2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)

Performance Evaluation of Deep Neural Networks in Detecting Loop Closure of Visual SLAM

Yong Chen[1], Lin Zuo[1], ChangHua Zhang[1], FengLian Liu[2], YunFeng Wu[1]

1. University of Electronic Science and Technology of China, Chengdu, China
2. State Grid Sichuan Electric Power Research Institute, Chengdu, China

Email: chenyslam@gmail.com, linzuo@uestc.edu.cn, zhangchanghua@uestc.edu.cn, liangrass@163.com, yfwu@uestc.edu.cn

***Abstract*—This paper concerns the problem of Loop Closure Detection (LCD) of visual Simultaneous Localization and Mapping (SLAM). The LCD is a crucial model to reduce the accumulative error in visual SLAM. The traditional LCD methods use hand-crafted features, which ignore useful information. We propose a LCD method based on Convolutional Neural Networks (CNNs) without any manual intervention for visual features. We compare and analyze several popular deep neural networks models for LCD. Two open datasets has been used to evaluate the performance of LCD in terms of mean-per-class accuracy. The results show that deep neural networks are feasible for LCD and the ResNet50 network outperforms the other deep neural networks.**

Keywords-mobile robot; visual SLAM; loop closure detection; deep learning

I. INTRODUCTION

With the development of artificial intelligence, mobile robots have become an important research topic. To perform the task of autonomous navigation, mobile robots are required to carry out mapping, localization, path planning, and other operations. Simultaneous Localization and Mapping (SLAM) [1] plays a vital role during the whole process. There are two categories of SLAM, i.e., visual SLAM and lidar SLAM. As an essential part of the visual SLAM, loop closure detection (LCD) can identify the places where the robot has passed before, so as to eliminate the cumulative errors.

Most of the existing LCD methods are based on the assumption of appearance invariance, which intrinsically compares the similarity between two adjecent images for visual systems. The Bag-of-Words (BoW) model [2], as the de-facto standard in LCD, has been widely implemented in the literature [3]. The BoW model can group many feature descriptors into a dictionary by *K*-means clustering. However, the hand-crafted features are required in the BoW model, which may lead to a low computational efficiency. Moreover, the hand-crafted features ignore some useful information in images, a low accuracy of LCD, therefore, may be produced.

The deep learning has been extensively investigated in the field of computer vision. The good performance of deep learning in computer vision has been demonstrated by a vast of existing works. However, to the best of our knowledge, the application of deep learning in LCD is rarely studied to date. Among many deep learning networks, the convolutional neural network (CNN), proposed by Lencun in 1989 [4], has become a hot topic in the field of computer vision. In view of the effective feature extraction of CNN, it is reasonable to apply deep learning networks to detect the loops for visual SLAM.

In this paper, we study the application of the CNNs in LCD. Our work takes advantage of the pre-trained CNN model developed by Zhang [5] to detect the loops. Our main contributions to the existing body of knowledge of LCD are itemized as follows:

(1) Several popular pre-trained CNN models are firstly applied to the LCD for visual SLAM.
(2) The Zero-phase Components Analysis (ZCA) whitening is introduced to process the CNN features, and the median filtering is included to improve the accuracy of processing results of pre-trained CNN models.

This paper is organized as follows: some related works about LCD are briefly described in Section II. Section III provides the details of our proposed method. Then, we present the experimental results of two open datasets in Section IV. Finally, the conclusions are given in Section V.

II. RELATED WORKS

The BoW model has been widely used in LCD for visual SLAM. It was first investigated in LCD by Fast Appearance-Based Mapping (FAB-MAP) [6]. There are three main steps for the procedure of BoW model. First, the algorithm (SIFT [7], SURF [8], or ORB [9]) is used to extract the visual lexical vectors from the images of different classes. These vectors represent the local invariant feature points in the image. Second, the *K*-means clustering algorithm is utilized to combine visual vocabulary with similar word meanings and construct a word list containing *K* words. Finally, the number of times that each word in the word list appears in the image is counted, so as to represent the image with a *k*-dimensional numerical vector. Nevertheless, the hand-crafted features in the BoW model have a limitation, that is, these features ignore some useful information in images, which may lead to a low accuracy of LCD.

Nowadays, the successful applications of CNN models in computer vision provide an alternative to solve the LCD problem. The methods of using CNN models for robotic applications can be divided into two main categories [10]: (1) training a CNN model and (2) directly using a pre-trained CNN model. Training a CNN model is a complex process and it is difficult for LCD. Gao et al. [11] proposed a deep features extraction method based on a well-trained neural network for

978-1-7281-1859-8/19/$31.00 ©2019 IEEE
DOI 10.1109/IHMSC.2019.10136

similarity metric. However, training the whole network is very time-consuming. It is unsuitable for a real-time SLAM system. Hou et al. [12] used a pre-trained CNN model to generate an image representation that suits to LCD in visual SLAM. The results proved that conv3 and pool5 had the best performance. However, the dimensions of CNN descriptors were very high in [12]. Xia et al. [13] compared the performance of several networks (PCANet [14], CaffeNet [15], AlexNet [16] and GoogLeNet [17]) and traditional methods (BoW and GIST [18]) in LCD. The results showed that those networks was suitable for LCD. But they used Support Vector Machine (SVM) to detect the loops, which may be inappropriate for current developments. Zhang et al. [5] used a open-source pre-trained CNN model, namely OverFeat, to extract features. However, the Overfeat model is rarely utilized in recent years because a lot of labels were required to train data.

Inception network is the milestone in the development of CNN classifiers. ResNet network [19] won the championship on the ImageNet Large Scale Visual Recognition Challenge in 2015. The ResNet network has been widely implemented in image detection, image segmentation, and image recognition. To our knowledge, both of the Inception network and the ResNet network have never been used in LCD for visual SLAM. In this work, in view of their good performances in computer vision, both of the Inception and ResNet network are applied to LCD for visual SLAM.

III. Proposed Method

In this section, the principle of our LCD approach is decribed in detail. The principle diagram of the proposed method is shown in Fig. 1. First, some pre-trained convolutional neural networks (ResNet50, ResNet101, ResNet152 and Inception-v4 [20]) are introduced to extract image features, while the ZCA whitening is introducec to reduce the dimension of feature vector. Then, the similarity matrix is constructed to detect the loops. Additionally, the median filtering algorithm is utilized to eliminate salt and pepper noise.

A. CNN Architecture

The ResNet network in [19] influenced the development direction of deep learning in academia and industry. It provides a "reference" to the input of each layer and forms a residual function. Residual error is designed to solve the degradation problem of deep learning networks, and the gradient disappearance problem, so that the performance of networks can be improved. The ResNet network provides five kinds of deep network structures, as shown in Table I, that are 18-layer, 34-layer, 50-layer, 101-layer, and 152-layer. It can be seen from the first column in Table I, all networks are divided into five parts: conv1, conv2_x, conv3_x, conv4_x, and conv5_x.

ResNet50 obtains good classification results on ImgeNet dataset, and the pre-training network is used in this paper. The ResNet50 contains 49 convolution layers and 1 full connection layer. In this work, we use the pre-trained models, i.e., ResNet50, ResNet101, and ResNet152 to extract features. Meanwhile, we also compare the performance of Resnet50 with the ResNet101 and ResNet152.

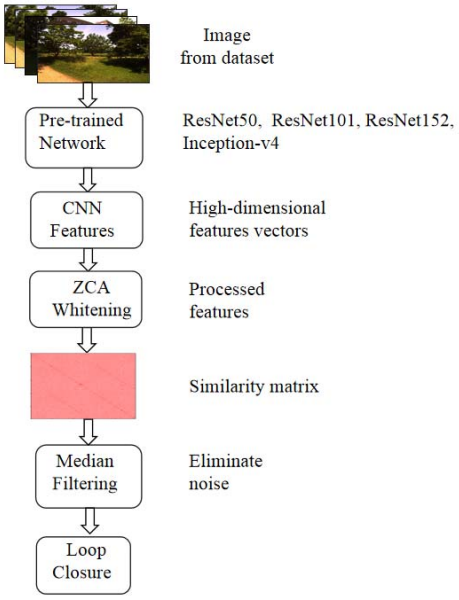


Figure 1. The principle diagram.

TABLE I. The Architecture of ResNet

Layer name	Out size	50-layer	101-layer	152-layer
Conv1	112×112	7×7, 64, stride 2		
		3×3 max, pool, stride2		
Con2_x	56×56	$\begin{bmatrix}1\times1, & 64\\ 3\times3, & 64\\ 1\times1, & 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, & 64\\ 3\times3, & 64\\ 1\times1, & 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, & 64\\ 3\times3, & 64\\ 1\times1, & 256\end{bmatrix}\times3$
Con3_x	28×28	$\begin{bmatrix}1\times1, & 128\\ 3\times3, & 128\\ 1\times1, & 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, & 128\\ 3\times3, & 128\\ 1\times1, & 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, & 128\\ 3\times3, & 128\\ 1\times1, & 512\end{bmatrix}\times8$
Con4_x	14×14	$\begin{bmatrix}1\times1, & 256\\ 3\times3, & 256\\ 1\times1, & 512\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, & 256\\ 3\times3, & 256\\ 1\times1, & 512\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, & 256\\ 3\times3, & 256\\ 1\times1, & 512\end{bmatrix}\times36$
Con5_x	7×7	$\begin{bmatrix}1\times1, & 512\\ 3\times3, & 512\\ 1\times1, & 1048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, & 512\\ 3\times3, & 512\\ 1\times1, & 1048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, & 512\\ 3\times3, & 512\\ 1\times1, & 1048\end{bmatrix}\times3$
1×1		average pool, 1000-d fc, softmax		
FLOPS		3.8×10^9	7.6×10^9	11.3×10^9

The Inception network is a milestone in the development of CNNs. It started from the GoogLeNet [21], and has gone through several iterations until the latest Inception-v4. In the Inception networks, the original model is trained in a partitioned way, while the Inception module can be standardized and simplified after migrating to the TensorFlow framework [22]. One of the merits of Inception-v4 is the introduction of the residual network structure to the Inception network, which integrates the advantages of the two networks and optimizes the original network structure. Therefore, we apply Inception v4 network to the LCD for visual SLAM.

B. Image Descriptors

Firstly, the whole image feature descriptions can be extracted from the pre-trained network model, and they are actually high-dimensional vectors at different layers.

Let us use $X^{(I)} = (x_1^{(I)}, x_2^{(I)}, ..., x_d^{(I)})$ to represent the feature vector of the input image I. Then, we perform the L_2 normalization as shown below:

$$(x_1, ..., x_d) \leftarrow \left(\frac{x_1}{\sqrt{\sum_{i=1}^{d} x_i^2}}, ..., \frac{x_d}{\sqrt{\sum_{i=1}^{d} x_i^2}} \right) \quad (1)$$

Suppose that the matrix **M** has n normalized feature vectors, and I is denoted as:

$$\mathbf{M} = \begin{bmatrix} -X^{I_1} - \\ -X^{I_2} - \\ \cdots \\ -X^{I_n} - \end{bmatrix} \in \mathbb{R}^{n \times d} \quad (2)$$

Because of the high dimension of feature vectors, it is time-consuming to compute the distance between images. Compared with the principal components analysis (PCA) whitening, the ZCA whitening makes the processed features much closer to the original features. Therefore, we use ZCA whitening to reduce the dimension of feature vectors. The detailed process of the ZCA whitening is illustrated in Algorithm 1.

Algorithm 1: ZCA whitening

Input : CNN feature vectors

Output : ZCA whitening vectors

1: $\bar{X} \leftarrow \frac{1}{n} \sum_{i=1}^{n} X^{I_i}$

2: For i =1 to n do

3: Replace X^{I_i} in **M** with $X^{I_i} - \bar{X}$

4: End For

5: cov $\leftarrow \mathbf{M}^T \mathbf{M}$

6: $[U, S, W] \leftarrow$ svd(cov)

7: $X_{reduced}^{I_i} \leftarrow X^{I_i} U[:, :d]$

8: $X_{whitened,j}^{I_i} \leftarrow \frac{X_{reduced}^{I_i}}{\sqrt{\lambda_j + \varepsilon}} U^T$

9: Return $X_{whitened,j}^{I_i}$

C. Similarity Matrix

Based on the pre-processed CNN features, the similarity matrix can be defined, which is used to measure the similarity between images. It is a good way to visualize LCD, as shown in Fig. 2. Each row of the matrix contains the value of the similarity, with a range of (0, 1). Different values are depicted by different colors. A higher value indicates more similarity between images. The value of 1 indicates that the detection is a loop closure.

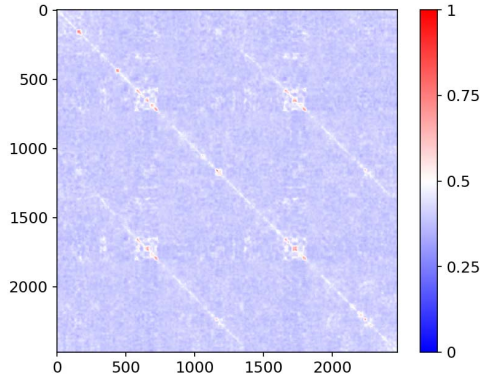


Figure 2. An example of LCD similarity matrix.

We use the Euclidean distance to compute differences between images i and j.

$$D_{i,j} = \left\| \frac{X_w^{I_i}}{\left\| X_w^{I_i} \right\|_2} - \frac{X_w^{I_j}}{\left\| X_w^{I_j} \right\|_2} \right\|_2 \quad (3)$$

Then, we can compute the normalized similarity between images i and j by:

$$S_{i,j} = 1 - \frac{D_{i,j}}{\max(D_{i,j})} \quad (4)$$

According to formula (4), the result of similarity scores is in the range [0,1]. Therefore, we can judge whether loop closure occurs or not by the value of similarity matrix. If the value of the similarity matrix exceeds the predefined threshold, the loop closure is occurred.

IV. Experiment Results

In this work, the experiment is conducted on Tensorflow [22], which is an open-source framework. We compare the similarity matrix of different deep neural networks models for LCD. Two open datasets are used to evaluate the performance of LCD in terms of mean-per-class accuracy.

A. Datasets

The proposed method are performed on two open datasets, i.e., New College and City Centre [6], which are commonly used in LCD for visual SLAM. When the robot walks through the outdoor urban environment, the images are collected by two cameras every 1.5m, and the lighting condition is stable. Then, the two datasets have 1237 and 1073 image pairs. The ground truth is also available according to true loop closures in these datasets, so it is convenient to measure the correctness of the proposed method. More details about these two datasets are shown in Table II.

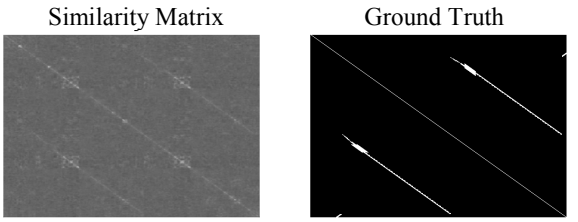
B. Comparision of Similarity Matrix

According to the formula (4), the similar score for images can be calculated and the corresponding similarity matrix can be obtained. Fig. 3 gives the results of the similarity matrices

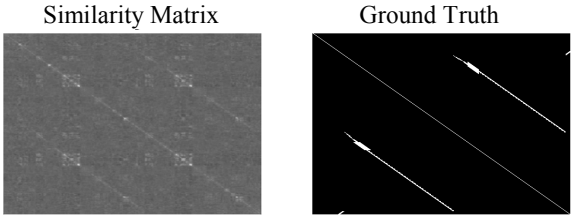
obtained by the proposed method and the ground-truth loops on City Centre dataset.

TABLE II. DATASETS DETAILS

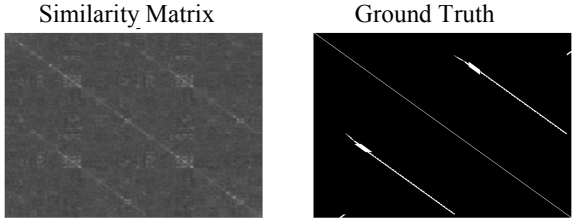
	Number	Image size	Image type	Sensor	Dataset open
City Centre	2474	640×480	RGB	Two cameras	Yes
New College	2146	640×480	RGB	Two cameras	Yes



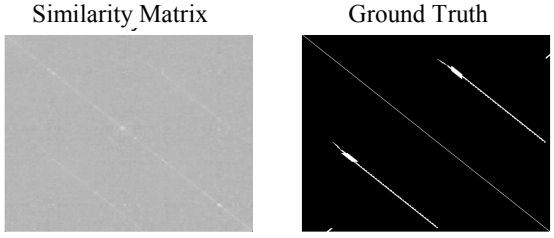
(a) ResNet50



(b) ResNet101



(c) ResNet152



(d) Inception-v4

Figure 3. Comparative results of similarity matrices

As shown in the above similarity matrices, colder colors show a smaller amount of similar for image pairs while warmer colors represent more similar for image pairs. Fig. 3 shows the proposed method is feasible to detect most of the loops and the ResNet50 network is better than the other three networks.

C. Evaluation Metrics

The detection of loop closure is also quantified by the Mean-Per-Class Accuracy (MPCA), the results are shown in Table III. As shown in Table III, these networks are all feasible for LCD of visual SLAM. Moreover, the results show that the ResNet50 network has better results than the other three deep learning networks for both datasets.

TABLE III. THE RESULTS OF MPCA

Network	ResNet50	ResNet101	ResNet152	Inception-v4
City Centre	0.832	0.785	0.786	0.593
New College	0.880	0.841	0.858	0.791

V. CONCLUTIONS

This paper investigated the pre-trained convolutional neural networks (ResNet50, ResNet101, ResNet152 and Inception-v4) in studying the LCD for visual SLAM. To our knowledge, this paper is the first time to introduce the above deep learning networks into LCD. The ZCA whitening was used to process CNN features, and the median filtering was introduced to eliminate salt and pepper noise. Finally, the similarity matrix was utilized to detect the possible loops in the datasets. We have done the comparative experiments of those pre-training neural networks on two open datasets. The results showed that the deep neural networks are feasible for LCD, and the ResNet50 network has the best performance. However, it is still difficult to apply deep neural networks in the real-time SLAM systems. It is one of our future directions.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation of China under Grants 61877009 and 61573081, Sichuan Provincial Science and Technology plan project under Grant 2018GZ0396, Sichuan Science and Technology Program under Grant 2019YFG0451.

REFERENCES

[1] Smith, C. Randall, and P. Cheeseman, “On the representation and estimation of spatial uncertainty,” The International Journal of Robotics Research, vol. 5, no. 4, pp. 56-68, 1986.

[2] G. Csurka, et al, “Visual categorization with bags of keypoints,” In Workshop on Statistical Learning in Computer Vision, ECCV. vol. 1. no. 1-22, pp. 1-2, 2004.

[3] R. Mur-Artal, J. M. M. Montiel, J. D. Tardos, “ORB-SLAM: A versatile and accurate monocular SLAM system,” IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, 2015.

[4] Y. Lecun, et al, “Backpropagation applied to handwritten Zip code recognition,” Neural Computation, vol. 1, no. 4, pp. 541-551, 1989.

[5] X. Zhang, Y. Su, and Y. Zhu , “Loop closure detection for visual SLAM systems using convolutional neural network,” In: Proceedings of 2017 23rd International Conference on Automation and Computing, 2017.

[6] M. Cummins and P. Newman, “FAB-MAP: Probabilistic localization and mapping in the space of appearance,” The International Journal of Robotics Research, vol. 27, pp. 647-665, 2008.

[7] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[8] H. Bay, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” In: Proccedings of Computer Vision–ECCV 2006, pp. 404-417, 2006.

[9] Rublee, et al, “ORB: An efficient alternative to SIFT or SURF,” pp. 2564-2571, 2011.

[10] D. Bai, et al, "CNN feature boosted seqslam for real-time loop closure detection," Chinese Journal of Electronics, vol. 27, no. 3, pp. 488-499, 2018.

[11] X. Gao, T. Zhang, "Unsupervised learning to detect loops using deep neural networks for visual SLAM system," Autonomous Robots, vol. 41, no.1, pp. 1-18, 2017 .

[12] Y. Hou, H. Zhang, and S. Zhou, "Convolutional neural network-based image representation for visual loop closure detection," In: Proceedings of 2015 IEEE International Conference on Information and Automation, pp. 2238-2245, 2015.

[13] S. Xia, et al, "An evaluation of deep learning in loop closure detection for visual SLAM," In: Proceedings of IEEE International Conference on Internet of Things IEEE, 2018.

[14] T. H, "PCANet: A simple deep learning baseline for image classification?," IEEE Transactions on Image Processing, vol. 24, no. 12, pp. 5017-5032, 2015.

[15] Y. Jia, et al, "Caffe: Convolutional architecture for fast feature embedding," 2014.

[16] A. Krizhevsky, I. Sutskever, G. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, no. 2, 2012.

[17] C. Szegedy, et al, "Going deeper with convolutions," In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[18] A. Oliva, A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," International Journal of Computer Vision, vol. 42, no. 3 ,pp. 145-175, 2001.

[19] K. He, et al, "Deep residual learning for image recognition," In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[20] C. Szegedy, et al, "Inception-v4, inception-resnet and the impact of residual connections on learning," In: Proceedings of Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[21] C. Szegedy, et al, "Going deeper with convolutions," In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[22] S. S. Girija, Tensorflow, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016.