Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Deep Fusion of Multi-Layers Salient CNN Features and Similarity Network for Robust Visual Place Recognition*

Zhenyu Li and Aiguo Zhou†
School of Mechanical Engineering
Tongji University
Shanghai, 201804, China
{zhenyu.li & zhouaiguo}@tongji.edu.cn

Mingyang Wang and Yong Shen
School of Automotive studies
Tongji University
Shanghai, 201804, China
wangmingyang@sina.cn
shenyong@tongji.edu.cn

***Abstract* - Severe changes in appearance and viewpoint of the same place caused by extreme weather and sharp turning are the major challenges for mobile robot's autonomous navigation. Coping with these problems, and improving the performance of place recognition is more and more in the focus of robotics research. In this paper, we propose a novel visual place recognition approach with a deep fusion of the salient features and similarity network. The process of recognition can be simplified into two stages: feature representation and similarity function learning, which is trainable end-to-end. Instead of extracting features directly from the last convoluted layer of the network, the proposed approach extracts feature of the maxpooling layer of each convolutional module, and then fuses these features together. At this stage, the fully connected layer is not used. In order to measure visual similarity between representations in two images, at the second stage, a novel technology that learns similarity function directly from image representations to measure similarity between two images to rank images according to a similarity score is used. Extensive experiments have been conducted to evaluate the performance by comparison with existing state-of-the-art methods, experimental results on five challenging datasets show more accuracy and robustness of the proposed approach under severe changes in appearance and viewpoint.**

Index Terms - CNN feature; Place recognition; Similarity network; Datasets; Image match

I. INTRODUCTION

For humans, no matter how long the time interval, but once again to have been to the same place will be in the mind of the various scenes of this place, to some extent, a natural task. However, in the mobile robot's autonomous navigation domain, this is a challenging task, and off-the-shelf algorithms do not always be successful in recognizing previous place where robots visited. This mainly because the same scenes that appears in the view of robots at different time are changing greatly. For example, in the same place, great changes will take place in the spring and winter, the trees are clustered in spring, and snow is heavy in winter. As well as in the daytime and evening and in sunny and rainy days, the scenes in the same place can also change dramatically. It is difficult for mobile robots to discriminate such changes. Professionally speaking, the ability of recognizing the previous place robots visited also can be interpreted as an image retrieval problem, that is, the problem of similarity between two scenes represented by two images. In the last years, some advanced methods for place recognition are proposed, which can deal with the problem of place recognition in extreme environment. In the literature [1], SeqSLAM is presented for robot navigation, which calculates the best candidate matching place within every local navigation sequence instead of calculating the single place most likely given a current image. This method is a route-based visual navigation, which is suitable for sunny summer and rainy winter nights, and has the strong environmental adaptability. FAB-MAP [2] is an advanced visual place recognition algorithm, which builds on local features such as SIFT [3] or SURF [4], and matches the appearance of the current scene to a place where robots previously visited by converting the image into bag-of-words representations. The SIFT and SURF feature is just related to points of interest in the local appearance of the object, regardless of the size and rotation of the image, therefore, using this method can robustly recognize places. All the methods mentioned above are based on handcrafted feature. However, with the rapid development of machine vision and deep learning, features obtained from deep learning methods become the new state-of-the-art in some recognition tasks, such as object detection, image classification and scene recognition. Compared with the method of labeling feature manually, CNN-based methods can automatically extract features and learn feature representation. The CNN-based feature itself has enough discrimination, and does not need any complex aggregation technology, which greatly improves the performance of scene representation. A great deal of literatures [5-8] show that they can robustly implement multiple recognition tasks in complex environments, and often even

* This work is partially supported the National Key Research and Development Program Grant 2016YFB0100902.

† This is the correspond author.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

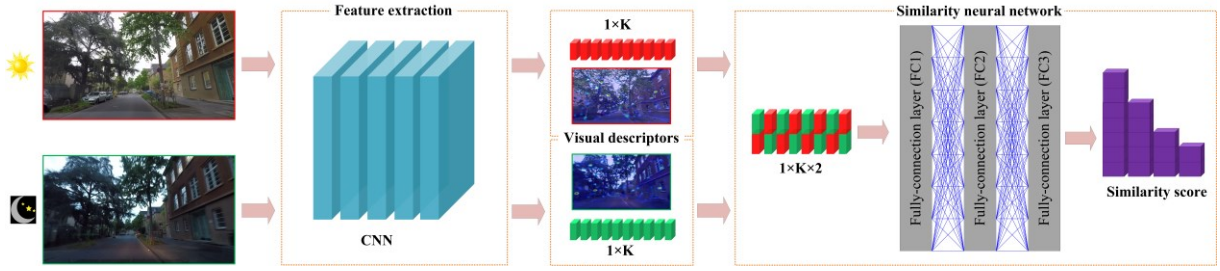


Fig.1 Full pipeline of the recognition system. Given a query image and some reference images, K-dimensional visual representations are extracted from fusion multi-maxpooling layers, and then visual representation vectors are concatenated and fed into the visual similarity neural network, which output a similarity score that can evaluate the similarity between a query image and reference images.

show superior performance against traditional hand-engineered feature.

In this paper, we also utilize prevailing CNN features to represent scenes, as well as present an approach with a deep fusion of multi-layers salient CNN features and similarity network for robust visual place recognition. The proposed approach extracts deep features as front-end operation of place recognition based on network architecture of VGG-16. Unlike the general, CNN-based feature extraction methods, the proposed method first extracts feature from the pooling layer of each convolutional module, and then fuses these features together to form a new feature representation. Any image can be represented as a $1 \times K$ dimensional feature vector after feature extraction. When a query image is given, what we need to do is to find out the most similar image in the reference images for place recognition. At this time, we can regard the query images and reference images as multiple $1 \times K$ dimensional vectors, and then import these vectors into similarity function to compute the similarity score between each reference vector and query vector. In order to find out the most similar reference image, we learn the similarity function and rank the similarity score by training three fully connected neural network layers, which can be seen as the back-end operation of recognition process. The process is shown in Fig.1

II. RELATED WORK

A. Deep learning for place recognition

The successful application of deep learning technology in image representation triggered an upsurge in exploring new methods. The appearance of many algorithms based on learning feature has played a subversive role in image recognition tasks. Compared with traditional hand-engineered methods, CNN-based methods have better robustness in complex environment. Severe changes in the appearance of the environment due to diurnal cycles, seasonal alternations or weather changes are a challenging problem for place recognition. To deal with these problems, [9] proposed a new, untrained, local region-based detector single image matching program, which is used for powerful convolutional neural network (CNN) based descriptors. The camera mounted on the vehicle will vibrate in the course of following the vehicle, which will lead to obvious changes in the viewpoint of the collected images. In the literature [10], a novel omnidirectional convolutional neural network (O-CNN) is proposed, which is used to handle severe camera pose variation. Some harmful factors, such as severe illumination invariant, rotation invariant and robust against moving objects, which are unrelated to the place recognition. However, a novel point-cloud-based place recognition system [11] that adopts a deep learning approach for feature extraction, which is also useful to cope with these problems. In order to cope with the changes of viewpoint, In the literature [9, 12-14], landmark-based CNN are proposed and used for multiple recognition tasks, such as face recognition, object track and medical detection. The input image is firstly divided into multi-scale landmarks with content information, and then the highly representative landmark features are extracted by convolutional neural network (CNN), which has the strong robustness to the appearance change of landmarks. To cope with a large amount of computational consumption, a slight-weight CNN are proposed, such as [15], only utilizes five conventional modules instead of the commonly used fully-connected layers' extracted features that reduces memory usage and speed up execution.

B. Similarity learning for ranking images

Learning a measurement of the similarity function between pairs of images is an important generic problem in machine learning. So far, lots of methods of similarity learning are proposed, such as [16], proposed a novel approach, which learns a non-metric visual similarity function directly from image representations to measure similarity between the query images and reference images, and then ranks the similarity score. In some popular face recognition algorithms, similarity learning technologies are also used. Such as [17], presents a method of training similarity measure of data that can be used to identify or validate applications, in which the number of categories is very large, the training period is unknown, and the number of training samples of a single category is very small. There are some similar methods, and the representative ones are [18] and [19]. The former is able to search for an image from a large scale image database, which is similar to a query image or finding videos that are relevant to a query video by learning the triplet loss function. And the latter presents a new scheme of learning similarity measure for content-based image retrieval, which separates the images in the database into two clusters by learning a boundary function. Similar to above mentioned methods, we also utilize the

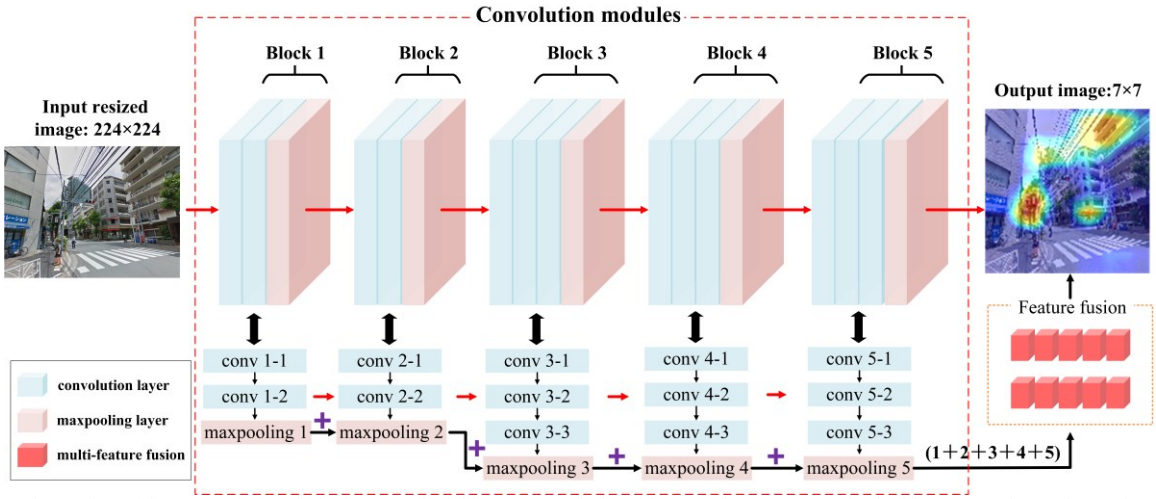


Fig.2 The network consists of five convolutional blocks, a total of thirteen convolutional layers. The parameters in the blue rectangle are denoted as Conv - (block number) - (layer number). All the convolutional layers use the same size convolution kernel 3 × 3. The final feature representation is obtained by fusing these features together, which are extracted from five pooling layers.

technology of learning a similarity measure for ranking similarity score between pairs of images. According to learning a similarity function, some similarity scores can be obtained, and the image with the highest score is the most similar with a query image. This process is similar to image retrieval, and the successful retrieval of an image pair indicates that the robot successfully recognizes the same place.

III. PROPOSED APPROACH

In this section, we present the logical structure of proposed systems, as shown Fig.1, as well as describe the used network architecture and process of network training. In addition, we also describe how the salient features are extracted from pooling layers and how these salient features are fused. Finally, the similarity function is learned by three fully connected layers for ranking the similarity scores.

A. The network architecture

The key task of this paper is to learn a representation of some scenes that can help solve the subsequent tasks of place recognition. The Fig.2 shows the operation mechanism of feature extraction. The network structure we used is based on the first five modules of VGG-16, which abandons the fully connectional layer. In addition, the weight of pre-training is applied to the process of convolution operations. In our work, some image datasets are used as the input of the proposed network, and all images are resized into 224 × 224 × 3 before entering the network. The parameters indicate that the length and width of the frame are 224 pixels and the number of color channels is 3. The whole network includes five modules, in which the first two modules consist of two convolutional layers and one maxpooling layer, respectively. The last three modules consist of three convolutional layers and one maxpooling layer, respectively. In the convolution process, 64 filters, 128 filters, 256 filters, 512 filters and 512 filters are used in the 1 to 5 modules, respectively. All pooling layers utilize maxpooling to extract features, which size of the kernel is 2 × 2, and size of the stride is also 2 × 2. During convolution operations, in order to ensure the size output feature map is the same as the size of the input, the padding technology is used, and the size of padding is set as 'Same'. The outline of the network architecture is shown in Table 1. All activation functions used in convolutional modules are ReLu (Rectified Linear Unit).

TABLE I
The outline of the used network architecture. The resized image is 224 pixels ×224 pixels, and the final output size of the image is 7 pixels × 7 pixels. The output size of each block is the input size of the next one

Block	Convolution operation	Activation shape	Padding	Activation size	Parameters
0	Input	224 × 224 × 3	—	150,528	0
1	Conv 1	224 × 224 × 64	Same	3,211,264	38,720
	Pooling 1	112 × 112 × 64	—	802,816	0
2	Conv 2	112 × 112 × 128	Same	1,605,632	221,440
	Pooling 2	56 × 56 × 128	—	401,408	0
3	Conv 3	56 × 56 × 256	Same	802,816	1,475,328
	Pooling 3	28 × 28 × 256	—	200,704	0
4	Conv 4	28 × 28 × 512	Same	401,408	5,899,766
	Pooling 4	14 × 14 × 512	—	100,352	0
5	Conv 5	14 × 14 × 512	Same	100,352	7,079,424
	Pooling 5	7 × 7 × 512	—	25,088	0

B. Convolution operations

We extract descriptors for each image region by training the VGG-16. The training weights are initially trained on the ImageNet dataset. Although CNN was originally trained for datasets used in image retrieval and automatic feature extraction, it has proved to be highly versatile and more

efficient than other visual features manually produced. Convolutional neural networks transform input images into multiple descriptors through a series of simple convolution operations or layer training, each layer performs end-to-end linear operations. We try to extract the local representation of a specific image area, and activate it directly from the pooling layer. When given image I, its activation (response) in the convolutional layer can be arranged as the size tensor H × W × D, where H and W represent the height and width of each feature map, respectively. And D is the number of feature channels. Activation (response) can be connected to a C-dimensional local descriptor that represents a local image region at all spatial locations of the feature map. As convolution operations go deeper, the semantic level of feature maps becomes higher and higher. For CNN, there is a very basic understanding: Low-level convolution learning texture and other simple information, high-level convolution learning semantic information. Semantic representation can better predict low-pixel images. In our work, we directly extract features from pooling layer. The convolutional layer in the convolutional neural network is to convolute a neighborhood of an image to obtain its neighborhood features. The sub-sampling layer uses pooling technology to integrate the feature points in a small neighborhood to get a new feature. Therefore, the features extracted from the pooling layer integrate the most excellent features of the convolutional modules. In addition, it has good translation and distortion invariance, which to a large extent can ensure the good performance of recognition under the change of appearance and viewpoint. The Fig.3 shows the feature maps that extracts from each pooling layer. It can be seen that the semantic effect of the feature maps becomes more and more obvious with the deepening of the convolution operations.

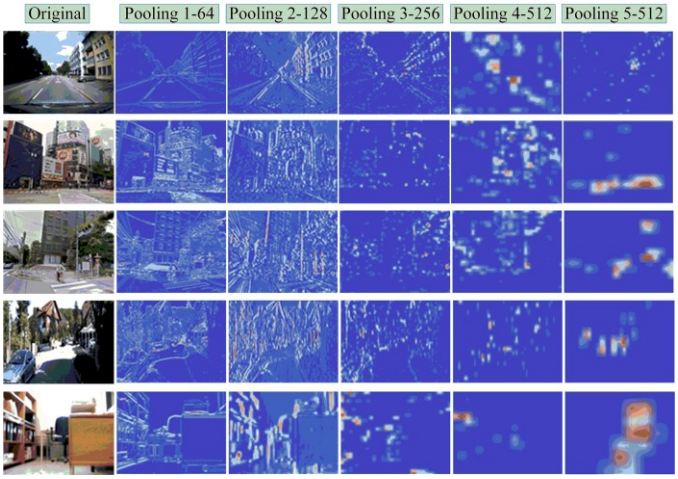


Fig.3 Examples of visualization of feature maps. From the first row to the last row, the original images come from VPRICE dataset [22], Tokyo 24/7 dataset [23], Tokyo Time machine dataset [24], KITTI dataset [25] and KTH-IDOL2 dataset [26], respectively. The first column is the original image. From the second column to the last column, the feature maps extracted from the first pooling layer to the fifth pooling layer are represented, respectively. The parameters in the title are denoted as pooling (block number - sequence number of feature map). For example, pooling (1-64) can be interpreted as the feature map extracted from the pooling layer of the first convolutional module.

Convolution is the basic layer of any CNNs and is usually the primary layer of all the blocks. Five modules consisting of thirteen convolutional layers that forms proposed convolutional architecture. Rectified Linear Unit (Relu) is an activation function used into proposed network, The Relu layer lies behind the last convolutional layer of each convolutional block. We chose Relu as the activation function because it has a faster convergence rate than tanh or sigmoid in gradient descent [18]. The Relu can be implemented by simply designing the threshold of matrix at 0, as exposed in formula (1):

$$f(x) = \max(0, x) \tag{1}$$

Where $x \in \mathbb{R}$ is an input map in each convolutional layer. In order to clearly describe the process of the convolution operation, we first number each pixel of the image, using $x_{i,j}$ to represent the row and column elements of the image. We number each weight of the filter, using w to represent the column weight of the i row and the j column weight of the filter, using b_{d_l} to represent the bias item of the filter. We also number each element of the feature map, using $A(i_l, j_l, d_l)$ to represent the column element of the i row and the j column element of the feature map. Finally, using the follow formula to calculate the convolution:

$$A(i_l, j_l, d_l) = f\Big(\sum_{d=0}^{D_l-1}\sum_{i=0}^{F_1-1}\sum_{j=0}^{F_2-1} w(i, j, d, d_l)\, x_l (iS + i, j_l S + j, d) + b_{d_l}\Big) \tag{2}$$

The confluence layer sampled according to the local statistical information in the neuron space, which reduced the space size of neurons while retaining useful information, further reduced the number of parameters and reduced the possibility of over-fitting. The pooling operation was carried out independently on each depth component. In our work, the maxpooling operator is implemented, which processes the maximum response of each feature channel in a $F_1 \times F_2$ patch, and The internal application of each pooling layer is formulated in formula (3):

$$A'(i_l, j_l, d_l) = \max_{0 \le i \le F_1, 0 \le j \le F_2} \{x_l(iS + i, j_l S + j, d)\}, \forall i_l, j_l, d_l \tag{3}$$

Where S donates the stride of each filter.

Activation across a spatial location of all feature maps can be connected to a C-dimensional local descriptor, which represents a local image region. The size of this region is equal to the receiving region of the filter. Therefore, each feature descriptor can represent a special region in an image. In our work, we interpret an input image a three-dimensional tensor, which represents the response as a two-dimensional feature channel:

$$X = \{X_i\}, i = 1, 2, \cdots, K \tag{4}$$

Where X_i is the two-dimensional tensor that represents the responses of the i^{th} feature channel, and $X_i(S)$ denotes the response at a particular position in an image. So the feature

vector can be constructed by a spatial maxpooling over all positions [20], which can be calculated by the follow formula:

$$\begin{cases} f_{\Omega} = \left[f_{\Omega,1}, f_{\Omega,2}, \cdots, f_{\Omega,K} \right]^T \\ f_{\Omega,i} = \sum_{p \in \Omega} X_i(p) \end{cases} \tag{5}$$

Where $f_{\Omega,i}$ is the i^{th} valid spatial position that represents the salient region in an image. Multiple salient regions are appearing in each feature map, and the sum of all features in all salient regions can be represented as formula (6). In order to form a more robust final descriptor from the convolutional neural network, according to [21], it is defined that d_{pool_i} and d_{pool} represent feature vector of each pooling layer and the sum of feature vectors extracted from five pooling layers. So the final vector is calculated by the formula (7):

$$d_{pool_i} = \sum_{i=0}^{N} f_{\Omega,i} \tag{6}$$

$$d_{pool} = d_{pool_1} + d_{pool_2} + d_{pool_3} + d_{pool_4} + d_{pool_5} \tag{7}$$

C. Training the similarity network

The similarity network consists of three fully connected layers, each one of them followed by a ReLu nonlinear layer [19, 27]. An image is convolved to output a K dimensional vector, and a pair of images output a $K \times 2$ dimensional vector, which is set as the input of the similarity network, as shown in Fig.1. The proposed similarity network consists of three fully connected layers, the first of which is the input layer, and the second of which is hidden layer, and the last of which is output layer. As mentioned earlier, the output size after convolution operations is $1 \times K \times 2$, so the size of the first layer of similarity network is $1 \times K \times 2 \times C_1$, where C_1 is the number of channels in the first layer. The size of the hidden layer of similarity network is $1 \times 2 \times C_1 \times C_2$, where C_2 is the number of channels in the hidden layer. Furthermore, the size of the output layer of similarity network is $1 \times 2 \times 1 \times C_3$, where C_3 is the number of channels in the output layer. Since the two images imported into the convolutional layer may or may not be similar, the output score should cover all values, including positive and negative. In our work, during learning the function, the $l2-norm$ loss function is used to calculate the regression loss, as shown in formula (8):

$$L(I_i, I_j) = \left| s_{i,j} - y_{i,j} \right| = \left| \chi\left(d^i_{pool}, d^j_{pool}, w_{\chi_a} \right) - y_{i,j} \right| \tag{8}$$

Where I_i, I_j denote query image and the reference image, respectively. And $s_{i,j}$ $y_{i,j}$ are the output score of similarity network and the annotated score, respectively. w_{χ_a} is the weight of similarity network, which is updated after each layer of network training.

On the back-end stage, we utilize the similarity network to train the similarity function based on cosine similarity [28]. We assume that d^i_{pool} and d^j_{pool} denote random vectors representation of query image and the reference image, respectively. The similarity label is obtained by computing the cosine similarity:

$$y_{i,j} = \frac{d^i_{pool} \cdot d^j_{pool}}{\left\| d^i_{pool} \right\| \left\| d^j_{pool} \right\|} \tag{9}$$

D. Image matching

The best similar image matching with the query image is the reference image with the highest score. The aim of learning the similarity function is to minimize the value of loss function, which can guarantee that the images taken at the same scene have the highest similarity. The practical purpose of place recognition is to design a correlation algorithm to make the two images taken at the same place similar, not the two images taken at the same place different. The formula (10) can ensure that the more similar between two images taken at the same place:

$$L^{'} = \min\left\{ L\left(I_i, I^{'}_j \right) \right\} \infty \Gamma = \operatorname{argmax}\left\{ \chi\left(d^i_{pool}, d^j_{pool}, w_{\chi_a} \right) \right\} \tag{10}$$

Where $L^{'}$ is the minimum loss in multiple pairs of images, Γ is the highest similarity score, and ' ∞ ' represents the equivalence of two formulas. In order to render our algorithm clearly to the reader, we summarize our algorithm in the following:

Place recognition algorithm

Input: query images $I = \left(I_1, I_2, \cdots, I_M \right)$, reference images $I^{'} = \left(I^{'}_1, I^{'}_2, \cdots, I^{'}_n \right)$.

Output: matched pairs of images.

1. To compute the similarity score according to formula (9).
2. To compute the regression loss, according to formula (8).
3. **for** $i = 1$ to $i = m$, $j = 1$ to $j = n$
4. Initialization: $i = 1$, $L_{i,j} = \left(L_i, L^{'}_j \right)$
5. **if** $L_1 = \min\limits_{1 \le j \le n} \left\{ L\left(I_1, I^{'}_1 \right), L\left(I_1, I^{'}_2 \right) \right\}$, **then**
6. $L_2 = \min\limits_{1 \le j \le n} \left\{ L\left(I_1, I^{'}_1 \right), L\left(I_1, I^{'}_3 \right) \right\}$
7. **until** $L = L_{1,j} \le \min\limits_{1 \le j \le n} \left\{ L\left(I_1, I^{'}_1 \right), L\left(I_1, I^{'}_2 \right), \cdots, L\left(I_1, I^{'}_n \right) \right\}$
8. **return** $L_{1,j} = L\left(I_1, I^{'}_j \right)$, the most similar image to I_1 is $I^{'}_j$.
9. **update** $i = i + 1$, **repeat** step 4 – 9.
10. **end**
11. **end for**
12. **return** pairs of similar image: $\left(I_1, I^{'}_a \right), \left(I_2, I^{'}_b \right), \cdots, \left(I_i, I^{'}_j \right)$.

IV. EXPERIMENT

In this section, the proposed approach is evaluated by comparing with existing state-of-the-art methods on five challenging datasets. In order to ensure the fairness of results, all experiments are performed on the same hardware platform.

We utilize pooling layer of each convolutional module to extract local descriptors, and utilize fusion of five pooling layers to discover salient regions. All images are first resized as 224 pixels × 224 pixels before they are fed into convolutional modules.

A. Datasets

We evaluate the proposed approach on five benchmark place recognition datasets. These datasets were taken with different types of environments. Details are summarized in Table 2. Each dataset consists of two traversals along the same path, the one of them is used for reference and the other one is used for testing.

TABLE II

Datasets used in our experiment that shows the change in appearance and viewpoint. We evaluate the proposed method on the following five datasets. These datasets were taken in different environments. Some datasets are road scenes, some are street scenes, and some are night scenes and indoor office scenes.

Datasets	No. of frames	Environment	Appearance	Viewpoint
VPRICE	800	Road	Severe	None
KITTI	9,080	Road	None	None
Tokyo 24/7	151,968	Cityscape	Severe	None
Tokyo Time Machine	98,208	Cityscape	None	Minor
KTH-IDOL2	184,000	Office	Severe	Severe

VPRICE is a very challenging dataset, which is captured in Bonn by a dashboard camera mounted on a car. The query images and reference images contain several revisits of the same place. The dataset is collected in the morning with slight rain an overcast and in the evening or very late evening on different days. The entire dataset consists of 800 images, all of which are of the same size 1920 pixels × 1080 pixels. The KITTI is the largest dataset for computer vision algorithm evaluation in the world under autonomous driving environment. It consists of 22 stereo sequences, and is saved in png format, and all of which are 1242 pixels × 375 pixels. The first eleven sequences with ground truth locus (00-10) are used for training, and the last eleven sequences without ground truth locus (11-21) are used for evaluation. In our experiment, sequence (00) was selected for reference images and sequence (10) was selected for query images. The Tokyo 24/7 dataset is part of Google Street View, which contains 76 thousand reference images and 315 query images that are taken by using mobile phone cameras. It is a very challenging dataset and changeable greatly in appearance, all of which are the same size 640 pixels × 480 pixels. Query images were taken during the day, sunset, and night, while reference images were taken during the day. The Tokyo Time Machine is part of the Google Time Machine, which contains daily street view images, while video is much larger for a variety of activities, including active days and 45-degree views of street view, all of which are the same size 640 pixels × 480 pixels. The KTH-IDOL2 dataset was taken indoor environment by the laser scanning method. It consists of 24 image sequences. Under different illumination conditions, all image sequences are continuously familiar with each other at the speed of 5-fps. Every image sequence shows serious viewpoints, and all of which are 309 pixels × 240 pixels. In order to ensure that the experiment is carried out in the environment of no change in appearance, but in the environment of viewpoint change, we only choose one of the sequences to carry out the experiment.

B. Evaluate metric

The proposed approach is evaluated against other state-of-the-art algorithms. The metric of performance evaluation is Precision-Recall curves. In our experiment, we compare proposed approach with both manual feature methods and CNN-based feature methods, such as SeqSLAM [29], VGG-16-Fully, VGG-16 and NetVLAD [30]. The difference between VGG-16-Fully and VGG-16 is that the former consists of five convolutional modules and three convolutional layers, while the latter just consists of five convolutional modules without full connected layer. We also show the results of visual place recognition in five datasets. We assume that TP is true positive, TN is true negative, FP is false positive and FN is a false negative, then the precision and recall can be calculated by the formula (11) and (12):

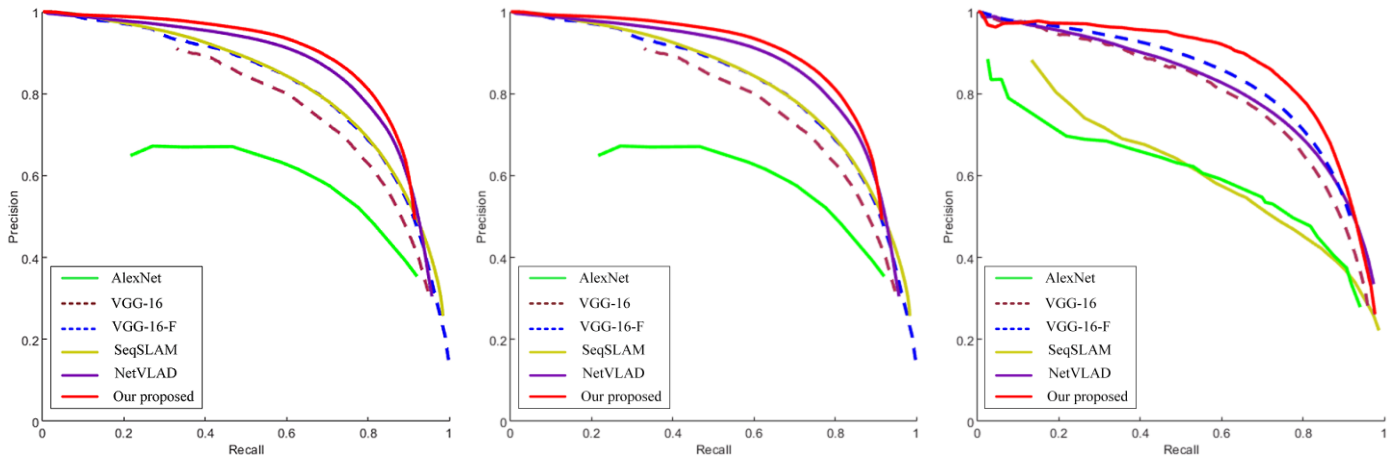
$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

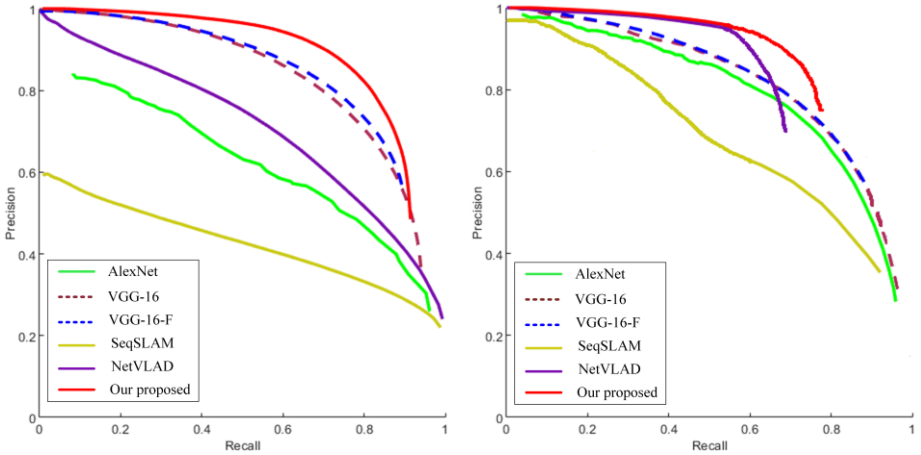
C. Experiment results and analysis

In this section, the experiment is divided into two parts, one of which is the performance comparison between the proposed method and existing state-of-the-art methods. And another one is examples of place recognition and the detection of local salient region. Please note that all the experiments are carried out on the same hardware and software platform.

We evaluate the performance of the proposed approach by comparing it with existing state-of-the-art on five popular place datasets. These methods used in this paper for evaluation are AlexNet, VGG-16, VGG-16-F, NetVLAD [29] and SeqSLAM [1]. The AlexNet only consists of five convolutional layers and two fully connected layers. Features used in the experiment are extracted from the last connected layer. VGG-16 and VGG-16-F have the same convolutional layer. And the difference between them is that the former extracts features from the last convolutional layer without using fully connected layer, the latter extracts features from the last Full-connected layer with using three fully connected layers. Two others all are the off-the-shelf advanced methods



(a) The VPRICE dataset (b) The KITTI dataset (c) The Tokyo 24/7 dataset



(d) The Tokyo Time Machine (e) The KTH-IDOL2 dataset

Fig.4 The results of experiment on five popular place datasets. Five methods are used for evaluating the performance. In our work, we take precision-recall curve as the only evaluation index of performance.

of place recognition, one of which extracts features from CNN layer, another one extracts feature by hand-engineered.

The Fig.4 (a) shows the results of proposed method compared with the five top performing baselines on the VPRICE dataset. It can be seen that the proposed method performs better than others, and the same excellent method as ours is NetVLAD. In addition, the VGG-16 with a fully connected layer outperform the VGG-16 with no fully connected layer, and have the same performance with SeqSLAM. The worst performance is AlexNet. According to performance comparison, for CNN, we can see that the deeper the network, the better the performance. The Fig.4 (b) represents the performance of the proposed method and other baselines on KITTI datasets. It can be seen that the proposed method is superior to others. VGG-16-F, VGG-16 and NetVLAD have the similar performance. However, compared with the above methods, AlexNet and SeqSLAM perform poorly. The Fig.4 (c) reports the results of the proposed method compared with others on the Tokyo 24/7 dataset. We can see that the NetVLAD has the best performance compared with others. However, proposed method performs the similar performance and outperforms other four methods obviously. The Fig.4 (d) shows the performance comparison with state-of-the-art methods on the Tokyo Time Machine dataset. As can be seen that the proposed method is better than others, the performance rankings of these methods are VGG-16-F, VGG-16, NetVLAD, AlexNet and SeqSLAM in turn. The Fig.4 (e) shows the results of our method and others on the KTH-IDOL2 dataset. Compared with the first four datasets, the KTH-IDOL2 dataset is collected indoors. While there are great changes in appearance, there are also serious changes in viewpoint. It can be seen that the proposed method also performs well on this dataset. The NetVLAD performs slightly inferior ours. It also can be seen that the performance of VGG-16-F and VGG-16 tends to be consistent, and perform better than AlexNet and SeqSALM.

V. CONCLUTION

We propose a novel approach with a deep fusion of multi-maxpooling layer's features and similarity network for place recognition, as well as evaluate the performance by comparison with existing state-of-the-art methods on five challenging place datasets. Whether in the case of extreme changes in appearance or in the case of severe changes in viewpoint, the proposed approach performs better in performance. The performance on five public datasets demonstrate that deep learning architecture can learn more effective image representation, that is, CNN feature has better robustness than manual feature. In addition, for CNN, the deeper the network, the better the performance. In the future, we will add time consumption as an evaluation metric on the basis of our work. We have learned that the increasing network depth can improve performance, but this improvement is not proportional to increasing network depth. When network depth reaches a certain level, performance will not be improved, on the contrary, it may also decrease. What we need to do in the future is to find out the most cost-effective network depth by evaluating the time consumption.

REFERENCES

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

[2] M. Cummins, P. Newman, "FAB-MAP: appearance-based place recognition and mapping using a learned visual vocabulary model," International Conference on International Conference on Machine Learning, pp. 3–10, 2010.

[3] T. W. Yan, H. Garcia-Molina, "SIFT: a tool for wide-area information dissemination," Usenix Technical Conference, 1995.

[4] H. Bay, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding, vol. 110, pp. 346-359, 2008.

[5] A. S. Razavian, H. Azizpour, J. Sullivan, et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 806-813, 2014.

[6] T. Zhou, M. Brown, N. Snavely, et al, "Unsupervised Learning of Depth and Ego-Motion from Video," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1851-1858, 2017.

[7] N. Suenderhauf, S. Shirazi, A.Jacobson, et al., "Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free," Proceedings of the 2010 Academy of Marketing Science (AMS) Annual Conference, 2010.

[8] M. Elawady, S. Yildirim, J. Y. Hardeberg, et al., "Editorial Image Retrieval Using Handcrafted and CNN Features," International Conference on Image and Signal Processing, pp. 284-291,2018.

[9] P. Neubert, P. Protzel, "Local region detector + CNN based landmarks for practical place recognition in changing environments," European Conference on Mobile Robots, pp. 1-6, 2015.

[10] T. H. Wang, H. J. Huang, J. T. Lin, et al., "Omnidirectional CNN for Visual Place Recognition and Navigation," International Conference on Robotics and Automation (ICRA), pp. 2341-2348, 2018.

[11] T. Sun, M. Liu, H. Ye, D. Y. Yeung, "Point-cloud-based place recognition using CNN feature extraction," arXiv preprint arXiv:1810.09631.

[12] X. Zhe, X. Cui, J. Zhang, et al., "Real-Time Visual Place Recognition Based on Analyzing Distribution of Multi-scale CNN Landmarks," Journal of Intelligent and Robotic Systems, vol. 94, no. 3, pp. 1-16, 2018.

[13] A. S. Jackson, M. Valstar, G. Tzimiropoulos, "A CNN Cascade for Landmark Guided Semantic Part Segmentation," European Conference on Computer Vision (ECCV), pp. 143-155, 2016.

[14] Z. Gan, L. Ma, C. Wang, Y. Liang, "Improved CNN-based facial landmarks tracking via ridge regression at 150 Fps on mobile devices," International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2017.

[15] C. Park, J. Jang, L. Zhang, et al., "Light-weight visual place recognition using convolutional neural network for mobile robots," IEEE International Conference on Consumer Electronics (ICCE), pp. 1-4, 2018.

[16] N. Garcia, G. Vogiatzis, "Learning Non-Metric Visual Similarity for Image Retrieval," Image and Vision Computing, vol. 82, pp. 18-25, 2017.

[17] S. Chopra, R. Hadsell, Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 539-546, 2005.

[18] G. Chechik, V. Sharma, U. Shalit, et al., "Large Scale Online Learning of Image Similarity Through Ranking," Journal of Machine Learning Research, vol. 11, no. 3, pp. 1109-1135, 2010.

[19] G. Guo-Dong, A. K. Jain, M. Wei-Ying, et al. "Learning similarity measure for natural image retrieval with relevance feedback," IEEE Transactions on Neural Networks, vol. 13, no. 4, pp.811-20, 2002.

[20] H. Azizpour, A. S. Razavian, J. Sullivan, et al., "From Generic to Specific Deep Representations for Visual Recognition," Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015.

[21] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, et al., "Fusion and Binarization of CNN Features for Robust Topological Localization across Seasons," International Conference on Intelligent Robots and Systems (IROS), pp. 4656-4663, 2016.

[22] O. Vysotska, C. Stachniss, "Relocalization under substantial appearance changes using hashing," Proceedings of the IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles, 2017.

[23] A. Torii, R. Arandjelovic, J. Sivic, et al., "24/7 Place Recognition by View Synthesis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 2, pp. 257-271, 2018.

[24] D. Anguelov, C. Dulong, D. Filip, et al., "Google Street View: Capturing the World at Street Level," Computer, vol. 43, no. 6, pp. 32-38, 2010.

[25] A. Geiger, P. Lenz, C. Stiller, et al., "Vision meets robotics: The KITTI dataset," International Journal of Robotics Research, vol. 32, no. 11, pp. 1231-1237, 2013.

[26] J. Luo, A. Pronobis, B. Caputo, et al., "The kth-idol2 reference KTH," CAS/CVAP, Tech. Rep, pp. 304, 2006.

[27] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," International Conference on Neural Information Processing Systems, pp. 1-9, 2012.

[28] H. V. Nguyen, L. Bai, "Cosine Similarity Metric Learning for Face Verification," Asian Conference on Computer Vision, pp. 709-720, 2010.

[29] M. J. Milford, G. F. Wyeth, "SeqSLAM: Visual Route-Based Navigation for Sunny Summer Days and Stormy Winter Nights," International Conference on Robotics and Automation (ICRA), pp. 1643-1649, 2012.

[30] R. Arandjelovic, P. Gronat, A. Torii, et al., "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 6, pp. 1437-1451, 2018.