of recognition accuracy more intuitively, we obtain the recall by setting the number of different loop-closure candidates N to plot the Recall@N curve.

3. **Average searching time**: In the mapping process of SLAM, the number of keyframes on the map gradually increase, and the loop detection algorithm must ensure real-time efficiency. We use the average searching time spent in a LCD in map to measure the efficiency.

### Comparative study

In the follow experiments, we compare our approach with different state-of-the-art LCD methods including NetVLAD, Max-Pool, Off-the-Shelf, and GIST.

1. **NetVLAD**\(^\text{17}\)**: NetVLAD is the latest representative in the CNN-based approach, which has been proven successful on multiple challenging data sets. It has shown state-of-the-art recognition accuracy and serves as the baseline. Our goal is to make NetVLAD with continual learning capability and add an index structure to speed up the image matching step. We use it as part of the proposed pipeline and improve the way to update the weights. The effectiveness of this module is verified in subsection “Evaluation of key-modules effectiveness.” In our experiment, the feature dimension generated by NetVLAD encoding is uniformly set to 256.

2. **Max-Pooling**\(^\text{38}\)**: Max-Pool is another well-known LCD method based on CNN, and its ideas are reflected in many papers. Based on this method, we design a variety of max-pool methods compared with the proposed method.

3. **Off-the-Shelf**\(^\text{47}\)**: Many recent papers have shown that the CNN features based on the existing pretrained model can perform well in the LCD task. The parameters in the pretrained model have been trained with a large amount of image data and the parameters have been adjusted. In the experiment, the two most popular pretrained models ImageNet\(^\text{44}\) and Places365\(^\text{48}\) are adopted. As part of the pipeline proposed in this article, it is evaluated in subsection “Evaluation of key-modules effectiveness” to verify the effect of different pretraining models.

4. **GIST**\(^\text{21}\)**: GIST is a famous global image descriptor, the method based GIST is as a typical representative of traditional and non-learning method, which is compared with the proposed method in our experiment.

5. **DBoW2**\(^\text{49}\)**: The bag-of-words (BoW) based on local features such as SIFT and ORB has a very wide application in the current SLAM system. The DBoW2 method is a representative method of BoW and has achieved good results in a variety of popular SLAM systems. In this article, our method compares the overall performance with DBoW2.

To distinguish, we denote the abbreviation of our method as follows. The Cross-Scene Descriptor (CSDesc) refers to the basic version of our proposed pipeline. That is, using the raw network without pretrained model Off-the-Shelf and NetVLAD module directly. When incorporating a pretrained model into the network, CSDesc is suffixed to become CSDesc-Pi (based on ImageNet) or CSDesc-Pp (based on Places365). When including the NetVLAD module, we use the suffix N. These results in the six variants of CSDesc being compared: CSDesc, CSDesc-Pi, CSDesc-Pp, CSDesc-N, CSDesc-N-Pi, and CSDesc-N-Pp. Note that to exclude the NetVLAD module, we use max-pooling operation and \(L_2\)-norm to replace NetVLAD.

### Experimental results

In this section, all the experimental results are showed. To ensure the objectivity and completeness of the experimental data, we construct the data sets in accordance with subsection “Data sets” and did not eliminate the useless data in the data sets. Therefore, some figures show that the area below the precision-recall curve is less than 0.5. We firstly demonstrate the effectiveness of two key modules NetVLAD and Off-the-Shelf. Next, we evaluate the convergence of CSDesc method, the overall performance, the cross-scene performance, and the efficiency of our method.

#### Evaluation of key-modules effectiveness

In this section, we evaluate the role of two key modules NetVLAD and Off-the-Shelf. NetVLAD module or the Off-the-Shelf module in traditional methods can significantly improve performance. We change the traditional weight update way of CNN during the learning process, so it is necessary to evaluate these two modules. This comparison involves two scene data sets, RobotCar-Day and RobotCar-Night, and the results in terms of precision and recall are shown in Figure 5 and Figure 6.

Firstly, we evaluate the effectiveness of adding NetVLAD module. We use the RobotCar-Day scene and RobotCar-Night scene in the experiments and get the same results. Due to space limitations, we only show the
RobotCar-Day results. As shown in Figure 5, the importance of the NetVLAD module is obvious. Examining CSDesc-Pi versus CSDesc-N-Pi and CSDesc-Pp versus CSDesc-N-Pp, we observe a significant improvement in performance when NetVLAD (N) is employed.

Then we evaluate the effectiveness of adding the Off-the-Shelf module. As shown in Figure 6, in some cases, without the Off-the-Shelf module, CSDesc-N is able to outperform CSDesc-N-Pi (cyan vs. magenta). That’s due to perceptual aliasing becomes more pronounced in a night scene. In terms of different pretrained models, Figure 5 shows that under different thresholds, the recognition accuracy is mutually competitive (CSDesc-Pi vs. CSDesc-Pp or CSDesc-N-Pi vs. CSDesc-N-Pp).

In summary, the result demonstrates the usefulness of NetVLAD and Off-the-Shelf independently, and they are both used in our final CSDesc LCD method.

Evaluation of convergence of CSDesc

In LCD, robots repeatedly go back to the same place that may show significant visual appearance differences. The changes (e.g. daytime vs night) are often repetitive during robot navigation. Continual learning can be problematic due to osculation and may never converges. Given the repeated changes, the convergence of continual learning is quite considerable. For example, robots work in the same place during the day and at night. As shown in Figure 7, we compare with a traditional non-learning method, typical CNN methods, and NetVLAD methods based on different pretrained models. Surprisingly, the performance of the Max-Pool method and NetVLAD which adopt the ImageNet pre-trained model is even worse than that of the non-learning method GIST. It indicates that the existing representative learning method is insufficient in the face of such cross-scene testing. CSDesc with cross-scene capability has achieved remarkable results in the face of this challenge. Although the
NetVLAD with pre-trained model Places365 is slightly better than our CSDesc-N-Pp method when we have higher requirements for recall, it still fails to match our CSDesc-N-Pi method. We also conduct a qualitative analysis on the results of the experiment. Figure 8 shows some example matches using the proposed approach and other comparisons.

**Evaluation of overall performance**

In the following experiment, we analyze the overall performance of our method in continually learning multiple scenes. We design two cases:

1. from the night with bright lights to Pittsburgh with the diverse environment, namely RobotCar-Night → Pittsburgh;
2. first experienced the two scenes in (1), and then to the sunny noon, that is, RobotCar-Night → Pittsburgh → RobotCar-Day.

As shown in Figure 9, it is very gratifying that our CSDesc-N-Pi and CSDesc-N-Pp methods significantly surpass the comparative methods. Meanwhile, the NetVLAD method almost fails, while the Max-Pool method and GIST had certain effects. We attribute it to its learning layer with clustering capability, which lost more memory in the learning process because of its lack of cross-scene capability.

Then, we increase the number of scenes to make experiments more challenging and set the continual learning of autonomous mobile robots in three scenes of RobotCar-Night → Pittsburgh → RobotCar-day. As Figure 10 shows, the performance of CSDesc-N-Pi and CSDesc-N-Pp methods proposed in this article is still significantly ahead of those comparative methods. GIST and DBoW2, the traditional methods, still achieve certain results and defeated all the methods without cross-scene capability. If the SLAM system based on the learning method doesn’t have continual learning capability, its performance may not be as good as the traditional non-learning method when faced with cross-scene environments. Therefore, it is very important for LCD with continual learning capability. We conduct a qualitative analysis on experiment results. Figure 11 shows some matching example using the proposed approach and other methods of comparison.

**Evaluation of cross-scene ability**

To demonstrate the continual learning capability of the method proposed in this article more clearly, we use the recall curve with respect to a varying N or the so-call Recall@N as the performance metric. The experiments based on the three cases in subsection “Evaluation of overall performance” and the test sets are RobotCar-Day, RobotCar-Night, and RobotCar-Night, respectively.
Take the Case (2) in subsection “Evaluation of overall performance” as an example to illustrate the evaluation steps, which are mainly divided into three steps:

1. After learning the RobotCar-Night scene, we got model A, and used model A to evaluate the RobotCar-Night test set;
2. After learning the Pittsburgh scene based on model A, model B was obtained, and model B was used to conduct experiments on the RobotCar-Night test set to evaluate the memory capability of the model;
3. After learning the RobotCar-Day scene based on the model B, the model C is obtained, and the RobotCar-Night test set is evaluated using the model C. It can be seen that our test method is to train each scene in turn and use the model obtained by each training to evaluate the initial scene. In this way, we can see the trend of recognition accuracy.

As can be seen from Figure 12, CSDesc still has a strong recognition capability for the RobotCar-Day scene after successively learning two scenes. The NetVLAD method has better performance in single scene learning, but after continual learning two scenes, performance decreases obviously. While the performance of the Max-Pool method also decreases slightly, but the decrease is smaller than that of NetVLAD. The traditional non-learning methods have certain competitiveness, which can be attributed to the relative simplicity of the two scenes.

As can be seen from Figure 13, the CSDesc method has prominent advantages. After continually learning the RobotCar-Night and Pittsburgh scenes, it has a strong recognition capability for the RobotCar-Night scene. Although the NetVLAD method has good performance when learning a single scene, the performance significantly decreases after undergoing cross-scene learning, and at this time the Max-Pool method has surpassed NetVLAD. The recall performance of the traditional non-learning method is still the worst, which can be attributed to the fact that the Pittsburgh has too many scenes, resulting in insufficient GIST representation.

Figure 14 is the evaluation result of multiple methods in the RobotCar-Night test set when the number of recalls $N=1$ is adopted. During the continual learning of three scenes, the performance of the CSDesc shows a slight increase or maintenance, while the NetVLAD shows the most obvious decline, and that of Max-Pool shows a slow decline.

![Figure 10](image-url) **Figure 10.** Evaluation of the overall performance on continual learning scenes RobotCar-Night → Pittsburgh → RobotCar-Day. The results in terms of the precision-recall curve in the figure show that the proposed methods of CSDesc-N-Pi and CSDesc-N-Pp significantly better than other competitors.

![Figure 11](image-url) **Figure 11.** Matched place examples from Figure 10 with under different environmental conditions. (a) is a query image from a RobotCar-Night scene, and (b), (c), (d), and (e) are images matched by different methods. Query image: It can be seen intuitively that only our method (CSDesc-N-Pi) can get the best matching result. Very surprised that the GIST method actually matched the images in the Pittsburgh data set (In subsection “Data sets”). (a) Query image; (b) Matched by CSDesc-N-Pi; (c) Matched by GIST; (d) Matched by DBoW2; (e) Matched by NetVLAD-Pi; (f) Matched by MaxPool-Pi.
Evaluation of the efficiency

In this subsection, we evaluate our method in terms of its matching efficiency. Note that the reported times in this article were those on a workstation with 10-core CPU at 2.2 GHz and 64 GB of RAM. We are most interested in the time it takes to match the map keyframes for the current observed view, which is the most critical step in the algorithm, and all the steps are constant time complexity. We use the data sets on three scenes from Case (3) in subsection “Evaluation of overall performance,” and the map contains a total of 23,796 keyframes. We report the details of the average matching time for each query in Table 4. It can be seen in Table 4 that for each query, our algorithm takes 18.28 ms at 20\% of the entire map size, which is 4.7 k, and 43.52 ms at 23.8 k. On average, our algorithm is therefor able to handle approximately 0.44 million images per second or a map of 1 million images in 2.27 s. It is sufficient for real-time LCD, since a large number of invalid frames have to be removed. If the distance traveled by a robot between two consecutive keyframes is one meter, the proposed method is able to handle a topological appearance map that covers a distance of 1000 kilometers in real time.

Conclusion and future work

In fact, only if the SLAM system has the ability of continual learning can the robot realize incremental map construction in a real scene. Those existing LCD methods have memory forgetting defects, so the existing SLAM cannot achieve incremental mapping, but to learn all the scene data at one time. However, it is almost impossible for practical application, which also limits the intelligent navigation level of robot system.

Table 4. Average searching time per query at different size maps.

| Map size | 4.7k (20%) | 9.5k (20%) | 14.2k (20%) | 19k (80%) | 23.8k (100%) |
|----------|------------|------------|-------------|-----------|-------------|
| Average searching time (ms) | 18.28 | 23.92 | 29.48 | 37 | 43.52 |
In this article, we introduce a novel cross-scene LCD method with continual learning for visual SLAM, which can enhance the capability of continual mapping by restraining the memory decay of robot SLAM system. The greatest contribution of this article is to introduce the continual learning mechanism into the process of LCD. We also achieved automatic optimization and reduced the weight of low-quality features in the scene, aiming at the side-effect problem. We propose a lightweight network structure and add the inverted product quantization index in searching, which can conduct real-time online LCD.

To evaluate the cross-scene performance of our method, lots of experiments were conducted on three kinds of scene data sets RobotCar-Day, RobotCar-Night, and Pittsburgh. Evaluation results have demonstrated that our method completely outperforms NetVLAD, Max-Pool, and GIST by a large margin across the three scene data sets and achieves state-of-the-art LCD accuracy.

In terms of matching efficiency, the average matching time of our method is 43.52 ms per query on the data set and its growth rate is extremely low as the database size increases. In summary, the proposed method is able to robustly perform LCD under practical and challenging conditions with a high efficiency that is highly scalable to large environments.

In the future, we plan to expand the current work with more complex neural network structures such as VGG, ResNet, and DenseNet et al. We intend to develop a deep learning framework to further improve the recognition performance of our current system by utilizing semantic visual information and cross-scene description. SVSF-SLAM is robust face parameters uncertainties and error modeling, thus SVSF-SLAM is very suitable for combining with our method to improve the robustness of the SLAM system.1,2 We will also study more effective human–robot interaction methods based on the similarity between semantic systems and human navigation. In summary, we hope our research can contribute to the further development of intelligent navigation, semantic cognition, and robust localization for mobile robots.

Acknowledgment
The authors would like to thank the reviewers for their constructive comments and suggestions to improve the quality of this article.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the Key Area Research Projects of Universities of Guangdong Province under Grant 2019KZDZX1026 and Grant 2018KZDMX074, in part by the Natural Science Foundation of Guangdong under Grant 2020A1515110255, in part by the National Natural Science Foundation of China under Grant 61603103, in part by the Innovation Team Project of Universities of Guangdong Province under Grant 2020KCXTD015.

ORCID iD
Shilang Chen https://orcid.org/0000-0002-0891-5956

References
1. Mosbah A, Demim F, Mansoul A, et al. Simultaneous localization and mapping navigation of unmanned ground vehicle based on second-order smooth variable structure filter with improved technique to combat fading for advanced wireless communications. Proc Inst Mech Eng Part I-J Syst Control Eng 2021; 235(7): 1258–1271. DOI: 10.1177/095965182065728.
2. Demim F, Benmansour S, Abdelkrim N, et al. Simultaneous localisation and mapping for autonomous underwater vehicle using a combined smooth variable structure filter and extended kalman filter. J Exp Theor Artif Intell 2021; 1(1): 1–30. DOI: 10.1080/0952813X.2021.1908430.
3. Cadena C, Carlone L, Carrillo H, et al. Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. IEEE Trans Robot 2016; 32(6): 1309–1332. DOI: 10.1109/TRO.2016.2624754.
4. Hou Y, Zhang H and Zhou S. Tree-based indexing for real-time ConvNet landmark-based visual place recognition. Int J Adv Robot Syst 2017; 14(1). DOI: 10.1177/1729881416686951.
5. Lesort T, Lomonaco V, Stoian A, et al. Continual learning for robotics: definition, framework, learning strategies, opportunities and challenges. Inf Fusion 2020; 58: 52–68. DOI: 10.1016/j.inffus.2019.12.004.
6. Dai Z, Huang X, Chen W, et al. A comparison of CNN-based and hand-crafted keypoint descriptors. In IEEE International Conference on Robotics and Automation, Montreal, QC, Canada, 20-24 May 2019, pp. 2399–2404. IEEE. DOI: 10.1109/ICRA.2019.8793701.
7. Leng C, Zhang H, Li B, et al. Local feature descriptor for image matching: a survey. IEEE Access 2019; 7: 6424–6434. DOI: 10.1109/ACCESS.2018.2888856.
8. Schlegel D and Grisetti G. HBST: a hamming distance embedding binary search tree for feature-based visual place recognition. IEEE Robot Autom Lett 2018; 3(4): 3741–3748. DOI: 10.1109/LRA.2018.2856542.
9. Mur-Artal R and Tardos JD. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans Robot 2017; 33(5): 1255–1262. DOI: 10.1109/TRO.2017.2705103.
10. Tsintotas KA, Bampis L and Gasteratos A. Probabilistic appearance-based place recognition through bag of tracked words. IEEE Robot Autom Lett 2019; 4(2): 1737–1744. DOI: 10.1109/LRA.2019.2897151.
11. Uy MA and Lee GH. PointNetVLAD: deep point cloud based retrieval for large-scale place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–23 June 2018, pp. 4470–4479. IEEE. DOI: 10.1109/CVPR.2018.00470.

12. Merrill N and Huang G. Lightweight unsupervised deep loop closure. In *Robotics: Science and Systems*. Pittsburgh, Pennsylvania, USA, 26–30 June 2018, pp. 1–10. DOI: 10.15607/RSS.2018.XIV.032.

13. Hou Y, Zhang H and Zhou S. BoCNF: efficient image matching with bag of ConvNet features for scalable and robust visual place recognition. *Auton Robot* 2018; 42(6): 1169–1185. DOI: 10.1007/s10514-017-9684-3.

14. Garg S, Sünderrauf N and Milford M. LoST? Appearance-invariant place recognition for opposite viewpoints using visual semantics. In *Robotics: Science and Systems*. Pittsburgh, Pennsylvania, USA, 2018, pp. 1–10. DOI: 10.15607/RSS.2018.XIV.022.

15. Lopez-Antequera M, Gomez-Ojeda R, Petkov N, et al. Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognit Lett* 2017; 92: 89–95. DOI: 10.1016/j.patrec.2017.04.017.

16. Li Z and Hoiem D. Learning without forgetting. *IEEE Trans Pattern Anal Mach Intell* 2018; 40(12): 2935–2947. DOI: 10.1109/TPAMI.2017.2773081.

17. Arandjelovic R, Gronat P, Torii A, et al. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Trans Pattern Anal Mach Intell* 2018; 40(6): 1437–1451. DOI: 10.1109/TPAMI.2017.2711011.

18. Cummins MJ and Newman PM. FAB-MAP: probabilistic localization and mapping in the space of appearance. *Int J Robot Res* 2008; 27(6): 647–665. DOI: 10.1177/0278364908090961.

19. Lowry S, Sunderhauf N, Newman P, et al. Visual place recognition: a survey. *IEEE Trans Robot* 2016; 32(1): 1–19. DOI: 10.1109/TRO.2015.2496823.

20. Zhang X, Wang L and Su Y. Visual place recognition: a survey from deep learning perspective. *Pattern Recognit* 2021; 113: 107760. DOI: 10.1016/j.patrec.2020.107760.

21. Torralba A, Murphy KP, Freeman WT, et al. Context-based vision system for place and object recognition. In *IEEE International Conference on Computer Vision*, Nice, France, 13–16 October 2003, pp. 273–280. DOI: 10.1109/ICCV.2003.1238354.

22. Rublee E, Rabaud V, Konolige K, et al. ORB: an efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision*, Barcelona, Spain, 6–13 November 2011, pp. 2564–2571. IEEE. DOI: 10.1109/ICCV.2011.6126544.

23. Mur-Artal R, Montiel JMM and Tardos JD. ORB-SLAM: a versatile and accurate monocular slam system. *IEEE Trans Robot* 2015; 31(5): 1147–1163. DOI: 10.1109/TRO.2015.2463671.

24. Yin P, Xu L, Li X, et al. A multi-domain feature learning method for visual place recognition. In *International Conference on Robotics and Automation*, Montreal, QC, Canada, 20–24 May 2019, pp. 319–324. IEEE. DOI: 10.1109/ICRA.2019.8793752.

25. Camara LG, Gábert C and Pfreundl L. Highly robust visual place recognition through spatial matching of CNN features. In *IEEE International Conference on Robotics and Automation*, Paris, France, 2020, pp. 3748–3755. IEEE. DOI: 10.1109/ICRA40945.2020.9196967.

26. Chacón M, Hernandez-Nunez L, Narendra A, et al. A hybrid compact neural architecture for visual place recognition. *IEEE Robot Autom Lett* 2020; 5(2): 993–1000. DOI: 10.1109/LRA.2020.2967324.

27. Khaliq A, Ehsan S, Chen Z, et al. A holistic visual place recognition approach using lightweight CNNs for significant viewpoint and appearance changes. *IEEE Trans Robot* 2020; 36(2): 561–569. DOI: 10.1109/TRO.2019.2956352.

28. Han F, Beleidy SE, Wang H, et al. Learning of holism-landmark graph embedding for place recognition in long-term autonomy. *IEEE Robot Autom Lett* 2018; 3(4): 3669–3676. DOI: 10.1109/LRA.2018.2856274.

29. Gao P and Zhang H. Long-term place recognition through worst-case graph matching to integrate landmark appearances and spatial relationships. In *IEEE International Conference on Robotics and Automation*, Paris, France, 2020, pp. 1070–1076. IEEE. DOI: 10.1109/ICRA40945.2020.9196906.

30. Mazuran M, Burgard W and Tipaldi GD. Nonlinear factor recovery for long-term SLAM. *Int J Robot Res* 2016; 35(1-3): 50–72. DOI: 10.1177/0278364915581629.

31. Labbé M and Michaud F. Long-term online multi-session graph-based SPLAM with memory management. *Auton Robot* 2018; 46(1): 1133–1150. DOI: 10.1007/s10514-017-9682-5.

32. Labbé M and Michaud F. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *J Field Robot* 2019; 36(2): 416–446. DOI: 10.1002/rob.21831.

33. Stenborg E, Tofti C and Hammarsstrand L. Long-term visual localization using semantically segmented images. In *IEEE International Conference on Robotics and Automation*, Brisbane, QLD, Australia, 21–25 May 2018, pp. 6484–6490. IEEE. DOI: 10.1109/ICRA.2018.8463150.

34. Cascianelli S, Costante G, Belloccio E, et al. Robust visual semi-semantic loop closure detection by a covisibility graph and CNN features. *Robot Auton Syst* 2017; 92: 53–65. DOI: 10.1016/j.robot.2017.03.004.

35. Garg S, Sunderhauf N and Milford M. Semantic–geometric visual place recognition: a new perspective for reconciling opposing views. *Int J Robot Res* 2019; 38(6): 416–446. DOI: 10.1002/rob.21831.

36. Chen Z, Liu L, Sa I, et al. Learning context flexible attention for long-term SLAM. *Auton Robot* 2018; 3(4): 4015–4022. DOI: 10.1007/s10514-017-9682-5.

37. Han F, Yang X, Deng Y, et al. SRAL: shared representative appearance learning for long-term visual place recognition. *IEEE Robot Autom Lett* 2017; 2(2): 1172–1179. DOI: 10.1109/LRA.2017.2662061.
38. Tolias G, Sicre R and Jégou H. Particular object retrieval with integral max-pooling of CNN activations. In *International Conference on Learning Representations*, San Juan, Puerto Rico, 2–4 May 2016, pp. 1–12. URL: https://hal.inria.fr/hal-01842218

39. Zeng G, Chen Y, Cui B, et al. Continual learning of context-dependent processing in neural networks. *Nat Mach Intell* 2019; 1(8): 364–372. DOI: 10.1038/s42256-019-0080-x.

40. Schroff F, Kalenichenko D and Philbin J. Facenet: a unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7-12 June 2015, pp. 815–823. IEEE. DOI: 10.1109/CVPR.2015.7298682.

41. Golub GH and Van Loan CF. *Matrix computations*. 3rd ed. JHU press, 2013.

42. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 27–30 June 2016, pp. 770–778. IEEE. DOI: 10.1109/CVPR.2016.90.

43. Chollet F. Xception: deep learning with depthwise separable convolutions. In *IEEE conference on computer vision and pattern recognition*, Honolulu, HI, USA, 21–26 July 2017, pp. 1251–1258. IEEE. DOI: 10.1109/CVPR.2017.195.

44. Krizhevsky A, Sutskever I and Hinton GE. Imagenet classification with deep convolutional neural networks. *Commun ACM* 2017; 60(6): 84–90. DOI: 10.1145/3065386.

45. Maddern W, Pascoe G, Linegar C, et al. 1 year, 1000 km: the oxford robotcar dataset. *Int J Robot Res* 2017; 36(1): 3–15. DOI: 10.1177/0278364916679498.

46. Torii A, Sivic J, Okutomi M, et al. Visual place recognition with repetitive structures. *IEEE Trans Pattern Anal Mach Intell* 2015; 37(11): 2346–2359. DOI: 10.1109/TPAMI.2015.2409868.

47. Sunderhauf N, Shirazi S, Dayoub F, et al. On the performance of ConvNet features for place recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany, 28 September–2 October 2015, pp. 4297–4304. IEEE. DOI: 10.1109/IROS.2015.7353986.

48. Zhou B, Lapedriza A, Khosla A, et al. Places: a 10 million image database for scene recognition. *IEEE Trans Pattern Anal Mach Intell* 2018; 40(6): 1452–1464. DOI: 10.1109/TPAMI.2017.2723009.

49. Gálvez-López D and Tardós JD. Bags of binary words for fast place recognition in image sequences. *IEEE Trans Robot* 2012; 28(5): 1188–1197. DOI: 10.1109/TRO.2012.2197158.