Improving Loop Closure Validation in Challenging Scenes by Aggregating Sequential Context Information

Zijun Lin*
Department of Electronic and Electrical Engineering
Southern University of Science and Technology
Shenzhen, China
12232147@mail.sustech.edu.cn

Yijun Zhao*
Department of Biomedical Engineering
Southern University of Science and Technology
Shenzhen, China
12232558@mail.sustech.edu.cn

***Abstract*—We concentrate on the performance of loop closure validation in dynamic and changing situations in the specific context of our research. We employ the temporal and spatial interactions between images in a sequential context information approach based on learning. The baseline method extracts and matches image features using the ORB descriptors, brute force matching, and epipolar constraints. The proposed approach has the capability to more accurately predict the probability of a valid match in complex and unpredictable environments, as shown by a higher recall rate at 100% precision. Project link: github link**

***Index Terms*—Loop closure validation, learned features, sequential context information**

I. Introduction

Visual Place Recognition (VPR) and Loop Closure Detection (LCD) have played an increasingly important role in Computer Vision and in the field of Robotics, as autonomous driving, robotics navigation, as well as augmented reality, is developing rapidly. LCD enables robots to recognize previously visited locations, allowing them to correct accumulated errors in their internal maps.

Loop closure validation, a key technique employed in Simultaneous Localization and Mapping (SLAM), is to verify whether the mobile robot has already been to the previously detected areas. In environments that experience minimal change, loop closure validation can be successfully implemented through the use of classical feature encoding. However, in a majority of real-world situations, the appearance of a scene may undergo significant alterations due to various factors, such as the time of day, seasonal changes, weather conditions, or even human intervention.

These considerable fluctuations present a substantial challenge to traditional loop closure detection methodologies that depend on static features. Conventional hand-crafted feature extraction techniques used in Loop Closure Validation (LCV) as well as place recognition are highly susceptible to environmental changes. To address these difficulties arising from scene variations, learning-based feature extraction methods have been proposed. When compared to traditional methods, these learning-based feature extraction techniques exhibit greater robustness in LCV.

Besides the learning-based feature extraction approaches, incorporating sequential information is another potential strategy for enhancing loop closure validation performance. Sequential information, which refers to the temporal ordering and dependencies between observations, has the potential to improve the reliability of loop closure validation by exploiting the inherent structure of the environment and reducing the impact of perceptual aliasing. By incorporating sequential information into the validation process, it is possible to achieve more accurate and reliable loop closure detection, ultimately contributing to an overall improvement in SLAM performance, as shown in Fig.1.

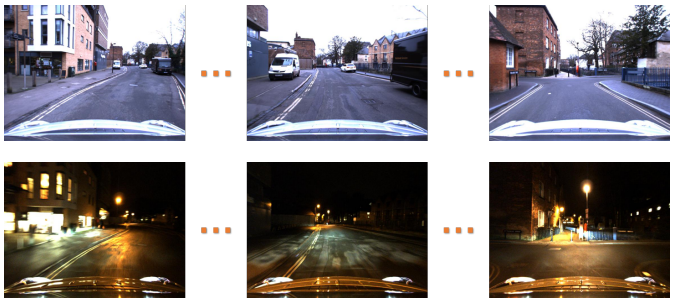


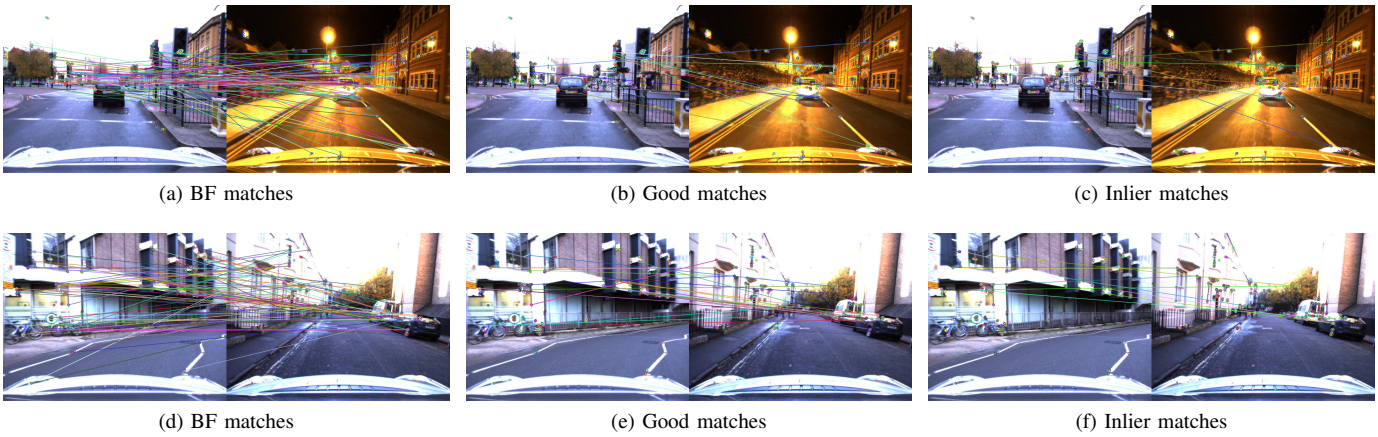
Fig. 1: **Image sequences provide rich information.** Compared to single-image matching, more visual aids are included, and it is possible to achieve more reliable loop closure recognition.

II. Related Work

A. Feature Extraction and Matching

Traditional feature extraction and matching are generally performed by i) detecting interest points, ii) computing visual descriptors, iii) matching these with a Nearest Neighbor (NN) search, iv) filtering incorrect matches, and finally v) estimating a geometric transformation.

* equal contribution



(a) BF matches (b) Good matches (c) Inlier matches

(d) BF matches (e) Good matches (f) Inlier matches

Fig. 2: **Feature extraction and matching.** a) to c) are matching results on the Autumn-Night dataset. d) to f) are matching results on Autumn-Suncloud dataset.

1) Feature Extraction: The field of feature extraction and matching traditionally relies on methods rooted in the use of detectors and descriptors. Several key algorithms such as Scale-Invariant Feature Transform (SIFT), Binary Robust Independent Elementary Features (BRIEF), Features from Accelerated Segment Test (FAST), and Oriented FAST and Rotated BRIEF (ORB) have established themselves as both detectors and descriptors. FAST and its derivatives, as cited in [1], [2], are the preferred choice for identifying keypoints in real-time systems that require visual feature matching, such as Parallel Tracking and Mapping (PTAM). The efficacy of these algorithms lies in their efficiency in locating corner keypoints that are significant and appropriate for the task at hand.

On the other hand, BRIEF [3] is a relatively newer feature descriptor that conducts simple binary tests between pixels within a smoothed image patch. It boasts performance metrics comparable to SIFT in multiple dimensions, exhibiting robustness against changes in lighting, blur, and perspective distortion. However, BRIEF is recognized to be highly susceptible to in-plane rotation.

Combining the strengths of the FAST keypoint detector and the BRIEF descriptor is the ORB [4]. This method offers a computationally-efficient alternative to SIFT, displaying similar matching performance. Additionally, ORB exhibits greater resistance to image noise and can be effectively used for real-time performance.

2) Feature Matching: Feature matching comprises the computation of distances and identification of matching pairs. Brute force matching, a basic method, finds the keypoints with the smallest distance in a secondary image as a pair. A more sophisticated approach involves using Fast Library for Approximate Nearest Neighbors (FLANN) [5], which builds a KD tree to identify matching pairs. The advantage of FLANN is that it circumvents the need for exhaustive matching, thus optimizing the process.

B. VPR/LCD Image Retrieval with Sequence

There is little work on VPR/LCD validation, however, the effectiveness of sequential information has been proven in image retrieval tasks. The use of sequential information is known to improve robustness against perceptual aliasing as more visual evidence is included in consideration and thus prevents false matches caused by limited information or occasional transient noise within single images. SeqNet [6] aggregates single image descriptors with 1D convolution to form a summary vector and then uses it to extract candidates at a coarse level. Fine-level scores are then calculated using single image descriptors in the query sequence and candidates. In contrast to SeqNet [6] which focuses on sequence description, SeqMatchNet [7] does not perform any descriptor aggregation but learns a linear transform of single image descriptors, and these single image descriptors are then matched in an order-preserving manner.

III. VPR/LCD Validation

A. Problem Formulation

We formulate the LCD/VPR validation task before explaining the proposed method.

The objective of validation task. The objective of a validation task is to judge whether two images are taken in the same place given two query images as inputs. This is slightly different from the image retrieval tasks, which require a fast extraction of candidates from a database given a query.

The essential feature of a validation algorithm. The essential feature of a validation algorithm is that 100% precision can be reached by adjusting some threshold. The first key point is 100% precision. False Positive (FP) samples are detrimental to the successfulness of robot navigation tasks, thus we must prevent the occurrence of FPs. The second key point is the threshold. All the previous validation methods reach 100% precision by adjusting some threshold. For example, in the validation methods that utilize feature matching and geometry information, we can not ensure all the matches are correct,

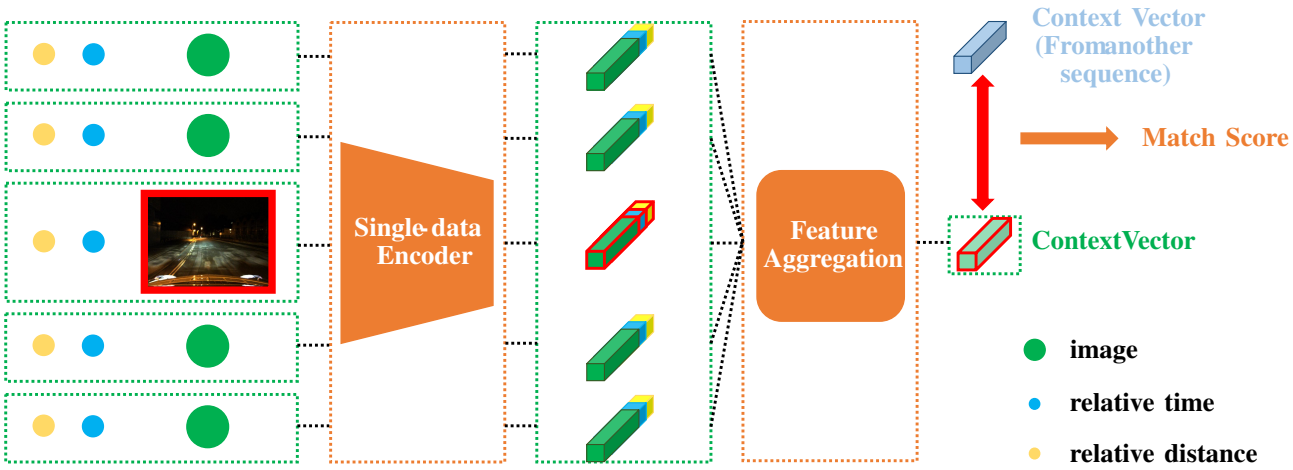


Fig. 3: **Overview of our proposed framework.** The proposed method consists of two main modules: the Single-data Encoder and the Feature Aggregation module. The Single-data Encoder first encode the image, relative time, and relative distance separately and then concatenates them into a fused vector. The fused vectors from different data in a sequence are then used to construct a graph and feed into the Feature Aggregation Module to get a context vector. Finally, the context vectors from two query sequences are used to compute a match score and we can get a binary result by an adjustable threshold.

and the existence of wrong matches can affect the results. Moreover, even for the correct matches, reproject errors can not be eliminated. Thus, we still need to adjust some thresholds according to some specific datasets to reach an "empirically" 100% perception. These are the unchanged rules for all the validation algorithms due to the essential requirement of the validation tasks.

The specific settings of our proposed method Since the scenario of this project is robot navigation, the proposed system can also have access to two image sequences which are composed of images that are adjacent to the query images. Relative timestamps naturally exist in these sequences, and relative distance can be easily calculated by off-the-shelf visual odometry algorithms.

B. VPR/LCD Validation Baseline Implementation

In our project, we've put together a three-step process to detect and validate loop closure between two images.

- **Step 1**: We use a method known as Oriented FAST and Rotated BRIEF (ORB) to extract the unique features of each image. ORB is known for being both strong and efficient, capable of quickly finding key details in an image that can be used to identify it later.
- **Step 2**: We use a technique called Brute Force Matching to find similarities between the features of two images. This matcher looks at each feature in the first image and finds the most similar feature in the second image. The matches are then sorted based on their differences, which are measured using something called the Hamming distance.
- **Step 3**: We check if the matched features between the two images satisfy certain geometric constraints, also known as epipolar geometry. We use a method called RANSAC to estimate the Fundamental Matrix, a representation of this geometry. If the number of matches that satisfy these constraints is above a certain threshold, we conclude that the two images are of the same location, indicating a loop closure.

In summary, the baseline method combines the ORB feature extraction method, Brute Force Matching, and the enforcement of epipolar geometry constraints to effectively detect and validate loop closure in a Simultaneous Localization and Mapping (SLAM) system. The matched points obtained through the steps above are shown in Fig. 2

C. VPR/LCD Validation with Image Sequence

1) Method Overview: The Fig.3 shows our proposed framework. The proposed framework consists of two main modules: the Single-data Encoder and the Feature Aggregation module. The Single-data Encoder first encode the image, relative time, and relative distance separately and then concatenates them into a fused vector. The fused vectors from different data in a sequence are then used to construct a graph and feed into the Feature Aggregation Module to get a context vector. Finally, the context vectors from two query sequences are used to compute a match score and we can get a binary result by an adjustable threshold.

2) Model Pipeline Details.: The details of the Single-data Encoder and the Aggregation Module are shown in Fig.4.

Single-data Encoder. The Single-data Encoder is comprised of three parts: the image encoder, the timestamp encoder, and the relative distance encoder. The image encoder is constructed on a frozen EfficientNetv2 [8]. We cropped the EfficientNetv2 [8] backbone from the layers before the last average pooling layer and replaced it with a Backbone Output Head, which is comprised of two convolution layers each followed by a Batch Normalization as normalization and a Leaky ReLU activation layer.

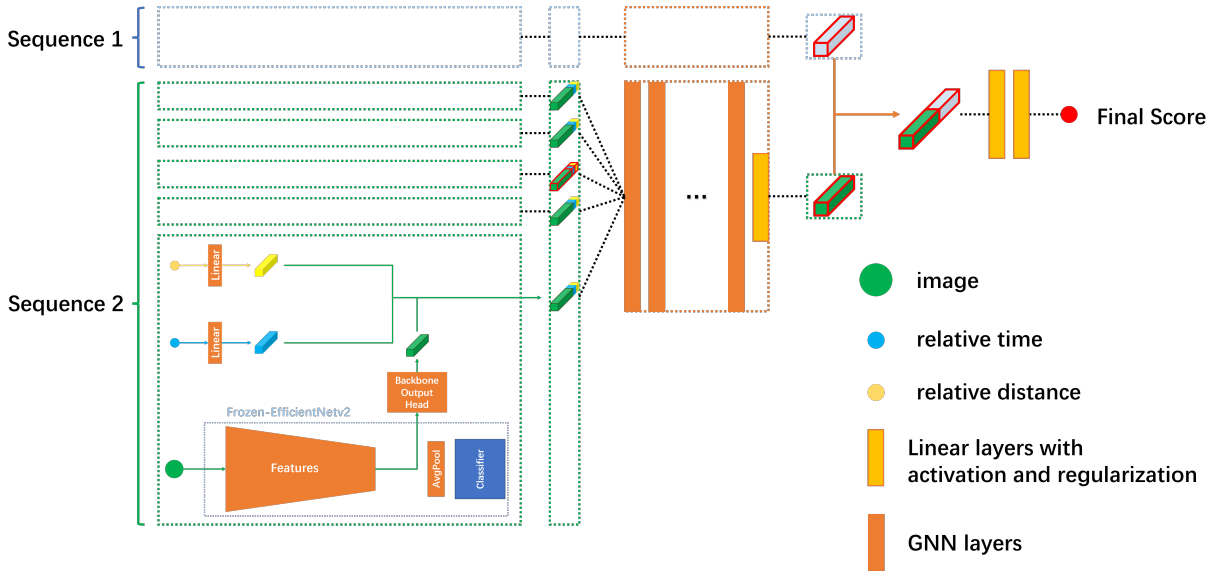


Fig. 4: **Details of the model pipeline.** Each data in a sequence are processed independently by the Single-data encoder to get a fused encoding. Subsequently, all the encodings in a sequence are aggregated by GNN layers to extract a context vector. We finally concatenate two context vectors and get a match score from the Output Head.

Feature Aggregation. After the fused vectors of each data are extracted, feature aggregation is conducted by the GNN layers and the following linear layers. Specifically, we have tried GCN [9], GATv2 [10], and GIN [11] layers for the GNN variant, and we finally adopt GCN [9]. Readers can refer to the experiment chapter for experiment results of different GNN variants. The number of layers is set to be an experiment variable. Note that the GNN layers are flexible enough to adopt graph inputs with any node length. Context vectors are learned from a linear transform of the query node information of each sequence and feed into the Output Head after concatenation. The Output Head is constructed by simply stacking linear layers with activation and regularization. We use Leaky ReLU as our activation function for all layers, and the Dropout as the regularization layer.

3) Dynamic Sequence Selection and Graph Construction: The process of dynamic sequence selection and graph construction have not been shown in the model pipeline, thus, this chapter will present a detailed illustration of how to select a sequence and construct a graph.

Dynamic Sequence Selection. Given two input queries, we need to find a proper sequence for each of the target images. Fig.5 show the process of dynamic sequence selection. Cumulative distances relative to the query are first calculated from visual odometry, thus candidate sequence images can be selected according to a threshold interval. We set another threshold to control the total length of the sequence. We repeat the process twice to find the bidirectional (left and right) adjacent sequence for the query images.

Graph Construction. Fig.6 shows the processs of graph construction. To preserve the temporal and ordering relationship between images in a sequence, we first connect each node to its adjacent nodes. Then, we connect each node to the query node to complete the graph since we expect to aggregate information more efficiently and with fewer GNN layers.

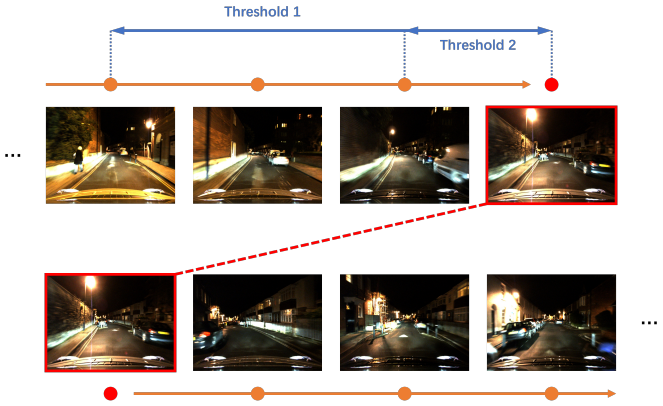


Fig. 5: **Dynamic sequence selection.** The red block indicates the query node. Left and right sequences are selected according to thresholds.

IV. Experiments

A. Experimental Setting

In this project, our dataset consists of two parts: The datasets given by the TA in EE5346 and the dataset that we made ourselves. All of these datasets come from Oxford RobotCar Dataset [12]

- The Autumn-Night and Autumn-Suncloud datasets, which were provided by TA in EE5346, were collected at 2014-12-16-18-44-24 (Night), 2014-12-09-13-21-02 (Autumn), and 2014-11-18-13-20-12 (Suncloud). There are 800 pairs in each of these two ground-truth tables.

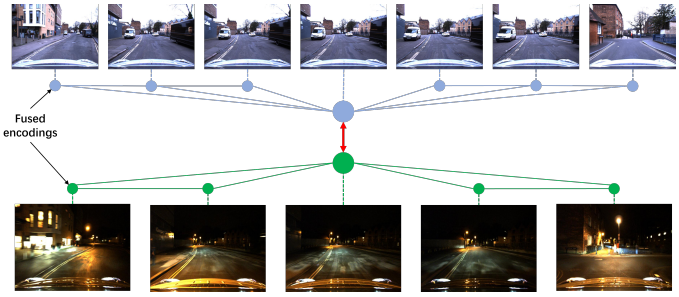


Fig. 6: **Graph Construction.** After we have got the fused encodings from the encoder, we form a graph centered at the query node for each sequence.

- We created the data set, which is based on the dates 2014-12-10-18 10-50 (Night) and 2015-03-17 11-08-44 (Day). We create a ground-truth table with 2000 pairs of data using the GPS data.

We divide all the data into the ratio train: val: test = 8:1:1 to make our training process for the suggested approach easier.

B. Evaluation Metrics

1) PR curve: A PR curve plots the Precision (y-axis) against the Recall (x-axis) for every possible cut-off. This curve provides a comprehensive view of the trade-off between Precision and Recall for different threshold settings and thus, assists in selecting the optimal threshold based on the problem's requirements. Using the PR curve for evaluation allows us to balance these two crucial aspects of an algorithm – Precision and Recall – ensuring we do not over-emphasize one at the cost of the other. It also assists in visually and quantitatively comparing the performance of different models.

2) Recall at 100% precision: In our evaluation metrics, an important criterion we enforce is ensuring a 100% precision rate for our SLAM system. The precision is a critical measure as it quantifies the system's ability to correctly identify true positives, i.e., correctly detecting loop closures. For our SLAM application, it is vital to maintain high precision because a false positive, detecting a loop closure when there isn't one, could have serious consequences including incorrect map construction and inaccurate localization. However, maintaining high precision, especially 100%, should not come at the cost of the recall, another crucial metric in our context. Recall, also known as sensitivity or true positive rate, measures the proportion of actual positives (loop closures) that are correctly identified. In other words, it evaluates the system's ability to detect and correctly identify as many loop closures as possible. Our evaluation, therefore, focuses on achieving the highest possible recall while maintaining a precision of 100%. This balances the two aspects: ensuring we correctly identify all loop closures (high precision), and detecting as many of these instances as possible (high recall).

C. Quantitative Results

1) PR curve:

- **The Autumn-Night dataset.** The outcomes of the baseline technique and the proposed method on the Autumn-Night dataset with 10% data during testing are shown in Fig. 7. The proposed strategy can achieve 100% precision on the Autumn-Night dataset while still keeping a good recall value. When the recall varies from 0 to roughly 0.1, the precision stays at 100%. When the recall is approximately 0.35, the precision then reaches its lowest point (close to 0.6). Later, when recall varies between 0.35 and 1.0, precision swings around 0.7. No matter how the recall value changes, the baseline technique can never achieve a high precision value. When the recall goes from 0.4 to 1.0, the precision value first rises as the recall goes up, then it stays at 0.5.
- **The Autumn-Suncloud dataset.** Fig. 8 displays the results of the baseline technique and the suggested method on the dataset for Autumn-Suncloud with 10% of the data during testing. Similar patterns can be seen in both curves, with precision remaining constant at 1 and then decreasing as recall rises. More particular, when the recall reaches 0.25 for the baseline technique, the precision begins to decline from 1. When recall is between 0.8 and 1, the precision value is roughly 0.5 at the lowest level. Comparatively, when the recall is 0.4, the proposed technique drifts from 1. By the time recall reaches its most significant value, the lowest precision value is around 0.8.

The comparison of PR curves between the two methods shows that the proposed method can better balance precision and recall since the precision of the proposed method can always reach 1 under some conditions.

2) The influence of thresholds on precision and recall:

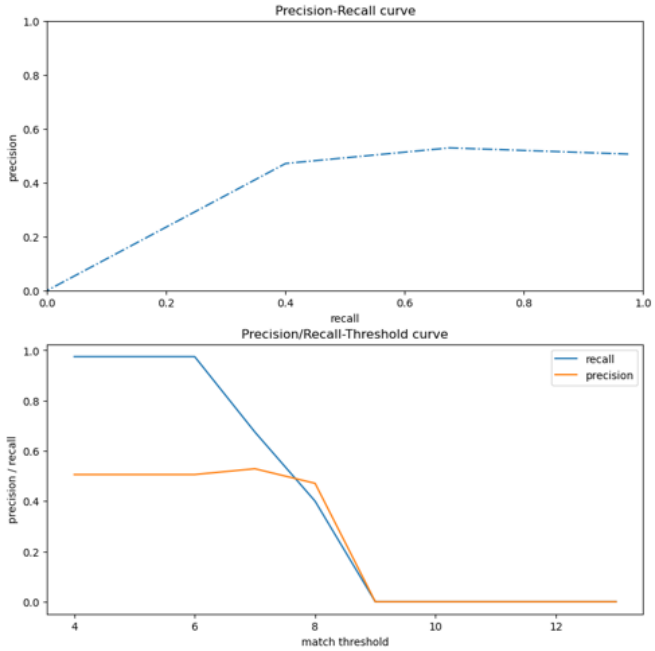
- **The Autumn-Night dataset.** Precision and recall for the baseline technique first stay at their highest value for a while before declining to their lowest value. The precision ranges from 0 to 0.5, whereas the recall ranges from 0 to 1. For the suggested method, recall can likewise vary between 0 and 1, whereas the smallest precision value is close to 0.6 and the maximum can reach 1.
- **The Autumn-Suncloud dataset.** The two approaches' precision and recall exhibit the same trend. While recall decreases as the threshold value rises, precision rises as it does.

D. Recall at 100% precision

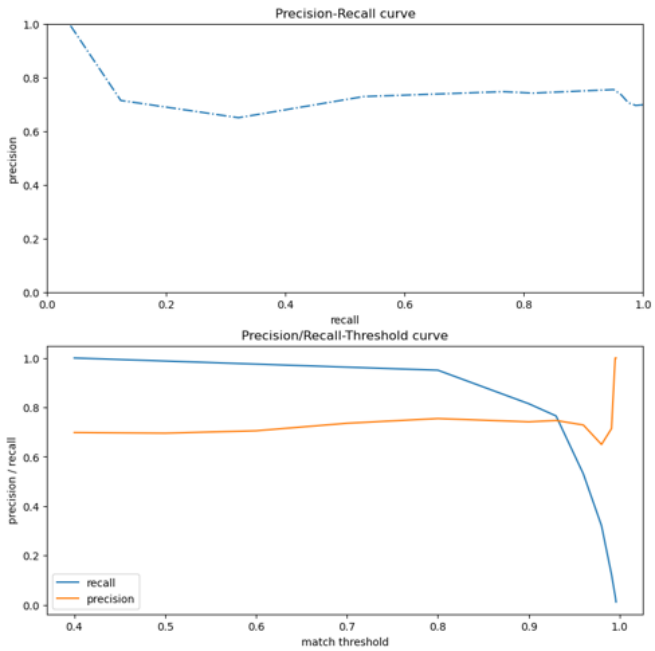
The recall value at 100% precision for both methods is summarized in the TABLE I. The proposed method can either reach 100% precision or reach 100% precision at a higher recall value compared to the baseline method, which shows its better performance.

V. Current Limits and Discussions

The most significant problem is the diversity of training data. We build the whole model pipeline from scratch (except the image feature backbone), and it needs to be trained on a large-scale dataset for generalizability. At the beginning of

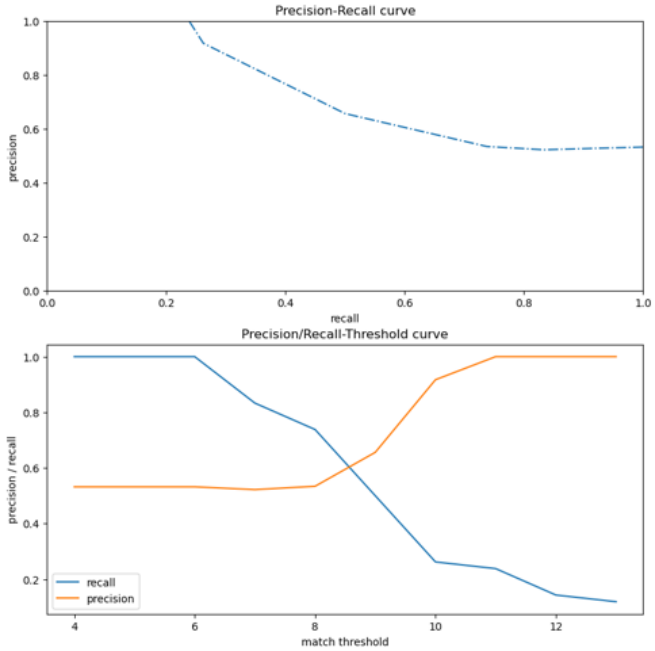


(a) Result of Baseline

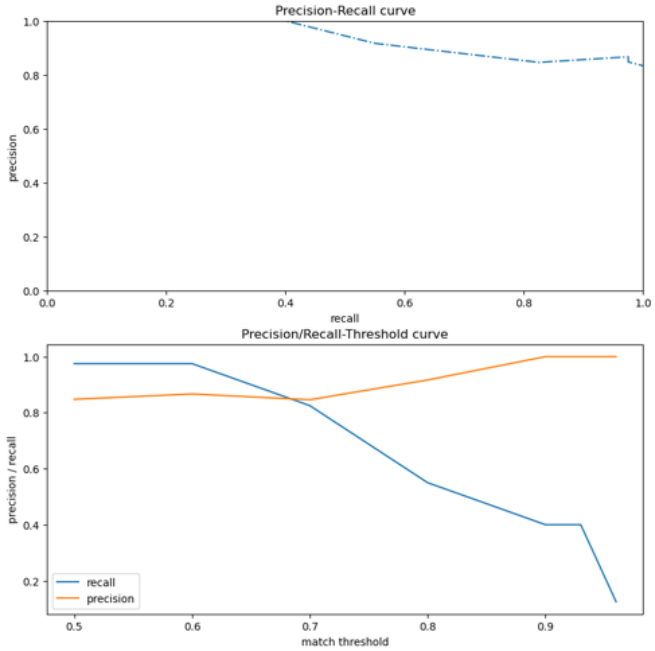


(b) Result of Proposed Method

Fig. 7: Results on the Autumn-Night test dataset. (a) shows the result of the baseline method, and (b) shows the proposed method result.



(a) Result of Baseline



(b) Result of Proposed Method

Fig. 8: Results on the Autumn-Suncloud test dataset. (a) shows the result of the baseline method, and (b) shows the proposed method result.

TABLE I: Recall at 100% precision.

datasets	Baseline method	Proposed method
Autumn-Night	-	0.037
Autumn-Suncloud	0.25	0.4

this project, we first tried to use the dataset provided by the course TAs: Autumn-Night and Autumn-Suncloud. However, we soon discover that the amount of data is far from enough, and there are overlaps between sequences, which may cause overfitting. We then make our own dataset based on the Oxford RobotCar sequence 2015-03-17-11-08-44 and 2014-12-10-18-10-50 match pairs. However, problems still exist. The amount of "efficient" data is still too small, and if we take too many samples from the same travel, there will be large overlaps between sampled sequences. Thus, the diversity of the "efficient" dataset is limited, which causes suboptimal performance while testing the model on the Autumn-Night dataset.

To meet the requirement of the IID (Independent Identical Distribution) assumption, we should adopt more travels to build our dataset. Unfortunately, we have no spare time to extend our dataset to train a better model since the project deadline is up.

VI. Conclusion

In our research, we focused on improving loop closure validation under dynamic and changing conditions. We applied an approach that leverages learning-based methods, considering both temporal and spatial interactions between sequential images. Compared to the baseline method utilizes ORB descriptors, brute force matching, and enforces epipolar constraints to extract and match image features, our proposed approach demonstrates enhanced proficiency in accurately predicting valid matches in complex environments, as evidenced by an increased recall rate at 100% precision. These findings highlight the potential of our proposed approach to improve the reliability of Simultaneous Localization and Mapping (SLAM) systems under dynamic and changing conditions, especially during day and night.

References

[1] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pp. 430–443, Springer, 2006.

[2] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 105–119, 2008.

[3] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pp. 778–792, Springer, 2010.

[4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*, pp. 2564–2571, Ieee, 2011.

[5] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration.," *VISAPP (1)*, vol. 2, no. 331-340, p. 2, 2009.

[6] S. Garg and M. Milford, "Seqnet: Learning descriptors for sequence-based hierarchical place recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4305–4312, 2021.

[7] S. Garg, M. Vankadari, and M. Milford, "Seqmatchnet: Contrastive learning with sequence matching for place recognition & relocalization," in *Conference on Robot Learning*, pp. 429–443, PMLR, 2022.

[8] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*, pp. 10096–10106, PMLR, 2021.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[10] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?," *arXiv preprint arXiv:2105.14491*, 2021.

[11] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018.

[12] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.