



Article

# Loop Closure Detection with CNN in RGB-D SLAM for Intelligent Agricultural Equipment

Haixia Qi <sup>1,2,\*</sup> , Chaohai Wang <sup>1</sup>, Jianwen Li <sup>1</sup> and Linlin Shi <sup>1</sup> 

<sup>1</sup> College of Engineering, South China Agricultural University, Guangzhou 510642, China; charwang@stu.scau.edu.cn (C.W.); ljw1@stu.scau.edu (J.L.); lynnshi@scau.edu.cn (L.S.)

<sup>2</sup> Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou 510642, China

\* Correspondence: qihaixia@scau.edu.cn

**Abstract:** Loop closure detection plays an important role in the construction of reliable maps for intelligent agricultural machinery equipment. With the combination of convolutional neural networks (CNN), its accuracy and real-time performance are better than those based on traditional manual features. However, due to the use of small embedded devices in agricultural machinery and the need to handle multiple tasks simultaneously, achieving optimal response speeds becomes challenging, especially when operating on large networks. This emphasizes the need to study in depth the kind of lightweight CNN loop closure detection algorithm more suitable for intelligent agricultural machinery. This paper compares a variety of loop closure detection based on lightweight CNN features. Specifically, we prove that GhostNet with feature reuse can extract image features with both high-dimensional semantic information and low-dimensional geometric information, which can significantly improve the loop closure detection accuracy and real-time performance. To further enhance the speed of detection, we implement Multi-Probe Random Hyperplane Local Sensitive Hashing (LSH) algorithms. We evaluate our approach using both a public dataset and a proprietary greenhouse dataset, employing an incremental data processing method. The results demonstrate that GhostNet and the Linear Scanning Multi-Probe LSH algorithm synergize to meet the precision and real-time requirements of agricultural closed-loop detection.

**Keywords:** intelligent agricultural equipment; RGB-D SLAM; loop closure detection; lightweight convolutional neural networks; multi-probe random-hyperplane locality-sensitive hashing



**Citation:** Qi, H.; Wang, C.; Li, J.; Shi, L. Loop Closure Detection with CNN in RGB-D SLAM for Intelligent Agricultural Equipment. *Agriculture* **2024**, *14*, 949. <https://doi.org/10.3390/agriculture14060949>

Academic Editor: Francesco Marinello

Received: 13 April 2024

Revised: 4 June 2024

Accepted: 11 June 2024

Published: 18 June 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In autonomous robotic systems, simultaneous localization and mapping (SLAM) has been a focal point of research for decades [1–3]. Its primary aim is to map unknown environments while concurrently localizing the robot within them, which is a critical function for agricultural robots performing tasks like navigation [4,5], path planning [6], and manipulation [7,8]. The classic SLAM process comprises four primary tasks: visual odometry, optimization, loop closure detection and mapping [9,10]. Loop closure detection serves the function of recognizing previously visited locations. If a loop closure has occurred, adjustments can be made to the robot's estimated trajectory based on the error between the current estimated map point and the same map point that was last visited. These adjustments correct for inaccuracies stemming from imprecise sensor measurements, uncertain environmental conditions, and errors in odometry estimation. Therefore, loop closure detection is crucial for correcting errors and optimizing the local map [11–13]. Common loop closure detection is divided into two steps: feature extraction and feature matching. Feature extraction is to extract the feature information of the current image, and feature matching is to use the feature information extracted in the previous step to match the previously obtained image features to determine whether the current location has ever been visited. However, the traditional manual features in feature extraction are

not effective in greenhouse scenes, so the paper uses convolutional neural networks (CNN) to extract image features. However, because the image features extracted by CNN are high-dimensional information, it is difficult to use the bag of words matching method, so the hash algorithm is proposed for feature matching.

With advancements in neural networks, researchers have noticed parallels between loop closure detection in visual SLAM and image recognition and classification problems addressed by neural networks. Both tasks boil down to addressing the challenge of associating image data accurately [14]. Chen et al. introduced a loop closure detection algorithm based on CNN and spatial-sequential filters, which improved the recall rate by 75% in the dataset [15]. It has been demonstrated that image features extracted from the neural network outperform manually designed ones, thereby enhancing the accuracy of loop closure detection.

### 1.1. Review

The current research on image appearance can be categorized into two main directions: unsupervised learning of image features using self-encoder, and extraction of image features through off-the-shelf CNN for loop closure detection. Gao et al. proposed a stacked denoising auto-encoder model for unsupervised learning, achieving satisfactory precision. However, the detection time of all frames will be about 2.2 s, which is not suitable for real-time loop closure detection [16]. Jia Xuewei introduced PCANet-LDA, combining unsupervised neural networks and linear discriminant analysis, and achieved a 60.2% reduction in time cost compared with the GoodLeNet network by extracting features based on network class differentiation [17].

Some approaches to loop closure detection rely on CNN. Hou et al. demonstrated that AlexNet extracts descriptors three times faster than traditional manual descriptors like SIFT features and Gist descriptors under significant illumination changes [18]. Xia et al. proposed using the cascaded neural network model PCANet for better loop closure detection. It takes more than 19% less time than SIFT features and Gist and guarantees a minimum average precision of around 75% [19]. They also found that AlexNet features trained twice using SVM yield optimal results in loop closure detection experiments, showing more robustness than manually designed features [20]. Retraining the network model is a promising approach to improve loop closure detection accuracy. Wang used PCA for feature compression and sparse constraints on feature vectors before comparing similarities, achieving that by expressing the features of an image with a 500-dimensional vector [21]. Lopez-Antequera observed enhanced loop closure detection accuracy by retraining AlexNet with the places dataset [22]. Similarly, Sünderhauf proposed an integrated hashing algorithm and semantic search space partitioning technique, which accelerated loop closure detection by utilizing the Hamming distance, resulting in a 99.6% speed increase [23]. Shahid fine-tuned pre-trained AlexNet with different distance metrics, concluding that cosine distance works more accurately than Euclidean distance [24]. Overall, either retraining the off-the-shelf CNN model with a more targeted dataset or directly using a pre-trained CNN limits the loop closure detection process to feature extraction and matching, allowing loop closure detection to be performed online in real-time, which is beneficial for practical engineering applications [25,26].

### 1.2. Related Work Overview

The aforementioned studies focused on improving the accuracy and real-time performance of loop closure detection based on a CNN. However, in intelligent agricultural equipment, it is very difficult to apply large CNNs to small embedded devices. Meanwhile, the high accuracy and real-time requirements required in agricultural scenes bring more challenges in the study of loop closure detection. Based on this, this paper aims to investigate the relationship between lightweight CNN structures and detection accuracy, as well as real-time performance, and explore the effect of a hashing algorithm on CNN network

acceleration in a greenhouse scenario. Therefore, the main contributions of this paper are the innovative applications of existing algorithms, as detailed below:

- (1) During the image feature extraction phase in closed-loop detection, pre-existing CNN models are employed to replace traditional manual methods, such as SIFT, for extracting image features. Taking the accuracy and detection time of the detection algorithm as the evaluation criteria, we compare the VGG19 CNN with three lightweight CNNs, i.e., GhostNet, ShuffleNet V2, and Efficientnet-B0 models, in an open dataset. Meanwhile, we establish the Greenhouse dataset to verify that the most suitable model for loop closure detection is GhostNet.
- (2) Using the Random-Hyperplane Locality-Sensitive Hashing (RHLSH) algorithm to reduce dimensionality and match features, which are extracted by CNN models. To further accelerate the loop closure detection, two multi-probe random-hyperplane locality-sensitive hashing algorithms are selected to speed up the detection algorithm. In the proposed Greenhouse dataset, the experiments show that the step-wise probing random-hyperplane locality-sensitive hashing using linear scanning can significantly reduce the feature matching time with less accuracy loss.

## 2. Methods

Feature extraction and feature mapping are two key steps in loop closure detection, which mainly affect the detection accuracy and time of the algorithm. Based on this, in this section, we first select the appropriate lightweight CNN models for intelligent agricultural equipment, which have been pre-trained, and we only use the CNN models for image feature extraction, which does not require training the CNN models. Then, the image features extracted by CNN models are matched by a hash algorithm. Then two improved algorithms based on the hash algorithm are used to accelerate the matching and compare the performance. Meanwhile, we establish the GreenHouse dataset to demonstrate its performance. Accuracy–recall curves and average accuracy, as well as average time, are used as performance evaluation metrics for loop closure detection.

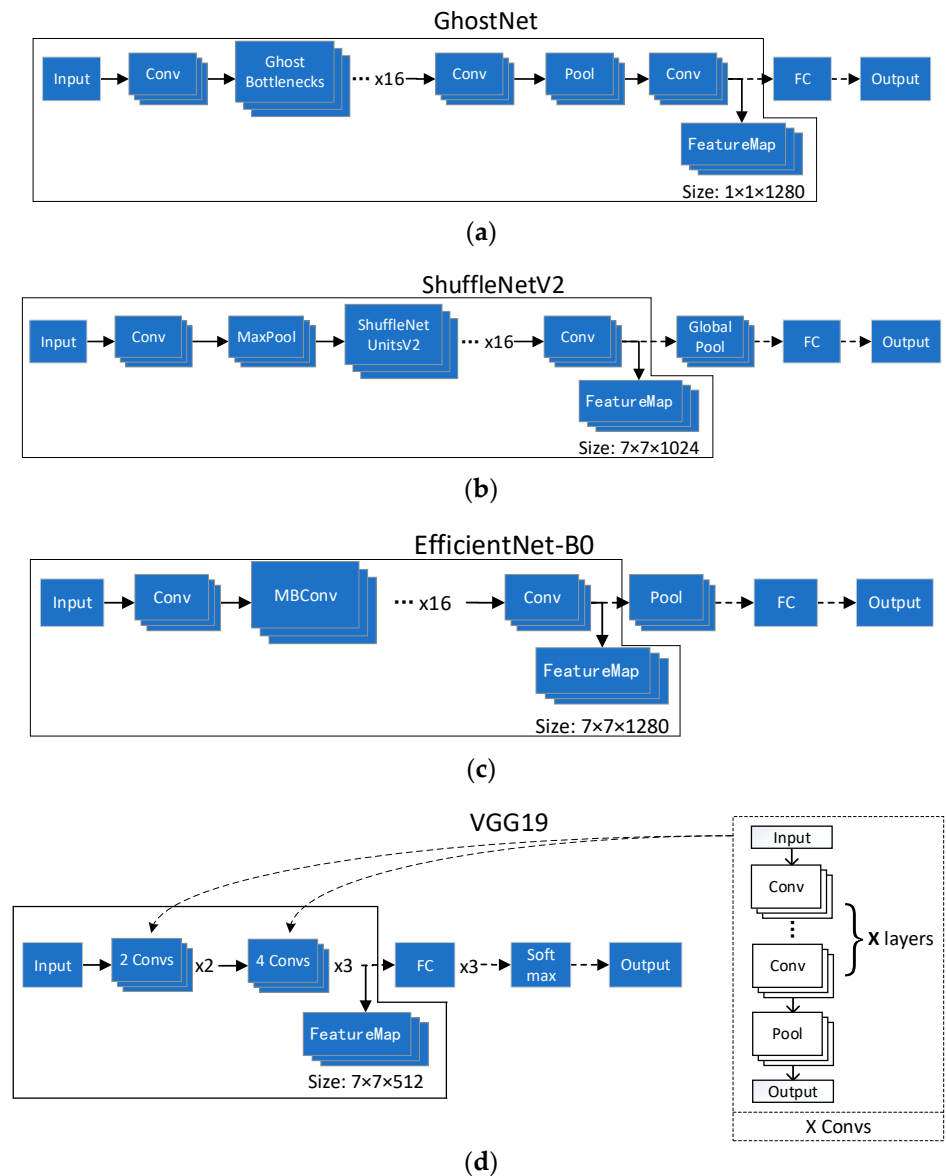
### 2.1. Feature Extraction Model Introduction in Loop Closure Detection

The current mainstream visual SLAM systems still rely on corner points to describe images, which limits their ability to characterize non-corner points, especially in images with fewer corners. In contrast, CNNs offer a more comprehensive approach to feature extraction by leveraging the rich data present in images. Lightweight CNNs, in particular, provide the advantage of compact model structures without sacrificing essential features found in larger CNNs.

The CNN model can be compressed by a variety of techniques, such as pruning, weight sharing, weight quantization, and Huffman coding, but these methods may overlook the significance of redundant features [27]. Of course, it is also possible to design efficient architectural models, thereby reducing model parameters and computational effort while preserving information about redundant features [28]. For example, ShuffleNet was constructed with specialized core units that combine resolution-related convolutional depth to minimize computational complexity and enhance efficiency [29]. GhostNet v2, another example, focuses on generating compact feature maps using linear operations while adopting channel mixing to optimize feature representation, effectively reducing the size of the convolutional network model [30]. VGG19 is based on deeper convolutional neural networks proposed by LeNet and AlexNet to achieve better performance [31]. Similarly, EfficientNet-B0 replaced the ResNet module with the MBConv module, enhancing the utilization of high-level feature information by redesigning the module architecture [28].

Due to the fact that efficient architecture models can reduce model parameters and computational workload while minimizing the loss of redundant information, this paper selects four lightweight CNNs—GhostNet, ShuffleNet v2, EfficientNet-B0, and VGG19—for image feature extraction, aiming to explore lightweight approaches that maintain crucial CNN features while enhancing loop closure detection performance. The structures of these

CNNs are depicted in Figure 1, illustrating their feature extraction process and closed-loop detection utilization. Each model, including GhostNet, ShuffleNet v2, and EfficientNet-B0, employs distinct feature reuse strategies to achieve efficiency and effectiveness in loop closure detection. The solid arrows in the figure represent the data flow within the parts of the CNN models used in this paper, while the dashed arrows indicate the original framework of the CNN models.



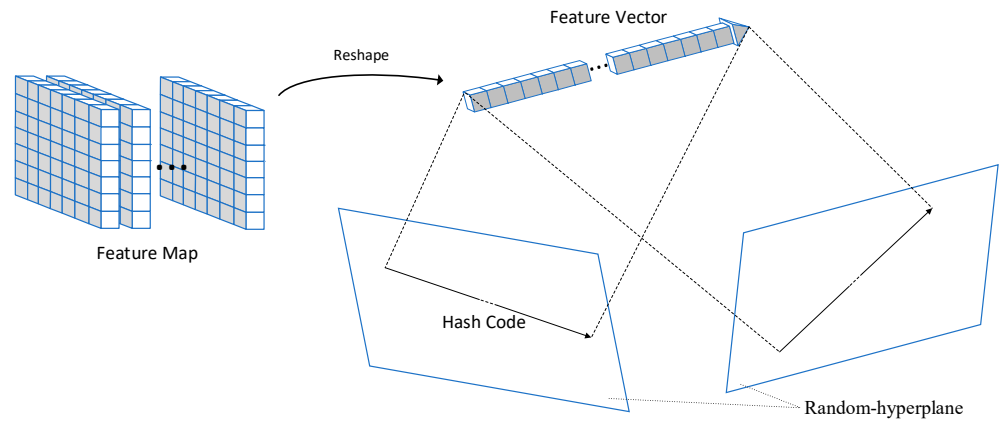
**Figure 1.** Structure images of four deep CNN models structure images. (a) The framework of GhostNet; (b) The framework of ShuffleNet; (c) The framework of EfficientNet-B0; (d) The framework of VGG19.

## 2.2. Feature Matching in Loop Closure Detection with CNNs

A visual bag-of-words (BoW) model based on manually designed features is the most commonly used solution for loop detection [32–39]. This method involves extracting feature points from images using algorithms such as SIFT, SURF, or ORB, followed by clustering to divide these points and their descriptors into multiple words. This allows the detection of related feature vectors for the image through the BoW mapping. Here, we adopt a BoW model based on SIFT feature points and use cosine similarity to measure image similarity.

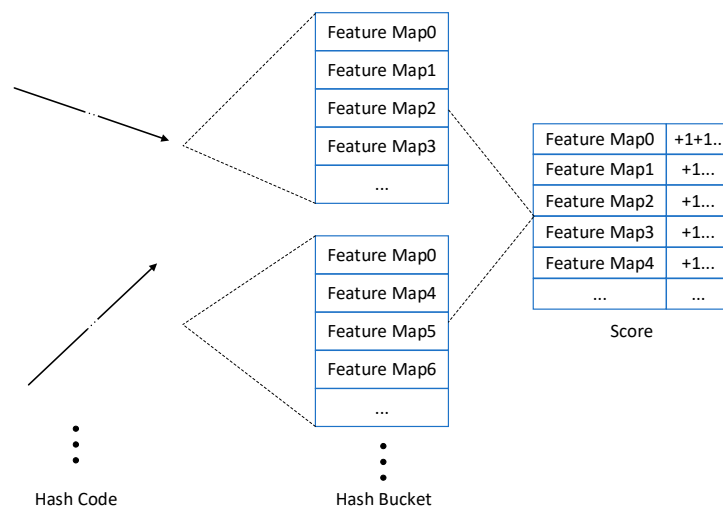
In agricultural settings, the abundance of local feature points and the scene's element similarity render traditional methods less practical compared to those based on CNNs.

However, CNN-extracted feature vectors often suffer from high dimensionality, necessitating methods like the RHLSH algorithm for downsizing and initial retrieval of image feature vectors. RHLSH partitions high-dimensional space using random hyperplanes and organizes vectors based on their positions [40]. As illustrated in Figure 2, the CNN-extracted feature map is reshaped into a feature vector and projected onto randomly generated hyperplanes via hash function families represented by Hamming code. This approach effectively represents the high-dimensional feature map using hash codes on randomly generated, relatively low-dimensional hyperplanes.



**Figure 2.** RHLSH projects high-dimensional feature graphs onto relatively low-dimensional graphs.

In high-dimensional space, any randomly sampled normal vector following the standard multivariate normal distribution  $N(0, I)$  has an equal probability of occurrence in all directions, ensuring uniform sampling [41]. Consequently, projecting onto multiple hyperplanes and calculating matching scores can enhance the matching accuracy of feature maps. The workflow is as follows: after an image is processed by CNNs and the hash code is generated, the corresponding hash bucket's feature map is tallied. Each hash code in different hyperplanes corresponds to a distinct hash bucket. Occasionally, multiple feature maps may reside in a hash bucket, indicating that a hash code may correspond to various feature maps. Therefore, statistical analysis of the feature maps within the hash bucket is necessary. Ultimately, the feature map with the highest score surpassing the preset threshold is deemed successfully matched. Otherwise, if the score falls below the threshold, the matching fails, indicating that loop closure has not occurred. This entire process is depicted in Figure 3.



**Figure 3.** RHLSH matches the image with a hash code.

Increasing the number of hash function families and hash tables can enhance search accuracy and recall, but it also escalates memory space usage. To mitigate this, expanding the search range within the same hash table can be beneficial. Multi-probe Random-Hyperplane Locality-Sensitive Hashing (RHLSH) is an exploration method that improves search recall to some extent. Key strategies for expanding the search range include Step-Wise Probing RHLSH (SWP-RHLSH) and Query-Directed Probing RHLSH (QDP-RHLSH).

For SWP-RHLSH, the Boolean hash value of the feature vector allows for gradual search range expansion based on the number of differing bits in the hash value. As feature vectors dynamically increase in loop closure detection, a linear scan is employed initially to determine hash bucket perturbations within a specified range, expediting the search.

For QDP-RHLSH, a random hyperplane within the same hash table further refines search probability. Hash buckets with a higher likelihood of containing nearest neighbor feature vectors are prioritized, reducing incorrect feature vector exploration. An evaluation probability function with respect to the random hyperplane can be defined as (2) for a given sequence of perturbation vectors  $V_p = [\delta_1, \delta_2, \dots, \delta_k]^T$  with  $\delta_l \in [0, 1]$ . When  $\delta_l = 0$ , indicating no perturbation, the probability of collision is (1) [41].

$$P\{H(v_1) = H(v_2) | \theta(v_1, v_2)\} = \prod_{j=1}^k \Phi\left(\frac{\|p_j^T \cdot v_2\|}{\|v_2\| \cdot \tan \theta(v_1, v_2)}\right) \quad (1)$$

$$W_{i,j}(1) = \prod_{j=1}^k \frac{P\{H(v_1) = H(v_2) | \theta(v_1, v_2)\}}{1 - P\{H(v_1) = H(v_2) | \theta(v_1, v_2)\}} \quad (2)$$

where  $\Phi$  is the standard normal distribution function;  $p_j$  is the normal vector of the random hyperplane;  $\theta(v_1, v_2)$  is the angle between the two nearest neighbor feature vectors, and the range of values is usually  $\theta(v_1, v_2) \in (0, \pi/2)$ .  $W_{i,j}(1)$  is defined as the evaluation probability function of the hash code corresponding to the  $j$  hash function under the  $I$  feature vector to be matched after adding perturbations.

Together with the use of the shift transform (3) and the expend transform (4) [42], the construction of the maximum heap with  $W_{i,j}(1)$  as the weights can be achieved to obtain the perturbation vector of the top  $M$  maximum weights, where the perturbation vector is transformed from the set of perturbations, taking  $k = 4$  as an example: assume that the results of the descending sort of  $W_{i,j}(1)$  are  $\{W_{i,3}(1), W_{i,1}(1), W_{i,4}(1), W_{i,2}(1)\}$ , and for the perturbation set  $S = \{1, 4\}$ , the first and fourth positions of  $W_{i,j}(1)$  after descending sorting are chosen as the perturbation positions, and the perturbation vector is  $V_p = [0, 1, 1, 0]^T$ .

$$\text{shift}(S) = \{\max(S) + 1\} \cup \{S - \{\max(S)\}\} \quad (3)$$

$$\text{expend}(S) = \{\max(S) + 1\} \cup S \quad (4)$$

Among them, the shift transform operation does not work on the empty set, and each operation only adds 1 to the value of the largest element in the perturbation set, while the expend transform operation adds an element larger than the largest element by 1 to the perturbation set. Due to the limitation of the number of perturbation bits  $M \in [0, k]$ , their two operations gradually stop.

In practical loop closure detection systems, when probing the initial  $M$  hash buckets, the number of hash buckets for previous image features that need matching increases dynamically. This results in a high proportion of hash buckets that do not exist, leading to significant search time consumption. Therefore, the probing count  $M$  should not be a fixed value but rather a segmented function that adjusts based on the number of hash buckets. We define the probing count  $M$  as 1000 when the number of hash buckets exceeds 500; otherwise, it is set to 1500.



### 2.3. Use of Hardware and Software

The hardware devices utilized in our study were sourced from Intel Corporation, a leading technology company based in Santa Clara, California, USA. Specifically, we employed an Intel NUC11 Mini PC equipped with a 2.80 GHz Intel Core i7 CPU and 16 GB RAM, along with an Intel D435i depth camera, which features active stereo IR technology for capturing depth images. The experimental software environment was an Ubuntu 20.04 LTS 64-bit system with Python3.6 under the deep learning framework PyTorch.

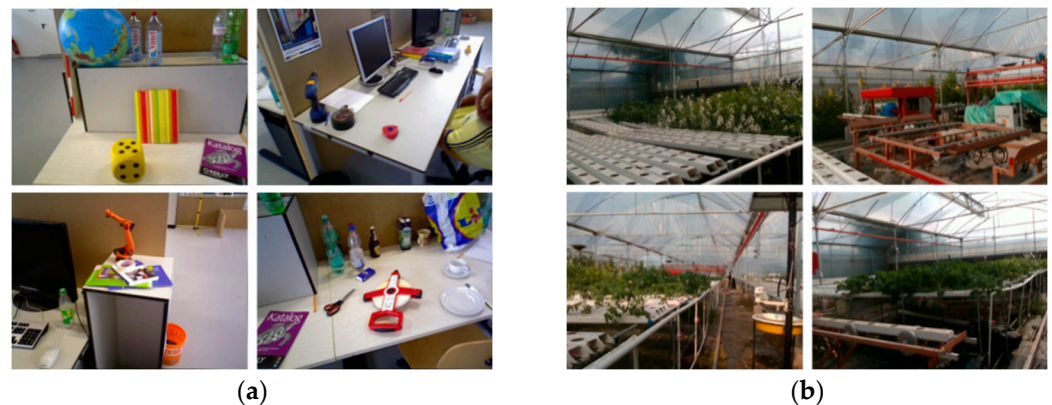
### 2.4. Datasets and Pre-Processing

We utilized the TUM dataset and the greenhouse scene dataset captured with the D435i depth camera, as presented in Table 1. The TUM dataset, sourced from the computer vision group at the Technical University of Munich, Germany, is commonly employed for RGB-D SLAM research. This dataset provides coordinate files of camera motion trajectories detected by high-precision sensors. On the other hand, the greenhouse scene dataset was gathered on 19 February 2021, at 10:00 a.m. in the plant factory of South China Agricultural University, located in Guangzhou, Guangdong Province.

**Table 1.** Specific information about the dataset.

Dataset Name	Number of Images	Duration/s
TUM fr3/long_office_household (TUM)	2486	87.09
the greenhouse scene dataset (GreenHouse)	2261	82.94

The TUM dataset contains a variety of objects, such as office desks, chairs, computer equipment, and robotic arm models, providing abundant texture and structure for image feature extraction. Additionally, the camera trajectory in this dataset forms a large circular closed trajectory with overlap at the initial and final points. This setup mirrors conditions often found in agricultural scenes, characterized by rich texture structures (as depicted in Figure 4a).



**Figure 4.** Example of a dataset image. (a) TUM; (b) GreenHouse.

However, the TUM dataset lacks ground truth information for evaluating loop closure detection algorithms. Instead, it offers camera motion trajectory coordinate files detected by high-precision sensors. To establish correlations between pose coordinate files and image data, the scripting tool provided by TUM was utilized. Matches were defined between pose coordinates and image data with a time difference within 0.02 s. The occurrence of loop closure was determined by calculating the pose error between any two frames within the matched camera pose coordinates. Given the relatively minor positional changes between

adjacent images, positional errors of the neighboring 150 images are disregarded. The positional error calculation is expressed as Equation (5).

$$e = \|T_i^{-1}T_j - I\| \quad (5)$$

where  $T$  is the camera pose; subscripts  $i, j$  are the image serial numbers,  $i = \{1, 2, \dots, n\}$  and  $j = \{1, 2, \dots, (i - 150)\}$ ;  $I$  is the unit matrix.

The collection time of the greenhouse scene dataset is chosen at 10:00 am when the light intensity is high. The dataset includes a variety of green vegetables that have been planted on cultivators, blank cultivators that have not been planted, automated agricultural equipment, and other common agricultural production environment elements. It is also ensured that a large circular closed trajectory exists in the dataset (as shown in Figure 4b). The GreenHouse dataset captures authentic greenhouse agricultural scenes using the D435i depth camera. Cameras are typically categorized as monocular, binocular, and RGBD. Monocular and binocular cameras require depth estimation through algorithms, while RGBD cameras can directly measure depth. Consequently, RGBD cameras exhibit the highest average depth accuracy among the three types. Therefore, the ORB-SLAM2 system is employed to compute the D435i camera's motion trajectory in the greenhouse scene dataset, serving as the reference trajectory. And Formula (5) is applied to derive the ground truth for loop closure detection.

The loop closure detection ground truth is saved in the form of a matrix. If the  $i$  image and the  $j$  image constitute a loop closure, the corresponding value of the ground truth matrix ( $i, j$ ) is 1, and the opposite is 0.

### 2.5. Experimental Evaluation Criteria

Loop closure detection performance is typically evaluated using accuracy–recall (PR) curves, with the overall assessment based on average accuracy (AP). However, beyond accuracy and recall, the time required for loop closure detection is also critical. Feature extraction time and feature matching time are the two primary time-consuming components in loop closure detection. When features are obtained directly from the front-end vision odometer, feature matching time becomes the dominant cost. To comprehensively assess the application of loop closure detection modules in different visual SLAM systems, this paper conducts experiments to separately analyze feature extraction time and feature matching time.

## 3. Results

### 3.1. Feature Extraction Comparative Experiment

To facilitate a more concise comparison of the performance variations among three lightweight CNN models—GhostNet, ShuffleNet v2, and EfficientNet-B0—in extracting features from RGB-D images, and to further explore the influence of lightweight CNN models on loop closure detection in agricultural scenes, we conducted three sets of experiments:

The first experiment is to respectively extract RGB image features from the TUM dataset using a visual word bag model method based on SIFT features and pre-trained VGG19 and three pre-trained lightweight CNN models.

The second and third sets of experiments utilized a pre-trained VGG19 model and the three pre-trained lightweight CNN models to extract RGB-D image features from the TUM and GreenHouse datasets, respectively.

In the latter two sets of experiments, the depth image was replicated to match the number of depth image channels with the RGB image channels. The merged and concatenated RGB and depth image features formed the image feature vector, ensuring the integrity of the extracted feature information. Through these combined approaches, a comprehensive analysis of the performance disparities among the lightweight CNN models and their impact on loop closure detection in agricultural scenes can be conducted, fostering a deeper understanding of their applicability and effectiveness. The accuracy of various algorithms



for feature extraction for loop closure detection was measured by calculating the cosine similarity between image features.

In the result analysis, Formulas (6) and (7) are used to calculate the degree of variation in the value–optimization rate.

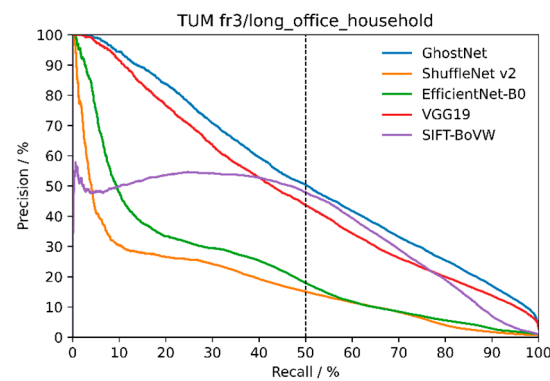
$$R_a = \frac{A_f - A_r}{A_r} \quad (6)$$

$$R_t = -\frac{T_f - T_r}{T_r} \quad (7)$$

where  $R_a$  represents the accuracy optimization rate;  $A_f$  represents the accuracy rate after optimization;  $A_r$  represents reference accuracy;  $R_t$  represents the time cost optimization rate;  $T_f$  represents the time cost after optimization;  $T_r$  represents the reference time cost.

### 3.1.1. The Results of the Extract RGB Image Features Experiment

The results of the first combined experiment are shown in Figure 5. From the trend of the PR curve, it can be seen that the curve with the largest bias at the top right is based on the algorithm of the GhostNet model. With the increase in the judgment threshold, the image similarity matching the actual situation obtained by the algorithm increases, and the recall rate also increases, but there are some wrong judgments, and the accuracy rate decreases. In addition, it is worth noting that the accuracy of the visual word bag model algorithm based on SIFT features cannot reach 100%. Among CNN algorithms, when the accuracy reaches 100%, the algorithms based on the GhostNet model have the slowest decline rate, followed by ShuffleNet v2, EfficientNet-B0, and VGG19. These observations show that the feature extraction model algorithm based on a CNN can maintain a certain recall rate under the condition of high accuracy, and the model algorithm based on GhostNet has the highest recall rate. At a 50% recall rate, the traditional algorithm has better accuracy than the three algorithms based on ShuffleNet v2, EfficientNet-B0, and VGG19 models, and is closer to the accuracy of the algorithm based on the GhostNet model.



**Figure 5.** PR curve of extracted RGB image features from the TUM dataset.

As shown in Table 2, the GhostNet model’s algorithm achieves a substantial 40.5% enhancement in average accuracy over traditional methods. The VGG19 model’s algorithm also records a significant 26.0% improvement. In contrast, the ShuffleNet v2 and EfficientNet-B0 models’ algorithms result in average accuracy decreases of 53% and 38.6%, respectively. Regarding feature extraction time, the ShuffleNet v2 and EfficientNet-B0 models offer notable reductions, with time savings of 41.2% and 25.5% for processing a single image, compared to conventional methods. Additionally, the GhostNet model’s algorithm realizes a 29.4% improvement in average extraction time. Conversely, while the VGG19 model does boost average accuracy, its feature extraction time more than doubles that of the traditional algorithm. In conclusion, the GhostNet model-based algorithm stands out for its feature extraction efficiency and accuracy in loop closure detection within the TUM dataset’s RGB images, suggesting it is superior for these tasks.

**Table 2.** Comparison of extracted RGB image features from the TUM dataset.

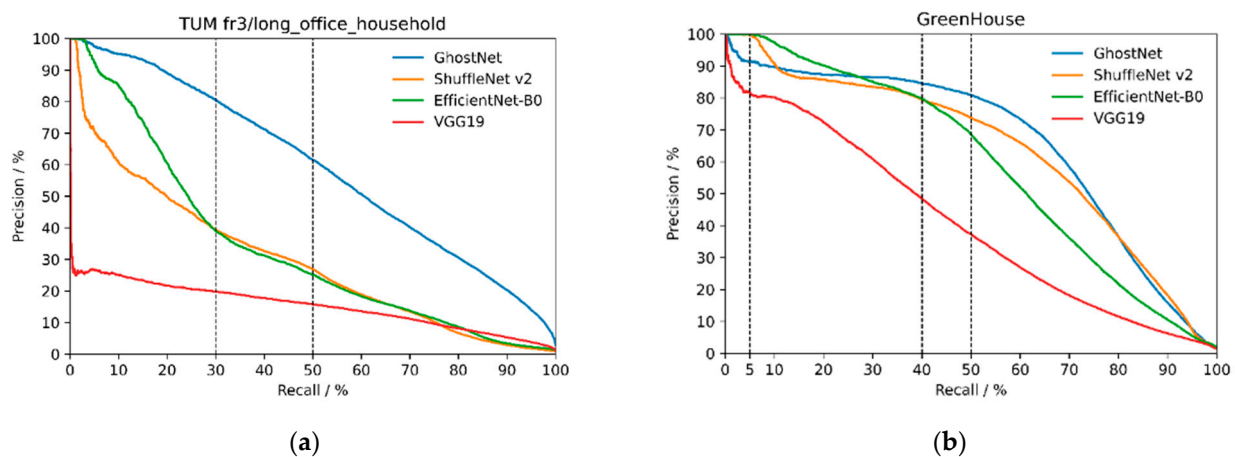
	Average Accuracy Rate/%	Average Time for Feature Extraction/s	Average Accuracy Optimization Rate/%	Average Time Cost Optimization Rate/%
GhostNet	53.1	0.036	40.5	29.4
ShuffleNet v2	17.8	0.030	−53.0	41.2
EfficientNet-B0	23.2	0.038	−38.6	25.5
VGG19	47.6	0.126	26.0	−147.1
SIFT-BoVW	37.8	0.051	0.0	0.0

Using SIFT-BoVW's data as reference values.

### 3.1.2. The Results of the Extract RGB-D Image Features Experiment

At this stage, we refrain from concluding that GhostNet is the optimal model for loop closure detection in intelligent agricultural equipment. The uncertainty arises from the unconfirmed impact of integrating depth image information. To address this, we conducted cross-sectional comparison tests for extracting RGB-D image features in both the TUM and GreenHouse datasets. Furthermore, considering that feature extraction from RGB and depth images occurs simultaneously, the average feature extraction time reflects the combined cost of feature extraction and stitching in RGB-D images.

Figure 6a illustrates that the algorithm based on the GhostNet model retains the most right-side-up PR curve even after incorporating depth image feature vectors, indicating its robustness. At a 100% accuracy rate, the GhostNet, ShuffleNet v2, and EfficientNet-B0 models all show relatively stable recall rates. In contrast, the VGG19 model's PR curve declines earliest, suggesting it is less effective at maintaining high recall rates and more susceptible to misclassification, which implies a significant perceptual bias. At a 50% recall rate, the GhostNet model maintains a high accuracy rate, with ShuffleNet v2 showing only a slightly lower performance. The EfficientNet-B0 model also exhibits a similar trend, although it begins to decline more noticeably around a 30% recall rate.



**Figure 6.** PR curve of extracted RGB-D image features from TUM/GreenHouse dataset. (a) For the TUM dataset; (b) for the GreenHouse dataset.

Figure 6b further demonstrates that the GhostNet model-based algorithm leads in terms of position towards the upper-right corner among the CNN models, signifying a higher recall rate without sacrificing accuracy. While the ShuffleNet v2 and EfficientNet-B0 models show higher recall rates at 100% accuracy, they fall behind GhostNet in terms of maintaining accuracy. Notably, the VGG19 model consistently exhibits the lowest recall rates. At a 10% recall rate, the GhostNet and ShuffleNet v2 models have identical accuracy rates, which then diverge as the GhostNet model's PR curve stabilizes before decreasing again around a 40% recall rate. Here, both the ShuffleNet v2 and EfficientNet-B0 models show slightly lower accuracies compared to GhostNet. At the 50% recall rate, all three

lightweight CNN models maintain high accuracy rates, whereas the VGG19 model lags behind. Additionally, the GhostNet model delivers superior accuracy at a 75% recall rate.

In conjunction with Table 3, the GhostNet model-based algorithm demonstrates the highest average accuracy among the four algorithms, reaching 59.4% for extracting RGB-D image features for loop closure detection. The feature extraction time for a single image is only 2 ms longer than that of the ShuffleNet v2 model algorithm. In summary, it shows that the GhostNet-based model algorithm outperforms the other three CNN-based model algorithms and the three lightweight CNN model algorithms in extracting features for loop closure detection on RGB-D images in the TUM dataset. The results shown in Table 4 are similar to those shown in Table 3: the GhostNet model-based algorithm is the best in terms of overall performance.

**Table 3.** Comparison of extracted RGB-D image features from the TUM/GreenHouse dataset.

	Average Accuracy Rate/%	Average Time for Feature Extraction/s	Average Accuracy Optimization Rate/%	Average Time Cost Optimization Rate/%
GhostNet	59.4	0.055	0.0	0.0
ShuffleNet v2	29.6	0.053	−50.2	3.7
EfficientNet-B0	33.5	0.072	−43.6	−30.9
VGG19	15.4	0.240	−74.1	−336.4

Using GhostNet’s data as reference values.

**Table 4.** Comparison of extracted RGB-D image features from the GreenHouse dataset.

	Average Accuracy Rate/%	Average Time for Feature Extraction/s	Average Accuracy Optimization Rate/%	Average Time Cost Optimization Rate/%
GhostNet	66.2	0.057	0.0	0.0
ShuffleNet v2	64.0	0.052	−3.3	8.8
EfficientNet-B0	59.2	0.072	−10.6	−26.3
VGG19	40.6	0.251	−38.7	−340.4

Using GhostNet’s data as reference values.

### 3.2. The Results of Feature Maps Match Experiment

Feature matching is another time-consuming aspect of loop closure detection. The GhostNet model, identified in Section 3.1, is employed to conduct a horizontal comparison between two multi-probe RHLSH algorithms. The cosine similarity matrix is still utilized to generate the PR curve, and the average feature-matching time is measured to assess the performance difference between the two feature-matching strategies in the greenhouse dataset.

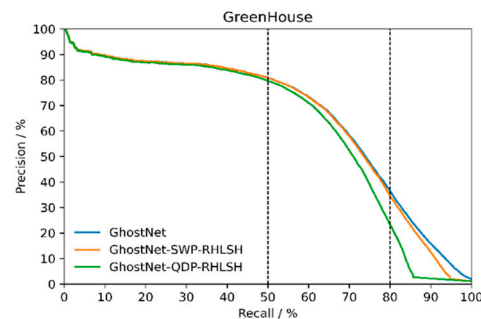
Considering practical application, data points in loop closure detection accumulate over time, transitioning from sparse to dense. In the loop closure detection module of RGB-D SLAM, keyframes are mainly used to increase data sparsity. Using too many bits of hash value in this case hinders real-time loop closure detection. Therefore, all comparison tests in this chapter use a 16-bit hash value, dividing the high-dimensional space into  $2^{16}$  regions by random hyperplane, while utilizing all images for loop closure detection. Specific parameter settings are detailed in Table 5.

**Table 5.** Parameter settings.

	Size of Hash Value/bit	Other Parameters
GhostNet	None	None
GhostNet + SWP-RHLSH	16	$m = 4$
GhostNet + QDP-RHLSH	16	$M = \begin{cases} 1000(N_1 < 500) \\ 1500(N_1 \geq 500) \end{cases}$

where  $m$  represents the size of the perturbation vector;  $M$  represents the probing count;  $N_1$  represents the number of hash buckets.

As depicted in Figure 7, the PR curves derived from SWP-RHLSH and those without a matching strategy are nearly indistinguishable until an 80% recall rate, while the PR curves from QDP-RHLSH only overlap significantly until a 50% recall rate. With increasing recall rates, the QDP-RHLSH-based algorithm exhibits more misclassifications compared to the SWP-RHLSH-based algorithm.



**Figure 7.** PR curve of loop closure detection for greenhouse scene dataset.

As analyzed in Table 6, the SWP-RHLSH-based algorithm experiences a slight decrease in average accuracy by 1.2% but achieves a faster average matching time by 45.1% compared to the loop closure detection algorithm without the matching strategy. Meanwhile, the QDP-RHLSH-based algorithm exhibits a reduction in average accuracy by 6.3%, with the average matching time being faster by 48.3%. Overall, both algorithms contribute to improved loop closure detection times while maintaining acceptable levels of accuracy loss. Notably, for each 1% improvement in the average matching time, the accuracy loss is only 0.027 for the SWP-RHLSH-based algorithm and 0.130 for the QDP-RHLSH-based algorithm. Therefore, for the image-matching acceleration algorithm in the closed-loop detection in the greenhouse scene, the SWP-RHLSH-based algorithm is a better choice.

**Table 6.** SWP-RHLSH-based and QDP-RHLSH-based loop closure detection algorithm.

	Average Accuracy Rate/%	Average Time for Feature Matching/s
GhostNet	66.2	1.734
GhostNet + SWP-RHLSH	65.4	0.952
GhostNet + QDP-RHLSH	62.0	0.896

#### 4. Physical Experiment

We have integrated the loop closure detection algorithm into the feature-based visual odometry system to optimize the trajectory it generates. We assembled a bespoke platform featuring a D435i camera, which we affixed to the physical mobile robot, ‘Thunder’, for our experiments. ‘Thunder’ is a mobile robot produced by Chaowenda Robot Technology, located in Shenzhen, China. We conducted tests in both a standard orchard and a greenhouse, as depicted in Figure 8.

We carried out four experiments under various conditions, detailed as follows:

For visual observation, we drove the robot around the field in a rectangular path.

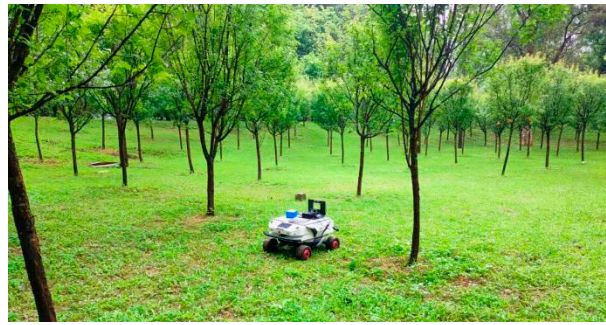
The camera trajectory generated by testing in the greenhouse is illustrated in Figure 9a.

The outdoor orchard test, conducted on a sunny day, yielded the camera trajectory shown in Figure 9b.

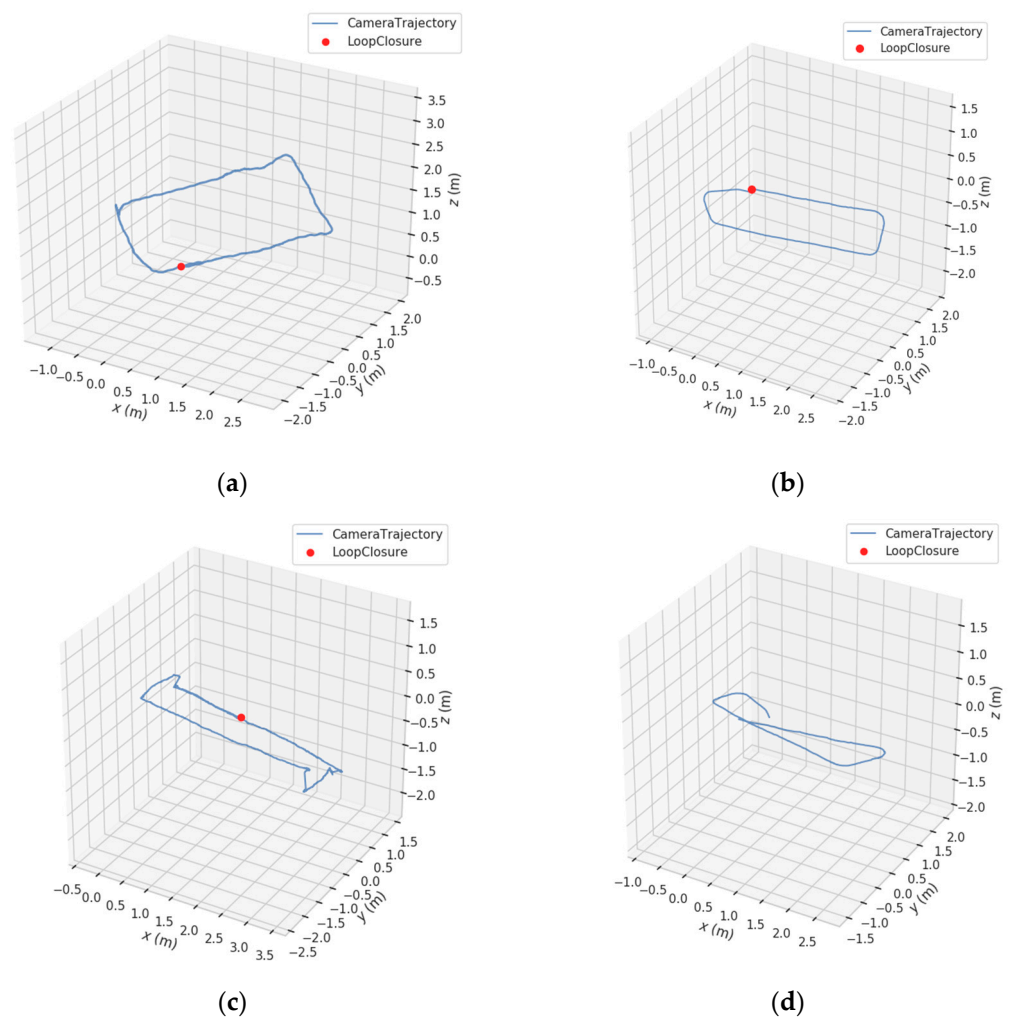
On an overcast day, the resulting camera trajectory from the outdoor orchard test is presented in Figure 9c.

The camera trajectory obtained from the outdoor orchard test on a rainy day, with a precipitation level of 2.81 mm, is also displayed in Figure 9d.

In Figure 9a,b, the blue curve represents the camera trajectory as estimated by the SLAM system, while the red circle highlights the location of the loop closure detection event.



**Figure 8.** The physical mobile robot, ‘Thunder’, collects data in a standard orchard.



**Figure 9.** Camera tracks produced by the SLAM system under various conditions. (a) In greenhouse; (b) sunny orchard; (c) cloudy orchard; (d) rainy orchard.

The experimental results reveal that the algorithm examined in this study functions effectively on both sunny and overcast days in greenhouses and outdoor orchards. However, its loop closure detection capability is compromised during rainy conditions. We attribute this malfunction to the significant interference of raindrops, which not only obfuscate the visual distinctions between different locations within the orchard but also exacerbate the image blurriness, hindering the algorithm’s performance. Moreover, Figure 9c,d suggest that reduced light intensity profoundly affects the visual odometer’s accuracy. Specifically, the trajectory illustrated in Figure 9c shows a substantial increase in camera trajectory



jitter during turns made under low-light conditions. Specifically, the trajectory depicted in Figure 9c demonstrates a significant augmentation in camera trajectory jitter during turns executed in low-light conditions, and Figure 9d reveals noticeable camera trajectory distortion. Figure 9d also reveals that the cumulative error of the SLAM system, in the failure of loop closure detection, is considerably large within an agricultural setting. Given that agricultural tasks frequently involve returning to previously visited locations, and the visual similarity within farms or fruit orchards is relatively high, the implementation of loop closure detection is more vital in agricultural scenarios compared to other contexts.

## 5. Discussion

In Section 3.1, image features extracted by CNN models are typically faster than manual methods. We consider that the TUM dataset contains rich image textures, resulting in the extraction of numerous corner points, which in turn increases the computation time required for SIFT descriptors. However, in scenarios where image textures and corner points are scarce, the accuracy of image feature matching based on descriptors may not be desired [43]. Therefore, it can be considered that the manual methods are less adept at capturing redundant information than CNN models. We identified the GhostNet model as the most suitable model for loop closure detection in intelligent agricultural equipment among the four CNN models discussed in the article. Figures 5 and 6a illustrate that the PR curves of algorithms based on the EfficientNet-B0 model exhibit a similar trend. This similarity is primarily attributed to the Squeeze-and-Excitation (SE) module, which effectively utilizes crucial deep feature information while disregarding less important details [44]. Given the resemblance between depth images and shallow feature information, the SE module tends to overlook more information, resulting in a similar trend in the curves.

Moreover, the overall trend of the PR curves of the ShuffleNet v2 model-based algorithm in these plots differs considerably due to the channel mixing operation. The feature information of the depth image retains the original shallow information, improving the reuse rate of the feature information. The GhostNet model-based algorithm significantly enhances the accuracy and real-time performance of loop closure detection, mainly due to the cheap linear operation inside the Ghost module and the Batch Normalization (BN) operation outside it [45]. The cheap linear operation enables deeper features to incorporate shallow feature information, ensuring comprehensive data description. Compared to traditional algorithms, CNNs offer better accuracy and stability in feature extraction. In the GreenHouse dataset, scene elements and details are richer than in the TUM dataset, demanding higher feature extraction ability from model algorithms. Thus, for agricultural scenes, loop closure detection algorithms based on lightweight CNN models require models that can retain more redundant feature information with fewer parameters, thereby facilitating adaptation to the farming environment.

Regarding Section 3.2, the SWP-RHLSH-based algorithm considers all elements in close hash buckets, improving the real-time performance of RGB-D SLAM loop closure detection through a linear scan. Conversely, the QDP-RHLSH-based algorithm's real-time performance improvement is less evident due to the resource consumption caused by its floating-point operations [40,41]. Based on comparison experiments, the GhostNet-based algorithm for image feature extraction combined with SWP-RHLSH for search range filtering is deemed most suitable for loop closure detection implementation in intelligent agricultural equipment among the four CNN models discussed in the article. The physical experiment outcomes demonstrate that the algorithm presented in this paper exhibits robustness within agricultural settings. It operates effectively in both the greenhouse environment and across various weather conditions encountered in outdoor orchards.

## 6. Conclusions

With the growing need for precision and intelligence in agricultural machinery, loop closure detection in visual SLAM must not only determine spatial congruence but also be adaptable to various small embedded devices on smart agricultural equipment. This

study compares multiple loop closure detection methods based on lightweight CNN features. It is observed that features extracted from GhostNet significantly enhance both the accuracy and real-time performance of loop closure detection. This is attributed to the Ghost module in GhostNet, which preserves redundant features, enabling deeper feature information to encompass shallow features through feature reuse. To further expedite loop closure detection, two multi-probe random-hyperplane locality-sensitive hashing (RHLSH) algorithms are compared experimentally. SWP-RHLSH, employing linear scanning, markedly reduces feature matching time with minimal accuracy loss, making it more suitable for use in intelligent agricultural equipment detection algorithms. This is due to the smaller number of hash buckets screened by SWP-RHLSH in small to medium-sized agricultural settings, eliminating the need for floating-point operations to evaluate probabilities.

However, this study still has limitations. It utilized pre-trained CNN models for image feature extraction and did not investigate the impact of training the CNN models on the TUM/Greenhouse datasets on image feature extraction. Furthermore, it is important to note that this is a preliminary, phased outcome. A comprehensive SLAM system is currently absent, which is essential for fully realizing the algorithm's potential. Future research will progressively refine a SLAM system tailored for agricultural settings and deploy it into devices for further testing. For instance, it will be applied to non-standard orchard scenarios and dense crop environments. Furthermore, there is a need to delve deeper into how SLAM technology in agricultural settings can be integrated with artificial intelligence techniques to enhance the accuracy and computational speed of positioning in smart agricultural equipment. At the same time, this would help conserve hardware resources for other tasks and improve operational efficiency, navigation accuracy, and task planning in real agricultural environments.

**Author Contributions:** Conceptualization, H.Q.; methodology, H.Q., C.W. and J.L.; software, C.W. and J.L.; validation, H.Q. and L.S.; formal analysis, C.W. and J.L.; investigation, C.W. and J.L.; resources, H.Q.; data curation, J.L.; writing—original draft preparation, J.L. and C.W.; writing—review and editing, C.W. and H.Q.; visualization, C.W.; supervision, H.Q.; project administration, H.Q.; funding acquisition, H.Q. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by the subject construction projects in specific universities, which is a subject construction project at South China Agricultural University, with the funding number 2023B10564003.

**Institutional Review Board Statement:** This study did not require ethical approval.

**Data Availability Statement:** The relevant GreenHouse dataset for this study is available at <https://github.com/SCAU-AIUS/SLAM-for-agricultural-equipment> (accessed on 30 May 2024).

**Acknowledgments:** The authors acknowledge the editors and reviewers for their constructive comments and all the support on this work. The authors acknowledge Quanchen Ding for polishing the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Pan, Z.; Hou, J.; Yu, L. Optimization RGB-D 3-D Reconstruction Algorithm Based on Dynamic SLAM. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–13. [[CrossRef](#)]
2. Nguyen, D.D.; Elouardi, A.; Florez, S.A.; Bouaziz, S. HOOFR SLAM system: An embedded vision SLAM algorithm and its hardware-software mapping-based intelligent vehicles applications. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 4103–4118. [[CrossRef](#)]
3. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [[CrossRef](#)]
4. Zou, Q.; Sun, Q.; Chen, L.; Nie, B.; Li, Q. A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 6907–6921. [[CrossRef](#)]
5. Wang, T.; Chen, B.; Zhang, Z.; Li, H.; Zhang, M. Applications of machine vision in agricultural robot navigation: A review. *Comput. Electron. Agric.* **2022**, *198*, 107085. [[CrossRef](#)]

6. Wen, S.; Zhao, Y.; Yuan, X.; Wang, Z.; Zhang, D.; Manfredi, L. Path planning for active SLAM based on deep reinforcement learning under unknown environments. *Intell. Serv. Robot.* **2020**, *13*, 263–272. [[CrossRef](#)]
7. Peng, J.; Shi, X.; Wu, J.; Xiong, Z. An object-oriented semantic slam system towards dynamic environments for mobile manipulation. In Proceedings of the 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Hong Kong, China, 8–12 July 2019; pp. 199–204.
8. Simon, J. Fuzzy Control of Self-Balancing, Two-Wheel-Driven, SLAM-Based, Unmanned System for Agriculture 4.0 Applications. *Machines* **2023**, *11*, 467. [[CrossRef](#)]
9. Zhu, H.; Xu, J.; Chen, J.; Chen, S.; Guan, Y.; Chen, W. BiCR-SLAM: A multi-source fusion SLAM system for biped climbing robots in truss environments. *Robot. Auton. Syst.* **2024**, *176*, 104685. [[CrossRef](#)]
10. Song, S.; Yu, F.; Jiang, X.; Zhu, J.; Cheng, W.; Fang, X. Loop closure detection of visual SLAM based on variational autoencoder. *Front. Neurobot.* **2024**, *17*, 1301785. [[CrossRef](#)]
11. Tsintotas, K.A.; Bampis, L.; Gasteratos, A. The Revisiting Problem in Simultaneous Localization and Mapping: A Survey on Visual Loop Closure Detection. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 19929–19953. [[CrossRef](#)]
12. Guclu, O.; Can, A.B. Integrating global and local image features for enhanced loop closure detection in RGB-D SLAM systems. *Vis. Comput.* **2020**, *36*, 1271–1290. [[CrossRef](#)]
13. Xu, M.; Lin, S.; Wang, J.; Chen, Z. A LiDAR SLAM System with Geometry Feature Group Based Stable Feature Selection and Three-Stage Loop Closure Optimization. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–10. [[CrossRef](#)]
14. Angeli, A.; Doncieux, S.; Meyer, J.A.; Filliat, D. Real-time visual loop-closure detection. In Proceedings of the IEEE International Conference on Robotics & Automation, Pasadena, CA, USA, 19–23 May 2008.
15. Chen, Z.; Lam, O.; Jacobson, A.; Milford, M. Convolutional Neural Network-based Place Recognition. *arXiv* **2014**, arXiv:1411.1509.
16. Gao, X.; Zhang, T. Unsupervised Learning to Detect Loops Using Deep Neural Networks for Visual SLAM System. *Auton. Robot.* **2017**, *41*, 1–18. [[CrossRef](#)]
17. Jia, X. *Research on Loop Closure Detection of Mobile Robots Based on PCANet-LDA*; Harbin Institute of Technology: Harbin, China, 2019. (In Chinese)
18. Hou, Y.; Zhang, H.; Zhou, S. Convolutional Neural Network-based Image Representation for Visual Loop Closure Detection. In Proceedings of the 2015 IEEE International Conference on Information and Automation, Lijiang, China, 8–10 August 2015.
19. Xia, Y.; Li, J.; Qi, L.; Fan, H. Loop closure detection for visual SLAM using PCANet features. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016.
20. Xia, Y.; Li, J.; Qi, L.; Yu, H.; Dong, J. An Evaluation of Deep Learning in Loop Closure Detection for Visual SLAM. In Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 21–23 June 2017.
21. Wang, K. *Research on Loop Closure Detection of Visual SLAM Based on Deep Learning*; Harbin Engineering University: Harbin, China, 2019. (In Chinese)
22. Lopez-Antequera, M.; Gomez-Ojeda, R.; Petkov, N.; Gonzalez-Jimenez, J. Appearance-invariant Place Recognition by Discriminatively Training a Convolutional Neural Network. *Pattern Recognit. Lett.* **2017**, *92*, 89–95. [[CrossRef](#)]
23. Sunderhauf, N.; Shirazi, S.; Dayoub, F.; Upcroft, B.; Milford, M. On the Performance of ConvNet Features for Place Recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015.
24. Shahid, M.; Naseer, T.; Burgard, W. DTLC: Deeply Trained Loop Closure Detections for Lifelong Visual SLAM. In Proceedings of the Workshop on Visual Place Recognition, Conference on Robotics: Science and Systems (RSS), Ann Arbor, MI, USA, 18–22 June 2016.
25. Yu, C.; Liu, Z.; Liu, X.-J.; Qiao, F.; Wang, Y.; Xie, F.; Wei, Q.; Yang, Y. A DenseNet feature-based loop closure method for visual SLAM system. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019.
26. Zhang, X.; Yan, S.; Zhu, X. Loop Closure Detection for Visual SLAM Systems Using Convolutional Neural Network. In Proceedings of the 2017 23rd International Conference on Automation and Computing (ICAC), Huddersfield, UK, 7–8 September 2017.
27. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *Fiber* **2015**, *56*, 3–7.
28. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning 2019, Long Beach, CA, USA, 9–15 June 2019.
29. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *arXiv* **2017**, arXiv:1707.01083v2.
30. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
31. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
32. Zisserman, S. Video Google: A Text Retrieval Approach to Object Matching in Videos. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 14–17 October 2003.

33. Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; Zisserman, A. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007.
34. Angeli, A.; Filliat, D.; Doncieux, S.; Meyer, J.-A. Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words. *IEEE Trans. Robot.* **2008**, *24*, 1027–1037. [[CrossRef](#)]
35. Cummins, M.; Newman, P. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *Int. J. Robot. Res.* **2008**, *27*, 647–665. [[CrossRef](#)]
36. Cummins, M.; Newman, P. Appearance-only SLAM at Large Scale with FAB-MAP 2.0. *Int. J. Robot. Res.* **2011**, *30*, 1100–1123. [[CrossRef](#)]
37. Liang, M.; Min, H.; Luo, R. Graph-based SLAM: A Survey. *Robot* **2013**, *35*, 500–512. (In Chinese) [[CrossRef](#)]
38. Zhang, G.; Lilly, M.J.; Vela, P.A. Learning Binary Features Online from Motion Dynamics for Incremental Loop-Closure Detection and Place Recognition. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016.
39. Labbé, M.; Michaud, F. RTAB-Map as an Open-source Lidar and Visual Simultaneous Localization and Mapping Library for Large-scale and Long-term Online Operation. *J. Field Robot.* **2019**, *36*, 416–446. [[CrossRef](#)]
40. Yuan, C.; Liu, M.; Luo, Y.; Chen, C. *Recent Advances in Locality-Sensitive Hashing and Its Performance in Different Applications*; Chengdu University of Technology: Chengdu, China, 2023.
41. Lv, Q.; Josephson, W.; Wang, Z.; Charikar, M.; Li, K. Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search. In Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB), Vienna, Austria, 23–27 September 2007.
42. Liu, S.; Sun, J.; Liu, Z.; Peng, X.; Liu, S. Query-Directed Probing LSH for Cosine Similarity. In Proceedings of the 2016 Fifth International Conference on Network, Communication and Computing (ICNCC 2016), Kyoto, Japan, 17–21 December 2016.
43. Lowe, G.D. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
44. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 2011–2023. [[CrossRef](#)]
45. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.