AN END-TO-END SIAMESE CONVOLUTIONAL NEURAL NETWORK FOR LOOP CLOSURE DETECTION IN VISUAL SLAM SYSTEM

Hong Liu, Chenyang Zhao, Weipeng Huang, Wei Shi

Key Laboratory of Machine Perception, Peking University, Shenzhen Graduate School
{hongliu, chenyangzhao, wepon, pkusw}@pku.edu.cn

ABSTRACT

Loop closure detection is essential and important in visual simultaneous localization and mapping (SLAM) systems. Most existing methods typically utilize a separate feature extraction part and a similarity metric part.Compared to these methods, an end-to-end network is proposed in this paper to jointly optimize the two parts in a unified framework for further enhancing the interworking between these two parts. First, a two-branch siamese network is designed to learn respective features for each scene of an image pair. Then a hierarchical weighted distance (HWD) layer is proposed to fuse the multi-scale features of each convolutional module and calculate the distance between the image pair. Finally, by using the contrastive loss in the training process, the effective feature representation and similarity metric can be learned simultaneously. Experiments on several open datasets illustrate the superior performance of our approach and demonstrate that the end-to-end network is feasible to conduct the loop closure detection in real time and provides an implementable method for visual SLAM systems.

Index Terms— Simultaneous Localization and Mapping, Loop Closure Detection, End-to-end Network, Siamese Convolutional Neural Network

1. INTRODUCTION

Loop closure detection aims to recognize the places where a mobile robot has been. It is one of the most significant requirements for visual simultaneous localization and mapping (SLAM) system which is utilized to map an unknown environment while simultaneously localizing the robot [1]. Correct and efficient loop closure detection can be used for robot relocation [2] after tracking failure. More importantly, it can add right constraints to pose graph in a SLAM algorithm and contribute to build a stable map by reducing the error and drift that accumulate over time [3]. The image-to-image methods and image-to-map methods are two main categories of methods for loop closure detection. It is concluded that image-to-image (or appearance-based) methods scale better than image-to-map methods [4] for large environment. Thus in visual SLAM systems, the basic techniques detect loop closures through matching the new frame with images in the database built by the robot online.

Relation to prior work: In image-to-image methods, the bag-of-words (BoW) model [5] is extensively used in state-of-the-art traditional algorithms [6–9]. Dorian *et al.* [8] used bag of words obtained from FAST+BRIEF features [10, 11] and built a vocabulary tree to speed up feature matching for detecting loops in real time. The DBoW open source library built based on [8] has become the standard baseline regarding loop closure detection because of its significant performance in the representative ORB-SLAM system [12]. However, generally empirically designed by researchers, the handcrafted features used by BoW-based traditional methods have various limitations. Recently, the development of deep learning [13] brings an alternative way to extract more complex feature structures in an image. It also provides new thoughts for loop closure detection problem using convolutional neural network (CNN) [14–16]. Gao *et al.* [15] used a well-trained stacked denoising auto-encoder (SDA) neural network to extract deep features for similarity metric. Although the CNN-based method can detect loops with a satisfactory precision, it cannot satisfy the real time requirement in the SLAM system because of the long time cost of feature extraction from the multi-layer network and the similarity calculation. In this paper, different from the existing methods which treat the feature extraction and the similarity metric as two parts, we propose an end-to-end siamese network to jointly optimize the two separate parts in a unified framework. The end-to-end network can measure the similarity of two places from image pixels directly and speed up the loop detection effectively.

Deep metric learning methods using siamese network have shown outstanding performance in many similarity metric tasks [17–19]. Motivated by these works, we design the two-branch siamese network with the contrastive loss to learn an optimal metric towards the distance indicating whether two images are captured from the same place. By learning

This work is supported by National Natural Science Foundation of China (NSFC, No.U1613209,61340046,61673030), Natural Science Foundation of Guangdong Province (No.2015A030311034), Scientific Research Project of Guangdong Province (No.2015B010919004), Specialized Research Fund for Strategic and Prospective Industrial Development of Shenzhen City (No.ZLZBCXLJZI20160729020003), Scientific Research Project of Shenzhen City (JCYJ20170306164738129), Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality (ZDSYS201703031405467).

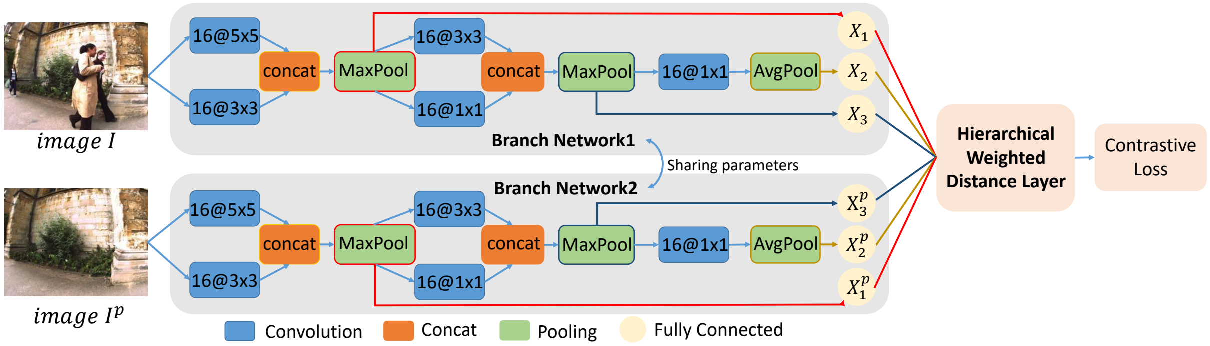

Fig. 1. Framework of the proposed end-to-end siamese convolutional neural network.

object features and metric simultaneously, the end-to-end network can reduce the redundant features and make features to align the metric better. In addition, a hierarchical weighted distance (HWD) layer is proposed to fuse multi-scale features from each convolutional module and enhance the effect of the two-branch network.

2. PROPOSED METHOD

In this section, we first describe the overview framework of our method and the corresponding architecture of the two-branch siamese convolutional neural network. Then, the proposed HWD layer is given in details. Finally, the contrastive loss is introduced as the cost function.

2.1. Overall Framework

The framework of the proposed end-to-end siamese convolutional neural network is shown in Fig. 1. The network has two symmetry branch networks connected by a HWD layer which outputs a distance evaluating the similarity of the features from two branches. Let $< I, I^p >$ represents an image pair sample, and its label is 1 or 0 which depends on whether the two images in the pair are captured from the same place. Each image is fed into a branch of convolutional neural network (CNN). Through the network, the image pair is mapped to a learned feature space, where X_i and X_i^p represent features from the *i-th* intermediate layer (section 2.2). Then the final distance of the image pair will be calculated by the proposed HWD layer (section 2.3) using the features from each intermediate layer. The weights in the HWD layer will be self-learning by the back propagation algorithm. We train the network using a contrastive loss function which aims to minimize the distance of positive scene pairs and maximize the distance of the negatives independently. Through the training process, the feature extraction and the similarity metric are jointly optimized and then we can obtain the well-trained network with image pairs as input and distances as output, in the end-to-end way.

2.2. Two-branch Convolutional Neural Network

The two symmetry branch networks in this paper are shown inFig.1 in detail. The two CNN branches share the same structure and parameters to extract features from two inputs. The branch networks support other effective structures and we use the branches set up by ourselves. It is composed of two inception modules [20], one convolutional layer and three fully connected layers. The two inception modules have the same structure which consists of 16 convolutional kernels in two kinds of size and the output feature maps are then concatenated together followed by a 2×2 MaxPooling layer. The filter size of the first inception module is 5×5 and 3×3, while the second is 3×3 and 1×1. In these two inception modules, kernels in different sizes are adopted to capture features with different resolution. Then another convolutional layer with 16 kernels of 1×1 is employed to reduce the depth of the feature map, also followed by a 2×2 AveragePooling layer. There are three fully connected layers and each one connects to a specific pooling layer. Each fully connected layer generates a 256 dimensional feature describing the respective information of different convolutional modules. And ReLU neuron [21] is used as activation function for each layer. Overall, through the two sub-networks, each input image is represented as (X_1, X_2, X_3), where X_1 and X_3 indicate 256 dimensional features from the first and second inception modules, X_2 indicates features from the whole network, as shown in Fig. 1. These features are then sent into the HWD layer.

2.3. Hierarchical Weighted Distance Layer

The hierarchical weighted distance (HWD) layer is designed to fuse the complementary information of different convolutional modules in score level by learning weights for distance of each feature pair. Then the output is used to form the metric-cost part together with contrastive loss function in training process.

Given an image pair $< I, I^p >$ and its corresponding features from different intermediate layers, the important normalization step is performed on X_i, though $L2$-norm, to constrain them to live on the 256 dimensional hypersphere:

$$\hat{X}_i = \frac{X_i}{\|X_i\|}. \tag{1}$$

Then the square Euclidean distance of *i-th* feature between I and I^p is calculated by:

$$d_i = \|\hat{X}_i - \hat{X}_i^p\|_2^2. \tag{2}$$

After that, the distances d_i are weighted and combined to produce the distance between I and I_p :

$$d = \sum_{i=1}^{3} w_i \|\hat{X}_i - \hat{X}_i^p\|_2^2 = \sum_{i=1}^{3} w_i d_i$$
$$s.t. \sum_{i=1}^{3} w_i = 1, \tag{3}$$

where w_i is the weight parameter measuring the importance of each feature. The contribution of features in different convolutional module cannot be predicted and it is infeasible to tune the weights by grid search or random search due to the large computation in training process. So we design the HWD layer to learn w_i automatically by standard back propagation algorithm, just the same way as other parameters in convolutional layer and fully connected layer.

Different convolutional modules can extract different scales and kinds of features. Low-level layers extract more detailed information such as edges and colors, while high-level layers contain more semantic information such as objects and backgrounds. The features from different intermediate layers may have better performance than the feature only from the last fully connected layers. The HWD layer can take each hierarchical convolutional module into consideration and learn their importance automatically.

2.4. Contrastive Loss

Contrastive loss[16] is adopted as the cost function due to its effectiveness for the pair data in siamese network. The function is defined as:

$$L = yd^2 + (1-y)\max(margin - d, 0)^2, \tag{4}$$

where $y = 1$ for positive pairs, $y = 0$ for negative ones, and margin is the threshold. The distance d is calculated by the proposed HWD layer. Either the distance between the positive pair is not 0 or the distance of negative pair is less than $margin$, there will be a loss. So by minimizing the loss, the distances between positive samples will be shrunken and the distances between the negative pairs will be enlarged to approach $margin$ value.

3. EXPERIMENTS

In the experiments, we evaluate the proposed end-to-end siamese convolutional neural network on New College dataset [6], City Centre dataset [6] and TUM open dataset [22]. The state-of-the-art traditional method DBoW [8] and the CNN-based method SDA [15] are compared in the experimental analysis. The methodology we followed to evaluate our algorithm is described in section 3.1. And the experimental results are presented and analyzed in section 3.2.

3.1. Methodology

Datasets and ground truth: The benchmark City Centre and New College datasets both published in [6] are widely used in visual SLAM research, especially in loop closure detection. The two sequences of the two datasets consist of 2474 and 2146 outdoor urban images respectively collected by a mobile robot with two cameras on the left and right side. Ground truth loops which are hand-labeled as a binary matrix are also available.

The TUM open dataset has many RGB-D sequences with ground truth trajectories. We choose the indoor office scene freiburg3_long_office_household (fr3_office for short) sequence to experiment. The end of the trajectory is overlapped with the beginning so that there is a large loop closure. The TUM dataset does not provide ground truth loops, so we generate the true loops using ground truth trajectory by computing distances between camera poses of image pairs. That the distance is small enough means the position and heading of the camera is close, then we mark the image pair as a ground truth loop.

Correctness measure: The correctness of the loop detection results is measured with the precision-recall (P-R) curve. Precision means the ratio between number of correct loops and number of all loops detected, while recall means ratio between correct detections and total loops in ground truth.

Train and test: To train and test the end-to-end network properly, one in every five images is selected as a key frame so that every dataset is separated into 4/5 to train and 1/5 to test. The vocabulary of one million words used for DBoW method is generated from all images in train sets and 10k images of an independent dataset (Bovisa 2008-09-01[1]), for keeping same with [8] where DBoW is proposed. And equally, we train the end-to-end network with Bovisa 2008-09-01 first, then finetune with train sets separately. As for testing, image pairs are composed of every two images in the same test set and sent to compute distances. A distance threshold is applied to determine whether the loop closure has occurred, and a precision and recall pair result can be got after all images in the dataset are considered. By varying the distance threshold, we can then produce a P-R curve.

The experiment results of the SDA method are from reference [15]. The experiments of DBoW method are run by utilizing the up to date DBoW3 C++ code[2] with a i7-6700 3.40GHz CPU and 8G memory and our method is tested on the same CPU as well as a NVIDIA GeForce GTX 1080 GPU.

3.2. Experimental Results

Evaluation of the proposed HWD layer. To evaluate the effect of fusing features from each hierarchical convolutional module, the experiments without the HWD layer are conducted. Without the HWD layer, the feature X_2 and X_2^p (see Fig. 1 and section 2.2) which are only from the last fully connected layer are adopted to calculate distance of the image pair. The P-R curves of our method without the HDW layer is shown in Fig. 2. It can be seen that better performance is achieved with the usage of the HWD layer. The fusion of features from different hierarchical convolutional modules has a more power-

[1] https://www.rawseeds.org/rs/datasets

[2] https://github.com/rmsalinas/DBoW3

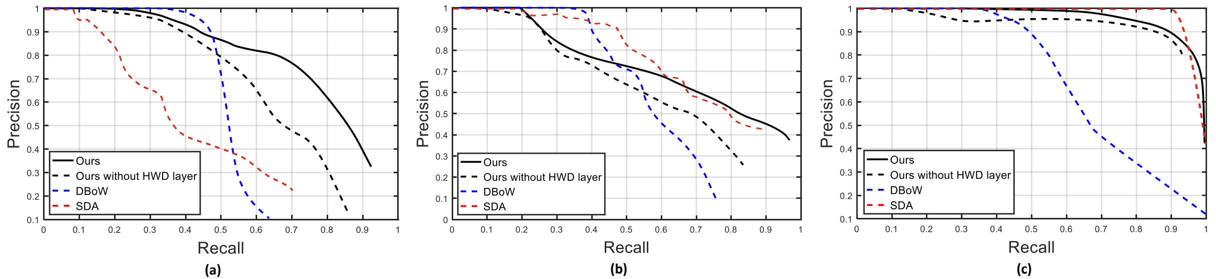

Fig. 2. Precision-recall curve comparison.**(a)** P-R curve on City Centre dataset. **(b)** P-R curve on New College dataset. **(c)** P-R curve on TUM fr3_office dataset (Best viewed in color).

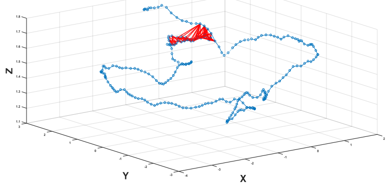

Fig. 3. The trajectory of fr3_office with detected loops (Best viewed in color).

ful representation of the scene image. The comparison proves the effectiveness of our HWD layer.

Comparison with state-of-the-arts. The loop closure detection results are shown by P-R curve compared with the state-of-the-art traditional method DBoW, and the CNN-based SDA method. Fig. 2(a) and Fig. 2(b) show the P-R curves on the City Centre dataset and the New College dataset, respectively. For both the two outdoor datasets, D-BoW has a higher precision at the low recall. However, when recall rate is larger than 0.45, the precision of DBoW has a decrease rapidly and our method obtains a much better precision. Compared to the SDA, our method has a significant progress on the City Centre dataset. And on the New College dataset, although the SDA has a better performance at the low recall, our method gives the similar precision as the recall rate increasing.

Fig. 2(c) illustrates the P-R curves on fr3_office dataset. The SDA method has an advantage that the recall rate at 100 % precision achieves 0.9, while compared to the DBoW method, our method performs better both on recall rate and precision rate. And our method also has a greater effect on this indoor dataset than on the two outdoor datasets. The fr3_office is a stable office scene with ample texture, while the City Centre and New College datasets are much more dynamic. Both the changing illumination and the passerby like the image pair exemplified in Fig. 1 increase the difficulty of loop detection.

The detected loop closures on the TUM fr3_office dataset. The trajectory and detected loop closures of fr3_office are shown in Fig. 3. The blue line shows the trajectory made up of positions of key frames and the red lines connect frame pairs considered as loops. It shows that the end of the trajectory is overlapped with the beginning and there is no false positive loops. The experiment result indicates that our method is effective enough in stable indoor situation.

Table 1. Comparison of computational time per image pair

Method	DBoW	SDA	Ours	
	CPU	CPU	CPU	GPU
Time(s)	0.00660	0.16	**0.00474**	0.00036

Calculation efficiency. The efficiency of descriptor extraction and similarity computation is an important consideration in loop closure task because of the real time requirement in SLAM systems. We measure the efficiency by the time the algorithm takes to process the image pair and gain the similarity, except the time loading images and the vocabulary or the well-trained model.

The average image pair processing times of different methods are listed in Table 1. It is shown that for the CPU-based processing, our method is much more efficient than the CNN-based SDA method and slightly faster than the DBoW method. When the image pair passes through the end-to-end network on our GPU, the average time reduces to 0.00036s per image pair, almost 10 times faster than processing on CPU. Since there are only five convolutional layers with 16 kernels in each layer in our end-to-end network, it has fewer parameters and less time consumption than the SDA. Due to the DBoW method is a typical loop closure detection method used on the representative real time SLAM system, the same efficiency with DBoW proves that our method can satisfy the processing speed requirement of SLAM systems.

4. CONCLUSION

This paper proposes an end-to-end siamese convolutional neural network for loop closure detection problem in SLAM system. To the best of our knowledge, it is the first time that the siamese network-based deep metric learning has been attempted on the loop closure detection problem, and to learn a similarity metric of two places from image pixel directly. Meanwhile, a hierarchical weighted distance layer is applied to learn weights for features of different convolution modules. Experimental results on three public datasets demonstrate that our approach is feasible for the loop closure detection problem. As a deep learning-based method, the well-trained end-to-end network still needs to fine-tune to adapt a new environment when it is applied to a real physical condition. Our future work will focus on how to make use of the deep metric learning in physical loop closure detection, which realize online training and then apply it in the visual SLAM system.

5. REFERENCES

[1] Hugh Durrant-Whyte and Tim Bailey, "Simultaneous localization and mapping: part I," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[2] Hong Liu, Weibo Huang, and Zhi Wang, "A novel retracking strategy for monocular slam," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 1942–1946.

[3] Karl Granström, Thomas B Schön, Juan I Nieto, and Fabio T Ramos, "Learning to close loops from range data," *The international journal of robotics research*, vol. 30, no. 14, pp. 1728–1754, 2011.

[4] Brian Williams, Mark Cummins, José Neira, Paul Newman, Ian Reid, and Juan Tardós, "A comparison of loop closing techniques in monocular slam," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1188–1197, 2009.

[5] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision*, 2003, vol. 2, pp. 1470–1477.

[6] Mark Cummins and Paul Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[7] Dorian Gálvez-López and Juan D Tardos, "Real-time loop detection with bags of binary words," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 51–58.

[8] Dorian Gálvez-López and Juan D Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[9] Adrien Angeli, David Filliat, Stéphane Doncieux, and Jean-Arcady Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1027–1037, 2008.

[10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua, "Brief: Binary robust independent elementary features," in *European Conference on Computer Vision*, 2010, vol. 6314, pp. 778–792.

[11] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*. 2006, vol. 1, pp. 430–443, Springer.

[12] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos, "ORB-SLAM: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[13] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[14] Yi Hou, Zhang Hong, and Zhou Shilin, "Convolutional neural network-based image representation for visual loop closure detection," in *IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2015, pp. 2238–2245.

[15] Xiang Gao and Tao Zhang, "Unsupervised learning to detect loops using deep neural networks for visual slam system," *Autonomous Robots*, vol. 41, no. 1, pp. 1–18, 2017.

[16] Qin Li, Ke Li, Xiong You, Shuhui Bu, and Zhenbao Liu, "Place recognition based on deep feature and adaptive weighting of similarity matrix," *Neurocomputing*, vol. 199, pp. 114–127, 2016.

[17] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in Neural Information Processing Systems*, 1994, pp. 737–744.

[18] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li, "Deep metric learning for person re-identification," in *IEEE International Conference on Pattern Recognition (ICPR)*. IEEE, 2014, pp. 34–39.

[19] Weihong Wang, Jie Yang, Jianwei Xiao, Sheng Li, and Dixin Zhou, "Face recognition based on deep learning," in *International Conference on Human Centered Computing*, 2014, pp. 812–820.

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] Jrgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *IEEE International Conference on Intelligent Robot Systems (IROS)*, 2012, pp. 573–580.