Binary GAN based Approach for Unsupervised Loop Closure Detection in Autonomous Unmanned Systems

Sheng Jin[a], Hui Yang[a], Liang Chen[a,*], Yu Gao[a], Rongchuan Sun[a], Seán McLoone[b]

[a] *School of Mechanical and Electric Engineering, Soochow University, Suzhou 215021, PR China*
[b] *School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, UK*
(e-mail: chenl@suda.edu.cn).

Abstract: Inspired by generative adversarial network (GAN), we propose a novel unsupervised approach for loop closure detection in autonomous unmanned systems. A binary GAN model dedicated to mobile application scenarios is designed to obtain binary feature descriptors, which are further incorporated into the most commonly used Bag of Visual Words (BoVW) model for loop closure detection. Compared with those hand-crafted features like SIFT and ORB, the performance of loop closure detection in complex environments with strong viewpoint and condition changes can be greatly improved. Compared with existing supervised approach based on convolutional neural network like AlexNet and AMOSNet, the cost-expensive task of supervised data annotation is totally avoided, which make the proposed approach more practical.

Keywords: loop closure detection, bag of visual word, vSLAM, binary GAN, feature extraction.

1. INTRODUCTION

Research on visual SLAM (vSLAM) is blooming due to its importance in tremendous autonomous systems such as mobile robots and automatic driving. A standard vSLAM system consists of the following modules: visual odometer, back-end graph optimization, loop closure detection and mapping. Due to the cumulative error of the visual odometer in the vSLAM, the map obtained when the robot moves back to the starting point is not consistent.

Loop closure detection in vSLAM is a challenging problem. A good method should provide key constraints for the back-end pose graph optimization (Kummerle et al. (2013)) and eliminate the cumulative error in the mapping process. In most cases, the image dataset is very large and the time for feature matching is limited due to the limited computing resources. Moreover, the appearance of the surrounding environment may change greatly, e.g. viewpoint changes and condition changes in season, illumination and dynamic objects (Zaffar et al. (2019)).

In traditional loop closure detection, Bag of Visual Words (BoVW) is the most widely used method, and its core is to extract the local feature descriptors by hand-crafted feature extraction methods, such as scale-invariant feature transform (SIFT) (Lowe et al. (2004)), speeded-up robust feature (SURF) (Bay et al. (2006)), oriented features from accelerated segment test, rotated binary robust-independent elementary feature (ORB) (Rublee et al. (2012)) etc. These local feature descriptors can be clustered by the K-means algorithm in an N-dimensional feature space. Each cluster center is called a visual word. All visual words make up a vocabulary, which can be represented by a k-d tree. In the testing process, BoVW method extracts the local feature descriptors and assigns each descriptor to the closest visual word from the vocabulary. Then, the method can obtain a feature histogram called the BoVW vector. The Term Frequency-inverse Document Frequency (TF-IDF) algorithm could be used to evaluate the importance of each visual word and give each visual word a different weight. At last, the method uses the BoVW vector and the weights to score the similarity between two images. The overall process of the BoVW model is shown in Fig. 1.

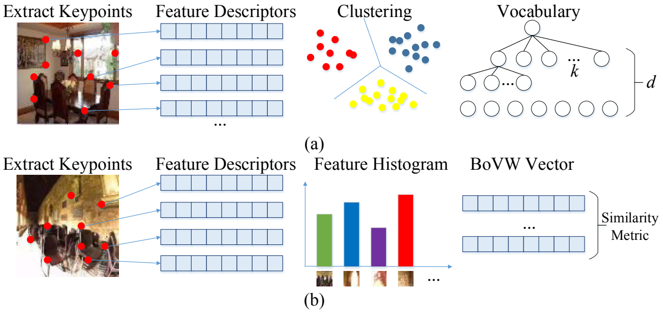

Fig. 1. BoVW model. (a) Training, (b) Testing.

However, the traditional BoVW model relies on hand-crafted features like SIFT, SURF and ORB which are not effective in complex environments with strong viewpoint and condition changes. In recent years, deep learning has become a new technology to extract more comprehensive features from images. In literature, deep neural networks used in loop closure detection problems have shown better results than those hand-crafted feature methods. Hou et al. (2015) used the Places-CNN model trained on the large-scale scene classification dataset to explore the performance of different layers of CNNs for loop closure detection in vSLAM. He found that compared with the traditional hand-crafted features, the features extracted by the CNN model can achieve better results under complex appearance changes. Sunderhauf et al. (2015) studied the performance by using

AlexNet (Krizhevsky and Hinton (2012)) for loop closure detection. Chen et al. (2017) used AMOSNet and HybridNet models which were trained on the large-scale scene classification dataset to extract the deep features of images. Olid et al. (2018) used a triplet loss scheme to train CNN models to extract features. Compared with the BoVW model, the deep learning based methods are superior in feature extraction, and can greatly improve the ability of loop closure detection in complex environments. However, most of these deep learning based method are supervised approaches, which need a large amount of labeled data for training. This is cost-expensive and not practical for real applications.

Therefore, it is of great significance to have an unsupervised approach for loop closure detection which is still rare in literature. Gao and Zhang (2017) used a stacked denoising autoencoder (SDA) method to support unsupervised feature extraction for loop closure detection. However, this method was an offline method and it was only verified when the training set and the test set were the same. Recently, generative adversarial network (GAN) is also introduced into unsupervised loop closure detection problem (Shin et al. (2019)). But the extracted feature descriptors are high-dimensional and take up more memory that would not make any contribution to the mobile applications.

In this paper, we propose a novel approach for unsupervised loop closure detection in vSLAM. A binary GAN model is designed to obtain binary feature descriptors of the images. These local features are more distinctive than those hand-crafted features. Therefore, the performance of loop closure detection in complex environments can be greatly improved. The binary GAN model is trained in an adversarial learning manner without any labelled data. Moreover, we customized two additional loss functions for loop closure detection, i.e. the binarization representation entropy (BRE) loss function (Cao et al. (2018)) and the distance propagating (DP) loss function. As a result, the local feature descriptors extracted by our approach are binary and more discriminative than those nonbinary descriptors. The results show that the approach is not only more sensitive to strong variations in condition and viewpoint, but also efficient and more suitable for mobile applications like autonomous driving with limited computing resource and storage space.

2. THE APPROACH

2.1 Overall Process

Fig. 2. shows the overall framework of our proposed approach for unsupervised loop closure detection based on a binary GAN model. It is mainly divided into two parts: model training process and loop closure detection process. During the model training process, we first input training images which are unrelated to any scene that the loop closure detection process may encounter. Then, we construct the local image patches by the method described in part 2.4. Subsequently, we carry out unsupervised learning to train the binary GAN model. Then, we use the K-means algorithm to cluster the derived descriptors by GAN model and build the vocabulary tree that are similar with the original BoVW method. In the loop closure detection process, the current frame acquired by the robot in real time is taken as input. We use the well trained binary GAN model to extract the image features from the local patches. Finally, loop closure detection can be carried out by the BoVW model.

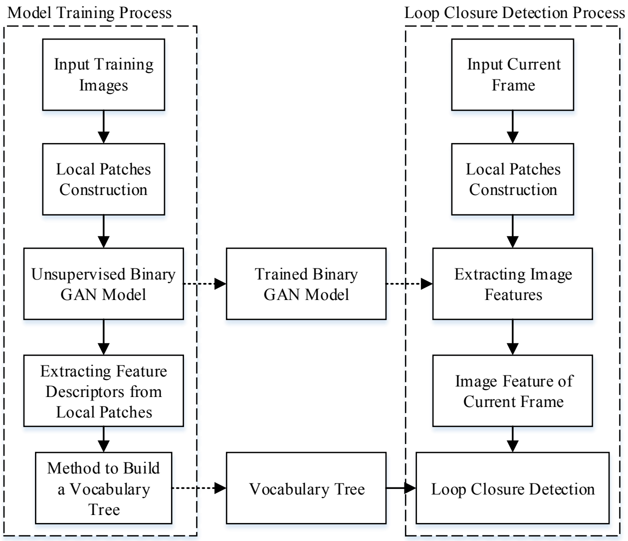
Model Training Process
Loop Closure Detection Process
Input Training Images
Local Patches Construction
Unsupervised Binary GAN Model
Extracting Feature Descriptors from Local Patches
Method to Build a Vocabulary Tree
Trained Binary GAN Model
Vocabulary Tree
Input Current Frame
Local Patches Construction
Extracting Image Features
Image Feature of Current Frame
Loop Closure Detection

Fig. 2. The overall framework of the proposed approach.

Instead of using traditional hand-crafted feature extraction methods, the kernel of our approach is using a binary GAN model to extract binary local feature descriptors. In the following sections, we first briefly review the GAN. Then, we discuss the details of the binary GAN model for loop closure detection. Subsequently, we describe the image patch construction method from the training image dataset. More implementation details are outlined at the end of this part.

2.2 The GAN Model

GAN is composed of two competing networks, i.e. generator G and discriminator D as shown in Fig.3. The generator G is trained to sample from the data distribution P_{data}(x) by transforming the random noise z. The discriminator D is trained to distinguish whether the input samples are generated by the generator or from the real data distribution P_{data}(x). The training problem can be formulated as follows:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim P_{data}(x)}[\log(D(x))] + E_{z\sim P_z(z)}[\log(1-D(G(z)))] \qquad (1)$$

where x represents a real sample. $D(x)$ represents the probability that the discriminator D judges that x is a real sample. z represents the input random noise. $G(z)$ represents the sample generated from the noise z by the generator. $D(G(z))$ represents the probability that the discriminator D judges that $G(z)$ is a real sample. The goal of generator G is to make the generated sample as close as possible to the real sample. The closer $D(G(z))$ is to 1, the smaller $V(D, G)$ will become. The goal of discriminator D is to make D(x) close to 1 and $D(G(z))$ close to 0.

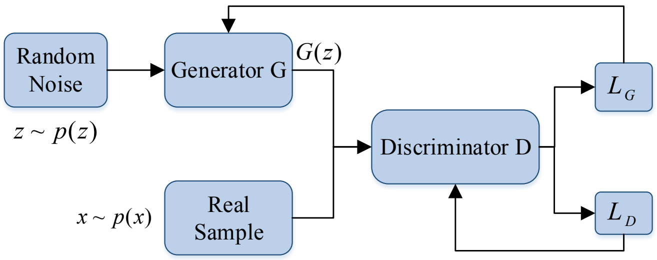

Fig. 3. Structure of the GAN.

In recent years, GAN has gained a significant amount of attention in image generation, image restoration and image classification. GAN can also be used to extract features, which was initially discussed in (Radford et al. (2016)), i.e. to use the discriminator *D* as a feature extractor. During the training process, the discriminator *D* is trained to extract more diverse and essential features. The training goal of generator *G* (Salimans et al. (2016)) is,

$$L_G = \left\| E_{x\sim p_{data}(x)} f(x) - E_{z\sim p_z(z)} f(G(z)) \right\|_2^2 \qquad (2)$$

where *f*(*x*) represents the feature descriptors obtained. Radford et al. (2016) showed that the best feature representations in GANs are extracted from the intermediate layer of the discriminator, and they are high-dimensional. Consider the problem of loop closure detection, however, it's expected to use more discriminative low-dimensional feature descriptors to accommodate the limited computing resource in embedded systems. Hence, it is encouraged to adopt binary descriptors if possible, which have several attractive properties, e.g. compactness, fast implementation, etc. In order to meet these requirements, in this paper, we adopt the BRE loss function and DP loss function to convert the high-dimensional feature descriptor in intermediate layer of the discriminator into low-dimensional binary representation. The above idea is realized through the adversarial training based on a binary GAN model described below.

2.3 Binary GAN model

The structure of the binary GAN proposed is illustrated in Fig. 4. The discriminator of binary GAN consists of 7 convolutional layers, two network-in-network (NiN) layers (Lin et al. (2013)) and one discriminative layer. For the convolutional network, the size of the kernel is 3x3 and the stride is [1, 1, 2, 1, 1, 2, 1], and the channel number is [96, 96, 96, 128, 128, 128, 128] for each layer. In our approach, we define the first NiN layer composed of 256 neurons as the low-dimensional feature and define the last convolutional layer composed of 9216 neurons as the high-dimensional feature. The generator of binary GAN consists of one fully connected layer and three deconvolutional layers with a kernel size 3×3, and the channel number is [256, 128, 3] for each layer.

In the proposed binary GAN, the definition of the loss function is crucial for adversarial training and should reflect the nature of the loop closure detection problem. Binary feature representations are more efficient and more suitable for embedded AI applications. In the problem of loop closure detection, it is expected to extract more discriminative low-dimensional binary feature descriptors This is important to enable a fast and stable detection under viewpoint changes and condition changes. However, the high-dimensional feature descriptors are more discriminative and reliabable for feature representation (Radford et al. (2016)). For this purpose, the Hamming distance can be propagated from the high-dimensional feature space to the low-dimensional feature space, while their distance relationship remains the same. These binary descriptors can be obtained from the original feature descriptors by using the following binary activation function (Dong et al. (2018)).

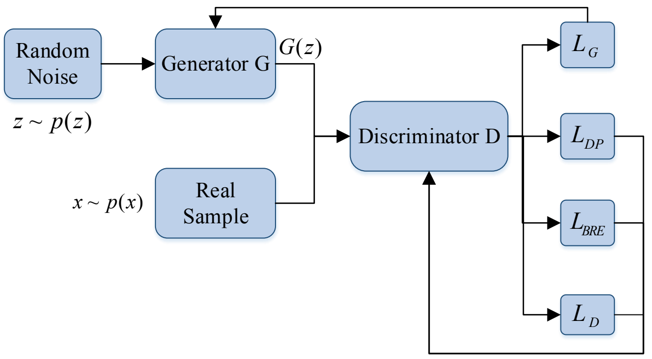

Fig. 4. Structure of the binary GAN.

$$BAF(x) = u(\sigma(x) - 0.5) \qquad (3)$$

where $u(\cdot)$ denotes the step function and $\sigma(\cdot)$ is the sigmoid function. In this way, we can turn a tensor whose values are in [0,1] into a tensor with values in {0,1}. We use the derivative of the sigmoid function in the backward pass.

Let L(x) and H(x) represent the feature descriptors extracted from the low-dimensional and high-dimensional intermediate layer of the discriminator with the number of neurons *K* and *M*, respectively. They can be further converted into binary descriptors b_L and b_H using the binary activation function. Without loss of generality, consider two binary descriptors b_1 and b_2, the Hamming distance can be represented as,

$$hamming\_dist(b_1, b_2) = A - b_1^T b_2 - (b_1 - 1)^T (b_2 - 1) \qquad (4)$$

where *A* represents the dimension of the binary descriptor, and the value of dot product $b_1^T b_2 + (b_1 - 1)^T (b_2 - 1)$ can reflect the distance between two binary descriptors, which can be defined as,

$$Dot_{b_1, b_2} = b_1^T b_2 + (b_1 - 1)^T (b_2 - 1) \qquad (5)$$

Given a min-batch $\{x_1, ..., x_N\}$, we define the distance propagation function as follows,

$$L_{DP} = \frac{1}{N(N-1)} \cdot \sum_{k,j=1,k\neq j}^{N} \left| \frac{Dot_{k,j}^L}{K} - \frac{Dot_{k,j}^H}{M} \right| \qquad (6)$$

where $Dot_{k,j}^L$ represents the value of dot product between the k^{th} binary descriptor and the j^{th} binary descriptor in the low-dimensional feature space. $Dot_{k,j}^{H}$ represents the value of dot product in the high-dimensional feature space.

The BRE loss function was first proposed in (Cao et al. (2018)). In this approach, we use the marginal entropy L_{ME} and the activation correlation L_{AC} to increase the diversity of the binary descriptors. Finally, the BRE loss function L_{BRE} is the sum of L_{ME} and L_{AC}. The binary GAN is trained in an unsupervised manner by alternately training discriminator D and generator G. The discriminative loss of discriminator D is defined as follows,

$$L_D = -E_{x\sim p_{data}(x)}[\log(D(x))] - E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{7}$$

The objective of training discriminator D is as follows,

$$L = L_D + \lambda_{DP}\cdot L_{DP} + \lambda_{BRE}\cdot L_{BRE} \tag{8}$$

where λ_{DP} and λ_{BRE} are hyperparameters.

Table 1 shows the hyperparameters to train the GAN model, and the Adam optimizer (Diederik et al. (2014)) is used.

Table 1 Hyperparameters of the model.

Parameter	Value	Parameter	Value
Batch size	25	λ_{DP}	0.5
Epoch	100	λ_{BRE}	0.1
Learning rate	0.0003	Momentum	0.5

2.4 Local Image Patch

In order to train the binary GAN model, we need to construct local image patches from the training dataset. For each training image in loop closure detection, we extract about 70 SURF features and discard some SURF features which are near image boundaries. Then, we construct local image patches of 32×32 pixels, which are centred at the remaining SURF feature points. Finally, we obtained about 140,000 local image patches. Fig. 5 illustrates the local patch construction procedure. We then train the binary GAN model, described in Section 2.3 by the extracted local image patches.

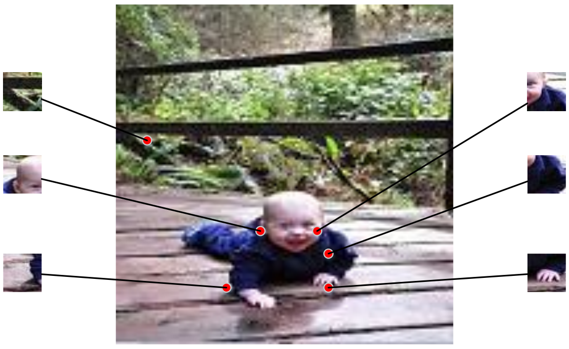

Fig. 5. Local Image Patch Construction.

In this paper, we used a large-scale place-oriented dataset (Zhou et al. (2018)) to perform the unsupervised training. This dataset is unrelated to any scene that the loop closure detection may encounter and the time-consuming process of label annotations is avoided.

3. EXPERIMENT

3.1 Evaluation Dataset

Three datasets are used to evaluate the performance of the proposed approach, i.e. New College (NC) dataset and City Center (CC) dataset (Cummins and Newman (2008)), and Korea Advanced Institute Science and Technology (KAIST) dataset (Choi et al. (2016)). NC dataset contains 2146 images and CC dataset contains 2474 images. These datasets were collected by placing a camera on the left and right sides of the mobile platform and acquiring an image every 1.5m. These datasets include dynamic objects, and in addition, they were collected on sunny and windy days, which makes the features of images with leaves and shadows unstable. KAIST dataset contains three different sequences (North/West/East). Each sequence contains 200 images. Images in the KAIST dataset were collected while driving a car along the same route at different times of the day. Details of the datasets are summarized in Table 2. Strong variations about viewpoint and condition changes can be found in the datasets. They could be great challenges for the hand-crafted feature based method to handle these strong variations, and they could be good examples to demonstrate the efficiency of the proposed approach.

Table 2. Dataset Descriptions.

Dataset	Environment	Viewpoint Variation	Condition Variation
NC	Campus	strong	moderate
CC	Downtown	strong	strong
KAIST	Driving	moderate	strong

3.2 Experimental Results

In this section, we compare our approach with several mainstream methods, including the BoVW with hand-crafted features like ORB, BRIEF and SURF, and some deep learning based methods such as AlexNet, AmosNet and HybridNet. In order to evaluate each method, we plot the Precision-Recall (PR) curves (Zaffar et al. (2019)) and calculate the area under the precision-recall curves (AUC) (Zaffar et al. (2019)) by the following formula:

$$AUC = \sum_{i=1}^{M-1}\frac{(p_i + p_{i+1})}{2}\times(r_{i+1}-r_i) \tag{9}$$

where M represents the number of image sequences, p_i represents the precision rate at point i and r_i represents the recall rate at point i.

The detailed results are shown from Fig. 6 to Fig. 10. The AUC score for each method on three datasets are compared in Table 3. It is found that

(1) The performance improvements are obvious by using a deep learning based method no matter it is supervised (AlexNet, AmosNet and HybirdNet) or unsupervised (ours). The reason is that deep learning can automatically extract features and better reflect the essence than those hand-crafted features in complex scenes.

(2) Despite the best results on the NC dataset, the proposed unsupervised method is found somehow less effective than those supervised methods. This is fair because supervised learning is more targeted but takes more time for training. The performance gap is relatively small and completely acceptable. Therefore, it is possible to use unsupervised method for loop closure detection, and we can benefit a lot since the tedious task of data annotation can be totally avoided. In the experiment, our method requires only 2,000 unlabelled images.

(3) From all the figures, it is interesting to find that binary feature descriptors are superior to the nonbinary descriptors. This is of special significance for loop closure detection problem because the binary features are more practical in an embedded vSLAM system.

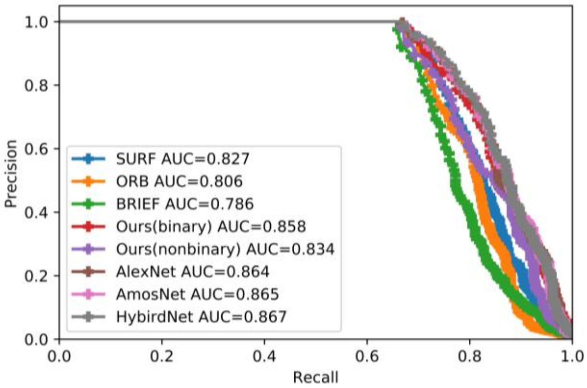

Fig. 6. AUC under PR curves on the CC dataset.

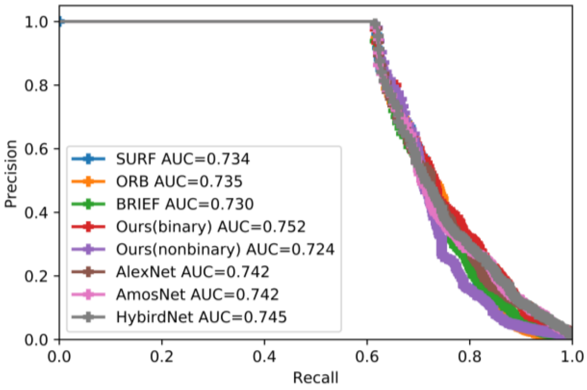

Fig. 6. AUC under PR curves on the NC dataset.

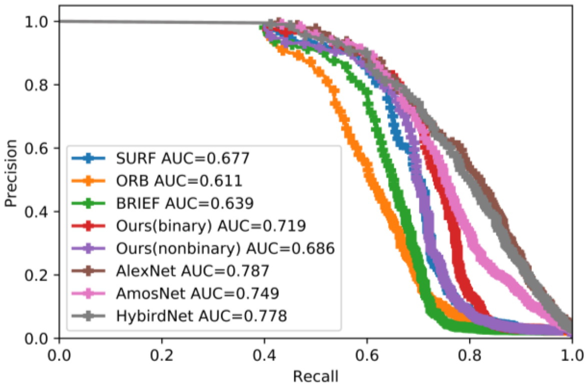

Fig. 7. AUC under PR curves on the KAIST(East) dataset.

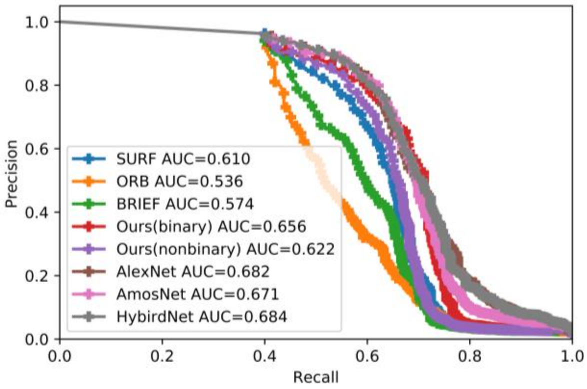

Fig. 8. AUC under PR curves on the KAIST(North) dataset.

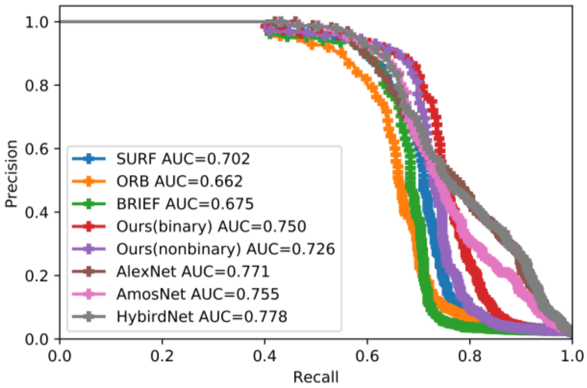

Fig. 9. AUC under PR curves on the KAIST(West) dataset.

Table 3 AUC results.

Methods	Datasets				
	CC	NC	North	East	West
SURF	0.827	0.734	0.610	0.677	0.702
ORB	0.806	0.735	0.536	0.611	0.662
BRIEF	0.786	0.730	0.574	0.639	0.675
Ours(binary)	0.858	0.752	0.656	0.719	0.750
Ours (nonbinary)	0.834	0.724	0.622	0.686	0.726
AlexNet	0.864	0.742	0.682	0.787	0.771
AmosNet	0.865	0.742	0.671	0.749	0.755
HybridNet	0.867	0.745	0.684	0.778	0.778

Furthermore, the results of the k-means clustering on local image patches from BRIEF, SURF, ORB and the proposed approach are studied and visualized through Fig. 11 to Fig. 14. Due to space limitations, we only selected the first 15 images of the CC dataset and visualize the clustered results of the class 1. The number of cluster center is set as 20. As we can find from the results, the local descriptors such as BRIEF, SURF, and ORB badly gathered the patches with different appearances. It is shown that the proposed approach well aggregated local patches with similar appearances. The overall effect is much better than the other three methods.

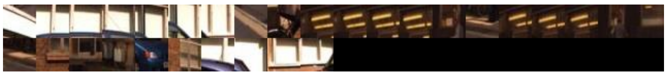

Fig. 11. Clustering results on BRIEF descriptor.

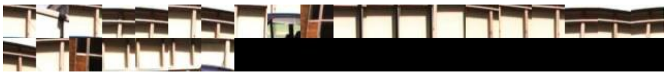

Fig. 12. Clustering results on SURF descriptor.

Fig. 13. Clustering results on ORB descriptor.

Fig. 14. Clustering results of the proposed approach.

Moreover, we examine the influence of the hyperparameters in the loss function on the performance. Table 4 shows the results. It is obvious that both DP loss function and BRE loss function play active roles on the performance improvement. We can therefore conclude that the performance of the proposed binary GAN can be further improved if we carefully tune the hyperparameter λ_{DP} and λ_{BRE}.

Table 4 AUC results with different hyperparameters.

Parameters	Datasets				
	CC	NC	North	East	West
$\lambda_{DP}=0.5\,, \lambda_{BRE}=0.1$	0.858	0.752	0.656	0.719	0.750
$\lambda_{DP}=0, \lambda_{BRE}=0.1$	0.752	0.690	0.511	0.522	0.574
$\lambda_{DP}=0.5, \lambda_{BRE}=0$	0.768	0.692	0.493	0.512	0.541

4. CONCLUSIONS

In this paper, we propose a novel loop closure detection approach by incorporating a dedicated binary GAN into the BoVW model. This binary GAN model is trained in an unsupervised manner without any image annotations. In order to extract more discriminative low-dimensional binary descriptors, we use the DP loss function to propagate the Hamming distance from the high-dimensional feature space to the low-dimensional feature space and use the BRE loss function to promote the diversity. We compared our proposed method extensively with other state-of-the-art methods. Powered by the binary feature descriptors, it is verified that unsupervised loop closure detection is possible for vSLAM even concerning limited computing resource in autonomous unmanned systems.

REFERENCES

Bay H., Tuytelaars T., and Gool L.V. (2006). SURF: Speeded up robust features, In *European conference on Computer Vision*. Springer.

Cao Y., Ding G. W., Lui K. Y.-C. and Huang R. (2018). Improving GAN training via binarized representation entropy (BRE) regularization, In *International Conference on Learning Representations*.

Chen Z., Jacobson A., Sünderhauf N., Upcroft B., Liu L., Shen C., Reid I. and Milford M. (2017). Deep learning features at scale for visual place recognition, In *International Conference on Robotics & Automation*, 3223-3230. IEEE.

Gao X., and Zhang T. (2017). Unsupervised learning to detect loops using deep neural networks for visual SLAM system, *Autonomous Robots*, Vol.41 (1), pp. 1-18.

Diederik P. K., and Jimmy B. (2014). Adam: A method for stochastic optimization, arXiv preprint arXiv: 1412.6980.

Dong H. W., and Yang Y. H. (2018). Training Generative Adversarial Networks with Binary Neurons by End-to-end Backpropagation, arXiv preprint arXiv: 1810.04714, 2018.

Hou Y., Zhang H., and Zhou S. (2015). Convolutional Neural Network-Based Image Representation for Visual Loop Closure Detection, In *International Conference on Information and Automation*. IEEE.

Kummerle R., Grisetti G., Strasdat H., Konolige K. and Burgard W. (2013). g2o: A General Framework for Graph Optimization. In *International Conference on Robotics and Automation*, 3607-3613. IEEE.

Lin M., Chen Q., and Yan S. (2013). Network in network, arXiv preprint arXiv:1312.4400, 2013.

Lowe D.G. (2004). Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, Vol.60 (2), pp. 91-110.

Olid D., Fácil J. M., and Civera J. (2018). Single-View Place Recognition under Seasonal Changes, In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Radford A., Metz L. and Chintala S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks, In *International Conference on Learning Representations*.

Rublee E., Rabaud V., Konolige K. and Bradski G. (2011). ORB: An efficient alternative to SIFT or SURF, In *International Conference on Computer Vision*. IEEE.

Salimans T., Goodfellow I., Zaremba W., Cheung V., Radford A., Chen X. (2016). Improved techniques for training gans, In *International Conference on Neural Information Processing Systems*.

Shin D. W, Ho W. S. and Kim E. S. (2019). Loop closure detection in simultaneous localization and mapping using descriptor from generative adversarial network, Journal of Electronic Imaging, Vol.28 (1), 013014.

Sünderhauf N., Shirazi S., Dayoub F., Upcroft B., and Milford M. (2015). On the performance of ConvNet features for place recognition, In *International Conference on Intelligent Robots and Systems*. IEEE.

Zaffar M., Khaliq A., Ehsan S., Milford M., and McDonald-Maier K. (2019). Levelling the Playing Field: A Comprehensive Comparison of Visual Place Recognition Approaches under Changing Conditions. In *International Conference on Robotics and Automation*. IEEE.

Zhou B., Lapedriza A., Khosla A., Oliva A., and Torralba A. (2018). Places: A 10 Million Image Database for Scene Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.40 (6), pp. 1452-1464.