sensors

MDPI

Article

Loop Closure Detection Based on Residual Network and Capsule Network for Mobile Robot

Xin Zhang [1,2,3,*], Liaomo Zheng [2], Zhenhua Tan [3] and Suo Li [1]

[1] School of Mechanical Engineering, Shenyang Ligong University, Shenyang 110159, China
[2] Shenyang Institute of Computing Technology Co., Ltd., Chinese Academy of Sciences, Shenyang 110168, China
[3] Software College, Northeastern University, Shenyang 110169, China
* Correspondence: zhangxin@sylgdx1.wecom.work; Tel.: +86-186-40218600

Abstract: Loop closure detection based on a residual network (ResNet) and a capsule network (CapsNet) is proposed to address the problems of low accuracy and poor robustness for mobile robot simultaneous localization and mapping (SLAM) in complex scenes. First, the residual network of a feature coding strategy is introduced to extract the shallow geometric features and deep semantic features of images, reduce the amount of image noise information, accelerate the convergence speed of the model, and solve the problems of gradient disappearance and network degradation of deep neural networks. Then, the dynamic routing mechanism of the capsule network is optimized through the entropy peak density, and a vector is used to represent the spatial position relationship between features, which can improve the ability of image feature extraction and expression to optimize the overall performance of networks. Finally, the optimized residual network and capsule network are fused to retain the differences and correlations between features, and the global feature descriptors and feature vectors are combined to calculate the similarity of image features for loop closure detection. The experimental results show that the proposed method can achieve loop closure detection for mobile robots in complex scenes, such as view changes, illumination changes, and dynamic objects, and improve the accuracy and robustness of mobile robot SLAM.

Keywords: simultaneous localization and mapping (SLAM); mobile robot; loop closure detection; residual network (ResNet); capsule network (CapsNet)

Citation: Zhang, X.; Zheng, L.; Tan, Z.; Li, S. Loop Closure Detection Based on Residual Network and Capsule Network for Mobile Robot. *Sensors* **2022**, *22*, 7137. https://doi.org/10.3390/s22197137

Academic Editor: Ramon Barber

Received: 23 May 2022
Accepted: 17 July 2022
Published: 21 September 2022

1. Introduction

Simultaneous localization and mapping (SLAM) is a mobile robot equipped with sensors, which constructs environmental maps by observing unknown environments and realizes simultaneous autonomous localization and navigation [1,2]. SLAM is widely applied in the autonomous navigation of mobile robots, virtual reality, smart homes, and other fields [3,4]. Due to the low cost of visual sensors that can obtain rich scene information, visual SLAM has attracted extensive attention [5]. Loop closure detection is an important part of SLAM, which plays an important role in reducing the accumulated error generated by the visual odometer, improving the accuracy of robot pose estimation, and constructing the global consistency map [6]. With the wide application of SLAM, the problems of low accuracy and the poor robustness of loop closure detection in complex scenarios need to be solved urgently [7].

Loop closure detection has been extensively studied by scholars [8]. Bag of Visual Words (BoVW) is a traditional method to achieve loop closure detection. To represent an image using the BoVW model, an image can be treated as a document. Similarly, "words" in images also need to be defined. Achieving this usually includes three steps: (1) feature extraction, (2) codebook construction, and (3) vector quantization [9,10]. It extracts artificial features such as the scale-invariant feature transform algorithm (SIFT) [11], the speed up robust features algorithm (SURF) [12], and oriented FAST and rotated BRIEF (ORB) [13].

Loop closure detection is realized by measuring the similarity of images [14]. Global characteristics information of a scene (GIST) adopts a two-dimensional filtering method to process regional texture information, extracts the overall features of the image, and improves the efficiency of loop closure detection [15]. However, traditional features are sensitive to environmental changes such as illumination changes, view changes, occlusion, dynamic objects, and a large scale, which affect the accuracy and robustness of loop closure detection [16,17].

Deep learning does not require the manual design of features and has strong feature extraction capability and good robustness to environmental changes. Therefore, deep learning is widely used in face recognition, scene classification, medical diagnosis, and other fields. Hou et al. [18] use deep learning to realize loop closure detection, which improves the accuracy of loop closure detection in illumination change scenarios compared with the BoVW and GIST methods. Sunderhauf et al. [19] used the AlexNet network to extract image features, which showed that mid-level features could better cope with changes in scene appearance and perspective. The Visual Geometry Group of Oxford University proposed the VGG network, which replaced AlexNet's convolution kernel with a continuous convolution kernel, enhanced network performance by increasing network depth, and achieved excellent results [20].

Traditional deep learning methods can extract image features autonomously, but shallow features struggle to accurately describe the rich information of the image, and the spatial details of the image are ignored. The deepening of the number of network layers enlarges the storage space and increases the computation of traditional deep learning methods. The maximum pooling layer of traditional deep learning methods cannot represent the spatial location relationship between features and loses image detail information. Since the input and output of neurons are scalars, traditional deep learning models have a weak ability to represent image features [21].

Wang et al. [22] designed a deep learning model that combines the advantages of ResNet and CapsNet for improving the original structure to effectively classify remotely sensed lidar data. The structure of the ResNet is modified based on ResNet-34, and the outputs of ResNet are sent to CapsNet for lidar classification. Xiang et al. [23] propose a 3-D tumor computer-aided diagnosis (CADx) system with U-net and a residual-capsule neural network (Res-CapsNet) for ABUS images and provide a reference for early tumor diagnosis, especially non-mass lesions. Jampour et al. [24] presented a regularized CapsNet conjugated with ResNet-18 for signature identification. CapsNet allowed a powerful understanding of the objects' components and their positions, while ResNet provided efficient feature extraction and description. The existing Res-CapsNet is applied in the fields of LiDAR data classification, image classification, and signature identification.

Loop closure detection based on a residual network and a capsule network (Res-CapsNet) is proposed to address the problems of low accuracy and poor robustness for mobile robot SLAM in complex scenes such as view changes, illumination changes, weather changes, and dynamic objects. It can improve the accuracy and robustness of the loop closure detection of the SLAM system and realize the autonomous positioning and navigation of mobile robots in complex scenes.

We combine the advantages of ResNet and CapsNet to design Res-CapNet for mobile robot SLAM. The main contributions of this paper are as follows: (1) A pre-trained ResNet model is used as a feature extractor to extract the shallow geometric features and deep semantic features of the image. The residual mechanism and GhostVLAD feature coding method are combined to obtain the global feature descriptors of the image. The GhostVLAD feature coding method can reduce the noise information in the image data and accelerate the convergence speed of the training model. (2) Dynamic routing is optimized by the entropy density peak, and the relative positions and directions between image features are extracted by CapsNet. The parameters are simple and robust to optimize the overall performance of the network. (3) Global feature descriptors and feature vectors are combined to contain the relative location distributions of features, retain the differences and correlations between

features, and improve the accuracy of the SLAM system. Finally, in order to verify the feasibility of the proposed method, loop closure detection and SLAM experiments are designed, and the results are analyzed. The experimental results show that the proposed method is effective and robust.

The paper is organized as follows: In Section 2, we briefly discuss the deep convolutional neural network framework. Section 3 describes the novel architecture of the Res-CapsNet in detail. The experimental results and analysis are discussed in Section 4. Finally, the paper is concluded in Section 5.

2. Related Work

A deep convolutional neural network has the characteristics of local area perception, the up-sampling of the time domain, and weight sharing, which can make great breakthroughs in the recognition and classification of speech, text, image, and video. Network layer deepening enhances the network's learning ability but reduces the convergence speed of the network. Gradient back propagation makes the gradient become infinitesimal, which makes it impossible to effectively adjust the weight of the network. It is difficult to realize reverse gradient conduction, resulting in gradient explosion, gradient disappearance, and large calculations.

In order to solve the problems of gradient disappearance and the network degradation of deep convolutional neural networks, He et al. [25] proposed a residual network (ResNet). It has a simple skip structure and a strong feature extraction capability, which is widely used in face recognition, automatic driving, and image classification. ResNet introduces a residual mechanism and adopts identity mapping to construct a residual unit, which reduces the number of network parameters and the computational complexity, improves the operational efficiency, solves the problem of network degradation, and improves network performance [26]. ResNet includes typical network structures such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. Among them, ResNet-18 and ResNet-34 are composed of basic residual modules, and ResNet-50, ResNet-101, and ResNet-152 are composed of bottleneck modules.

The basic structure of ResNetv2 can be seen in Figure 1. ResNetv2 is composed of a weight, batch normalization (BN), and a nonlinear activation function (ReLU). Assuming that the input of the residual unit l, is x_l, then the output is:

$$x_{l+1} = f(x_l + F(x_l, W_l)) \tag{1}$$

where $F(x_l, W_l)$ is the residual function, the residual function consists of two or three convolution layers, W_l is the weight coefficient corresponding to the residual function, and $f(\cdot)$ is the nonlinear activation function that matches x_l and $F(x_l, W_l)$ to the same dimension by performing linear mapping of W_s.

ResNetv2 model uses a pre-activation mode in backward and forward propagation to make the information propagate faster, allowing the network to obtain better results, and this structure effectively prevents the gradient disappearance problem. Therefore, ResNetv2 was used in this paper.

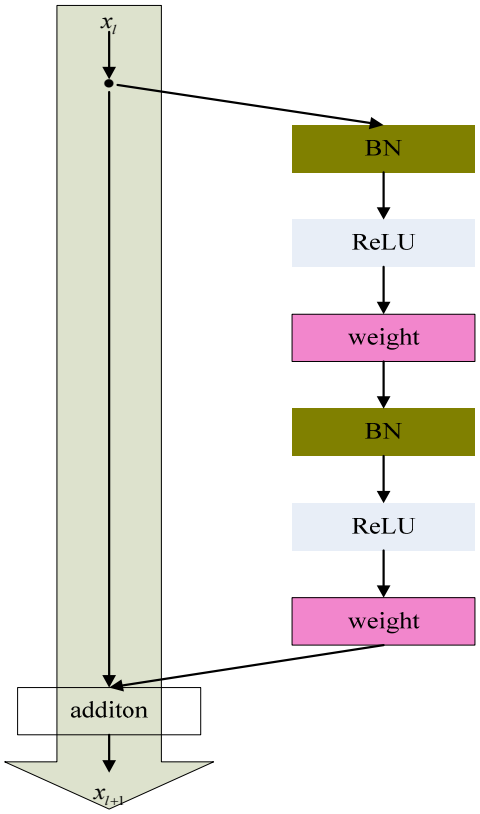


Figure 1. ResNetv2 [27].

3. Proposed Method

In order to improve the extraction and expression ability of image features, avoid the loss of spatial location features, improve the accuracy and robustness of loop closure detection, and realize the autonomous localization and mapping of mobile robots, in this paper a loop closure detection algorithm (Res-CapsNet) combining a deep residual network (ResNet) and a capsule network (CapsNet) is proposed.

3.1. Residual Network Model Based on Feature Coding Strategy

To solve the problems of gradient disappearance, network degradation, and the large amount of computation of deep neural networks and to speed up model convergence in training and meet the real-time requirements of a SLAM system, a residual network model based on a feature coding strategy is proposed in this paper.

Considering the number of model parameters and the training effect comprehensively, the ResNet-50 model is adopted as the basic network of feature extraction, as shown in Figure 2. This model is used to extract shallow geometric features and deep semantic features. Feature coding improves the recognition ability of ResNet by clustering the extracted image features. A vector of locally aggregated descriptors (VLAD) calculates the difference vectors of image feature descriptors and their clustering centers and aggregates local features into global features, which can solve the problem of image retrieval and image classification [28]. Arandjelović et al. [29] obtained global feature descriptions by clustering local features and extracting distribution relations among the features and proposed a VLAD coding algorithm, NetVLAD, combined with a neural network. Compared with the VLAD algorithm, this algorithm is more flexible and suitable for similar scene recognition. In order to extract high-quality image feature descriptors, Arandjelovi et al. [30] proposed the GhostVLAD algorithm by combining NetVLAD and "Ghost" central points, as shown in Figure 3.

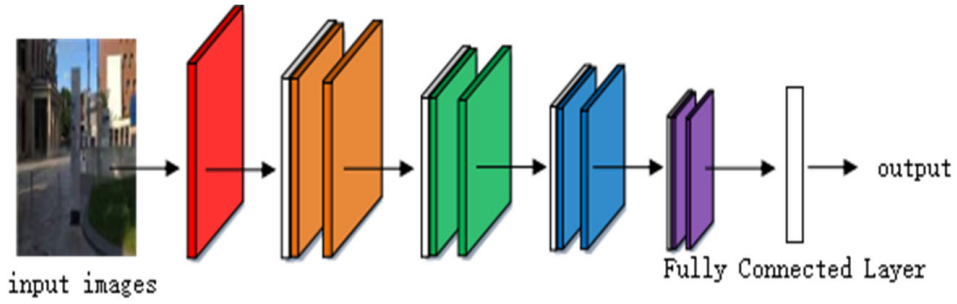


Figure 2. The structure of ResNet-50.

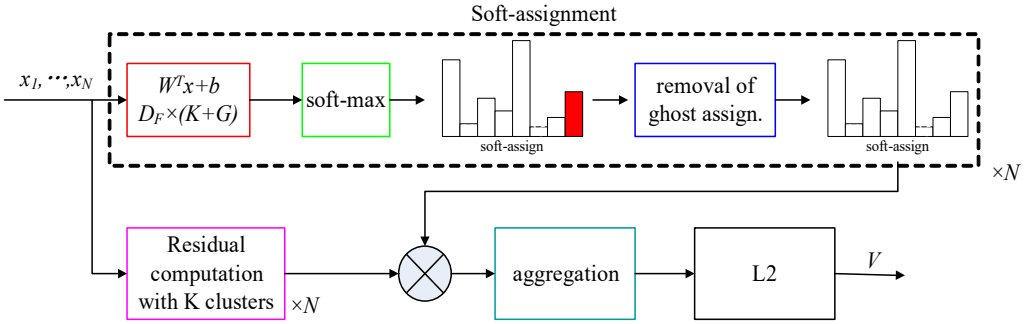


Figure 3. The algorithm flowchart of GhostVLAD.

Given N D-dimensional local image descriptors $\{X_i\}$ as inputs and K cluster centers $\{C_k\}$ as VLAD parameters, the output of VLAD is a $D \times K$ dimensional matrix, V. The element of $V(j,k)$ is computed as follows:

$$V(j,k) = \sum_{i=1}^{N} (x_i(j) - c_k(j)) \tag{2}$$

where $x_i(j)$ is the j-th dimension of the i-th descriptor, and $c_k(j)$ is the j-th dimension of the k-th cluster center.

Due to the different amounts of information contained in the local feature descriptors of each cluster center, we set the weight parameter, $a_k(x_i)$ as the weight of $(x_i(j) - c_k(j))$, which can describe the relationship between the local feature descriptors of each class:

$$V(j,k) = \sum_{i=1}^{N} a_k(x_i)(x_i(j) - c_k(j)) \tag{3}$$

Soft assignments $\overline{a_k}(x_i)$ are replaced with:

$$\overline{a_k}(x_i) = \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}} \tag{4}$$

Then, the global descriptor is:

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}} (x_i(j) - c_k(j)) \tag{5}$$

GhostVLAD is a global descriptor that describes the appearance of input images by adding a ghost clustering center and reduces the weight of low-quality images by automatic weighting. GhostVLAD is a generalization of NetVLAD, as with $G = 0$ the two are equivalent.

The input of the ResNet model is the color image of the real scene, and the image size is 224 × 224 × 3. The last mean pooling layer and full connection layer of ResNet-50 are removed as shown in Figure 4. The GhostVLAD layer is introduced, which distributes noisy information to the ghost classes to reduce the interference effect of noisy data. By training on the fused ResNet network and the GhostVLAD module, the GhostVLAD layer is dimensionally reduced to obtain a 512-dimension output vector, which can reduce the computational burden and effectively improve the robustness of scene recognition.

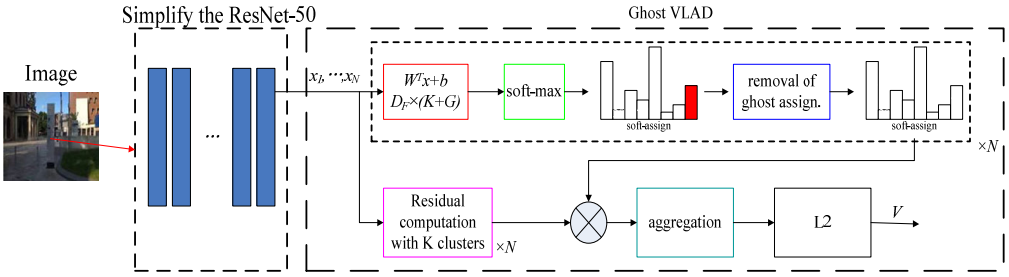


Figure 4. Network structure of ResNet based on GhostVLAD.

3.2. Peak Entropy Density Optimization of Capsule Network

In order to improve the accuracy of convolutional neural networks (CNN) image recognition and retain the spatial position relationship between image features, Hinton et al. [31] proposed the capsule network (CapsNet) for the first time in 2017. The dynamic routing mechanism of CapsNet adopts k-means clustering, which is only suitable for processing spherical data and is sensitive to the initial cluster center. Dynamic routing can be viewed as a parallel attention mechanism that allows each capsule at one level to attend to some active capsules at the level below and to ignore the others. This should allow the model to recognize multiple objects in the image, even if the objects overlap [22].

The optimal truncation distance is solved by optimizing the minimum value of entropy in this paper, and the dynamic routing optimized by density peak is adopted to improve the overall performance of CapsNet. Sabour et al. proposed CapsNet to improve the limitations of CNN feature extraction. By updating the dynamic routing mechanism between the master capsule and the digital capsule, high-level entity representation is obtained, which not only reduces network parameters but also avoids over-fitting. Through the experimental verification of MNIST datasets, compared with CNN, CapsNet has higher classification accuracy in digital recognition, traffic sign recognition, and medical image analysis [32–34].

The CapsNet structure is made of a network of capsules, which are used to represent image features instead of neurons in CNN [35]. Each capsule is a collection of neurons, and multiple capsules make up the entire capsule network. Each capsule represents all or part of the entity, the length of the vector represents the probability of the entity's existence, and the direction of the vector represents various attributes of the entity in the image, such as posture (position, size, and direction), texture, deformation, and color. Dynamic routing is used to replace the maximum or average pooling layer; the output of each capsule is a vector, not a scalar; and vectorized capsules are used to encode feature information [36]. The information transmission process between capsules is shown in Figure 5.

A capsule is the basic operational unit of a capsule network, and each capsule is a collection of neurons. The input vector, s_i, is nonlinearly compressed through capsule i, and the capsule feature vector, v_i, is output as:

$$v_i = \frac{\|s_i\|^2}{1 + \|s_i\|^2} \frac{s_i}{\|s_i\|} \tag{6}$$

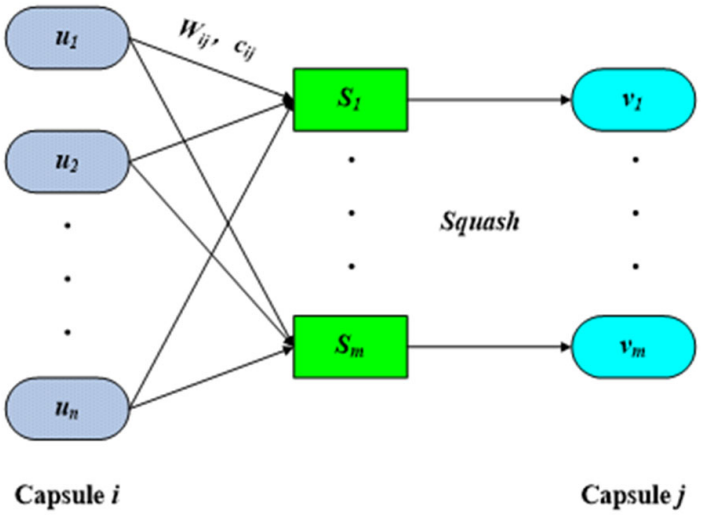


Figure 5. Information transmission between capsules [31].

The length of the capsule's feature vector, v_i, represents the probability of the existence of the target. The dimensional values represent the various attributes of the entity.

First, the capsule feature vector, u_j, of the upper layer in the network is multiplied by the weight matrix, W_{ij}, to generate the intermediate vector, $\widehat{u}_{i|j}$:

$$\widehat{u}_{i|j} = W_{ij} u_j \tag{7}$$

Then, the weighted sum of the intermediate vector, $\widehat{u}_{i|j}$, is used to calculate the input vector, s_i:

$$s_i = \sum_j c_{ij} \widehat{u}_{i|j} \tag{8}$$

b_{ij} is the coupling probability of capsule i and capsule j, and b_{ij} is initially set to zero. The dynamic routing process of CapsNet is the updating process of weighted coefficient c_{ij}:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_i \exp(b_{ij})} \tag{9}$$

Finally, the connection between adjacent capsules is completed to obtain the connection between low-level targets and high-level targets to realize the transmission and expression of characteristic information.

Classical CapsNet consists of an input layer, convolutional layer (Conv1), initial capsule layer (PrimaryCaps), digital capsule layer (DigitalCaps), full connection layer, and output layer, as shown in Figure 6. Compared with the pooling strategy of CNN, the information transfer mechanism of CapsNet fully preserves the spatial position relation between features and realizes the accurate transmission of image information. The weighting coefficient, c_{ij}, is determined by the inner product between the prediction vector, $\widehat{u}_{i|j}$, and the upper capsule, v_i.

The larger the inner product, the larger the weighted coefficient between the capsule neurons, indicating that the lower capsule transmits more characteristic information to the higher capsule. The smaller the inner product, the smaller the weighted coefficient between the capsule neurons, indicating that the lower capsule transmits less characteristic information to the higher capsule. Experiments available through three iterations can improve the coupling coefficient and will not increase the amount of calculation.

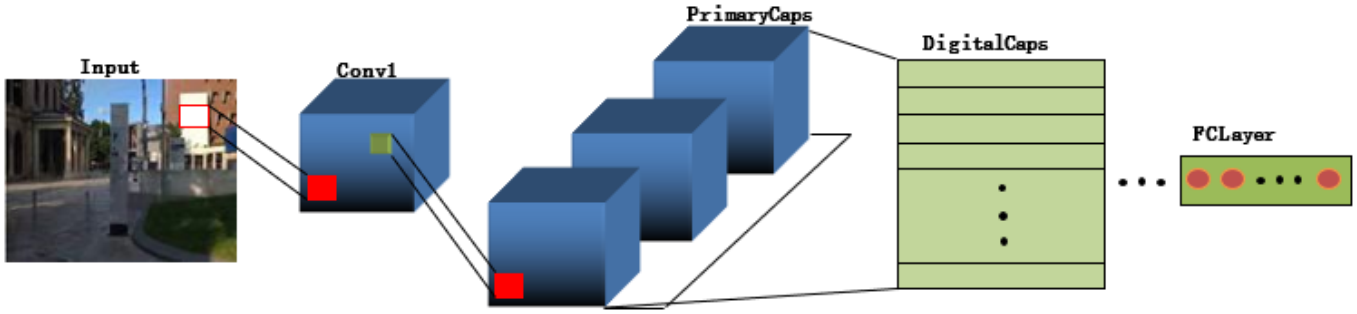


Figure 6. The structure of CapsNet.

The dynamic routing mechanism of the capsule network adopts the k-means clustering algorithm to transform low-level features into high-level features that are only suitable for processing spherical data and are sensitive to the initial clustering center. Rodriguez et al. [37] proposed the density peaks clustering (DPC) algorithm in Science, which is suitable for arbitrary shape data with simple parameters and strong robustness. The dynamic routing mechanism of the capsule network is optimized by the density peak in this paper, and the optimal truncation distance is solved by optimizing the minimum entropy. The sensitivity of the capsule network to the initial cluster center is solved, and the aggregation of low-level features to high-level features is realized. The vector is used to represent the relative positions and directions between features to improve the overall performance of the network.

DPC mainly includes the local density, ρ_i, and the adjacent distance, δ_i. A Gaussian kernel is used to define the local density as:

$$\rho_i = \sum_j \exp\left(-\left(\frac{d_{ij}}{d_c}\right)^2\right) \tag{10}$$

The proximity distance is the minimum distance between data point x_i and a point with a higher density, which can be expressed as:

$$\delta_i = \begin{cases} \min\limits_{j:\rho_j>\rho_i} \{d_{ij}\} \\ \max\{d_{ij}\} \end{cases} \tag{11}$$

Entropy is adopted to optimize truncation distance, and entropy is defined as:

$$H = -\sum_{i=1}^{n} \frac{\rho_i}{Z} \log\left(\frac{\rho_i}{Z}\right) \tag{12}$$

$$Z = \sum_{i=1}^{n} \rho_i \tag{13}$$

where Z is the standardization coefficient. By substituting Equation (10) into Equations (12) and (13), the function of the truncation distance is constructed, and the optimal truncation distance is solved by optimizing the minimum entropy. Experiments show that when entropy is minimal, the optimal value of the truncation distance, d_c, can be obtained. The specific optimization process is as follows:

Step 1. Realize the weight mapping of low-level capsules;

Step 2. The entropy is introduced to determine the truncation distance, d_c;

Step 3. Calculate the local density, ρ_i, and proximity distance, δ_i;

Step 4. Calculate the connection probability between capsules according to the formula $c_{ij} = softmax(b_{ij})$;

Step 5. Calculate the total input, s_j, of the next capsule;

Step 6. Compress s_j to $[0, 1]$, update b_{ij}, and return v_j.

3.3. Fusion of the Optimized Residual Network and Capsule

The residual network features are input into the GhostVLAD layer to obtain the sum of the residual errors of the feature points and clustering centers. The global feature descriptor is obtained by integrating the optimized features. Feature vectors representing feature distribution are obtained by the capsule network, then differentiated features are extracted, and global feature descriptors are combined with feature vectors. ResNet-50 features correspond to the abstract red, yellow, and green blocks in Figure 7. The full connection layer of CapsNet corresponds to the blue block and feature vector in Figure 7. The fusion of the two features includes the relative location distribution between the features, retains the difference and relevance between the features, and improves the accuracy of SLAM system localization and mapping.

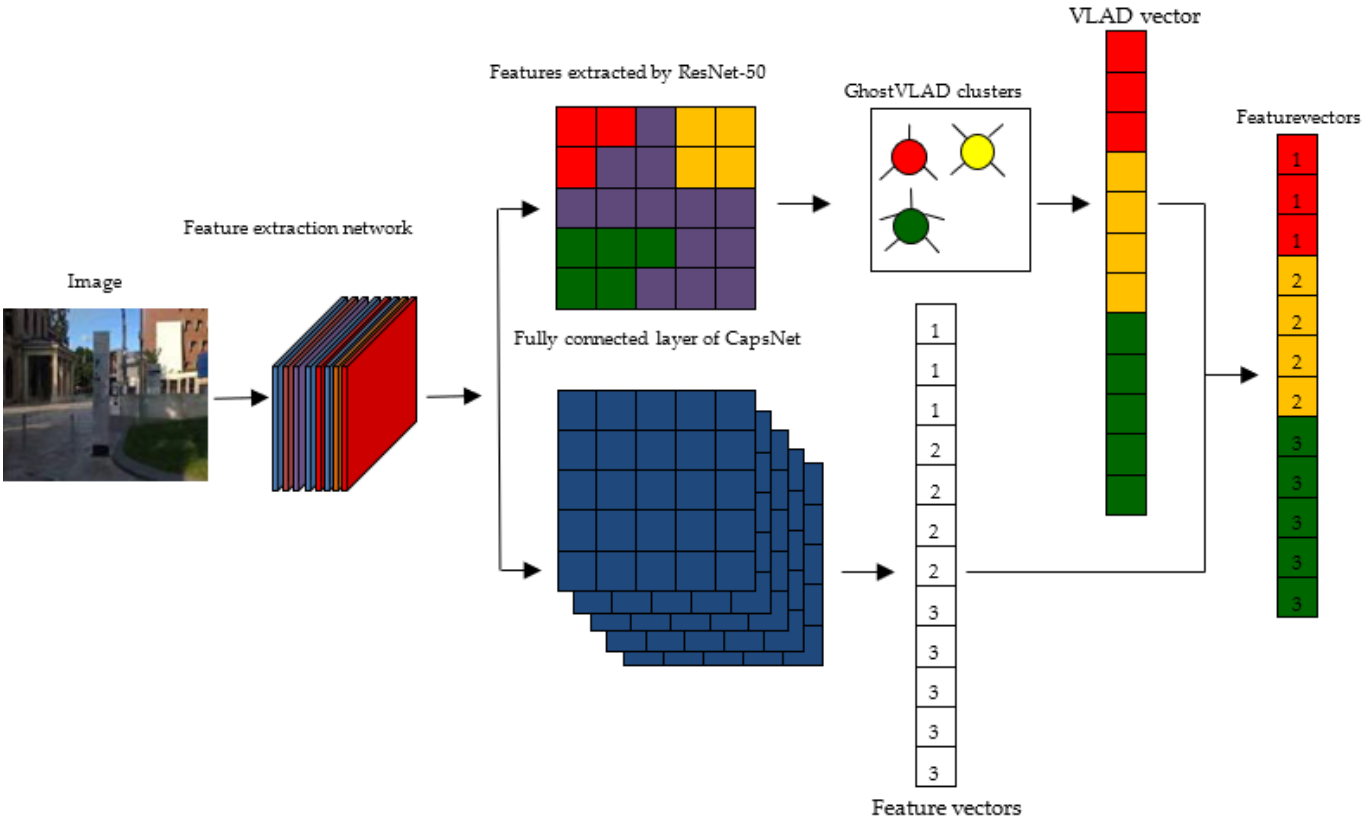


Figure 7. The feature vectors based on the capsule network and the residual network.

L2 normalization and principal component analysis (PCA) are used to reduce the dimensionality of the fused features, as shown in Figure 8. Then, the similarity of the image features is measured to determine whether a closed loop is formed. Res-CapsNet eliminates redundant image features and noise in the data, which not only improves the computational efficiency but also significantly improves the image expression ability, effectively establishes the environmentally consistent map, and improves the accuracy and robustness of SLAM system localization and mapping.

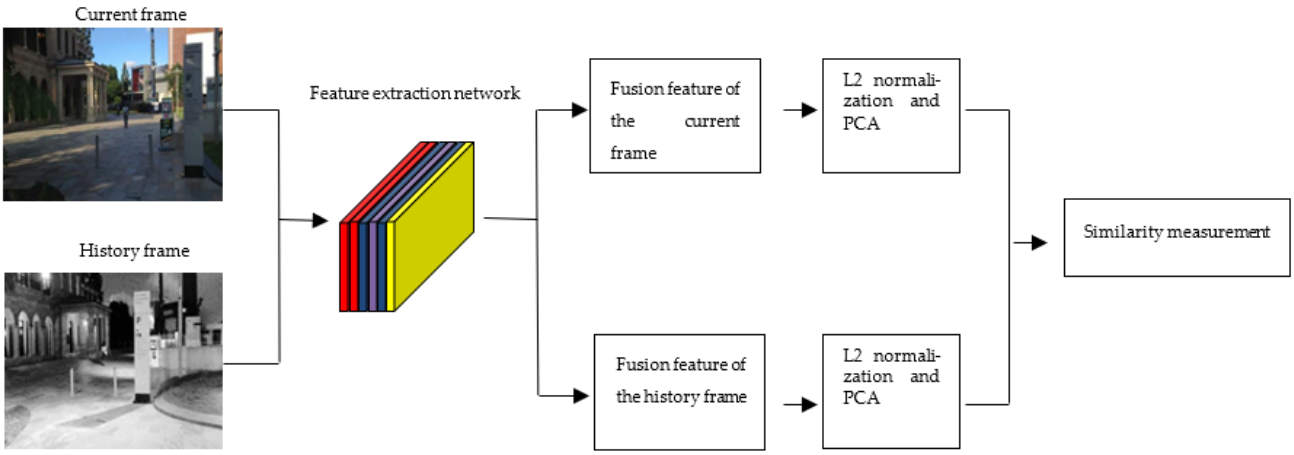


Figure 8. The flowchart of loop closure detection.

4. Experimental Results and Analysis

In order to verify the feasibility of the proposed method (Res-CapsNet), the standard SLAM datasets Gardens Point and TUM were used to evaluate the performance of our approach. Experimental platform: 8G memory and 3.5 GHz CPU.

In this application, the time cost was related to the size of the input data and the training epochs. For training the network, we used KITTI dataset [38] sequences 0–4 with dataset augmentation (approximately 100,000 images). Sequences 9 and 10 were used for validation. We kept a batch size of 5, as higher batch sizes resulted in bigger input tensors and, thus, were difficult to fit in GPU memory.

4.1. Evaluation Index

1. Precision and recall are commonly used indexes to evaluate the effectiveness of loop closure detection. The horizontal axis is the recall rate, the vertical axis is the precision, and the precision–recall curve is used to evaluate the effectiveness of algorithm.

$$precision = \frac{TP}{TP + FP} \tag{14}$$

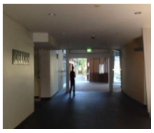
$$recall = \frac{TP}{TP + FN} \tag{15}$$

   where TP represents the correct number of closed loops, FP represents the number of closed loops for error detection, and FN represents the number of true closed loops that are not detected.
2. The area under the curve (AUC) is the main index to evaluate loop closure detection. The closer the AUC value is to 1, the higher the average accuracy of the algorithm.
3. The absolute trajectory error (ATE) is the difference between the estimated trajectory and the real trajectory, which is the main index to evaluate the localization accuracy of SLAM.

4.2. Experimental Results and Analysis of Loop Closure Detection

Gardens Point dataset: the dataset was collected on the campus of Queensland University of Technology, including view changes, illumination changes, dynamic objects, and occlusion factors [39]. An image sample is shown in Table 1. The datasets were composed of three image subsequences. The image subsequences day-left and day-right were collected from the scenes on the left and right sides of the road during the day. Image subsequence night-right was collected from the scene on the right side of the same road at night.

Table 1. The Gardens Point dataset.

Environmental Changes	Compare the Subsequence	Subsequence 1	Subsequence 2
Illumination changes	day-right vs. night-right (Fig.126-GP)		
View changes	day-left vs. day-right (Fig.105-GP)		
Dynamic environment occlusion	day-right vs. night-right (Fig.103-GP)		
Dynamic environment pedestrian	day-right vs. night-right (Fig.56-GP)		

The loop closure detection experiments were performed on the Gardens Point dataset to verify the effectiveness of the proposed method (Res-CapsNet). The dataset contains scene changes such as view, illumination, dynamic objects, and occlusion. The Res-CapsNet method in this paper was compared with the loop closure detection methods based on BoVW, GIST, AlexNet, and VGG. The experimental results are shown in Figures 9–11, where the purple lines denote the visual bag model (BoVW), the red lines denote the loop closure detection based on GIST, the green lines denote the loop closure detection based on AlexNet, and the orange lines denote the loop closure detection based on VGG. The blue lines denote the Res-CapsNet method in this paper.

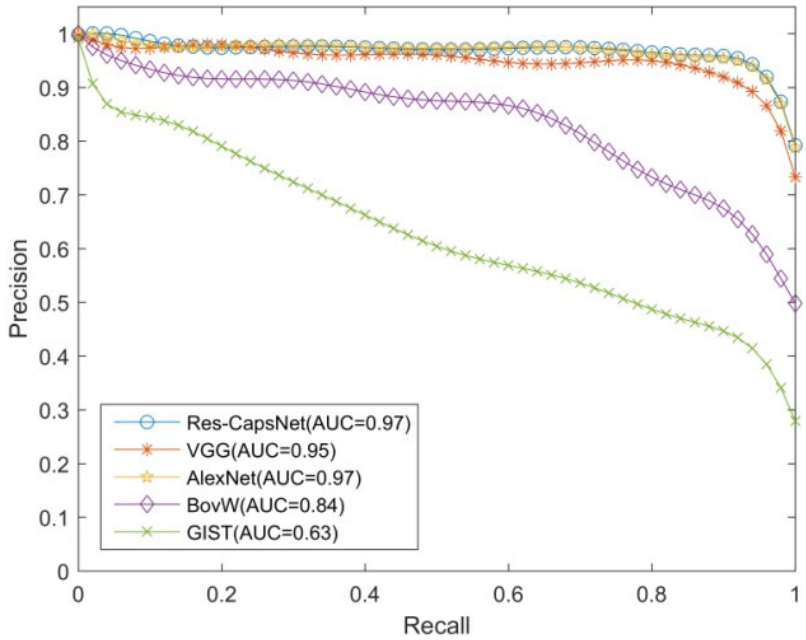


Figure 9. Precision–recall curves of the day-left vs. day-right datasets.

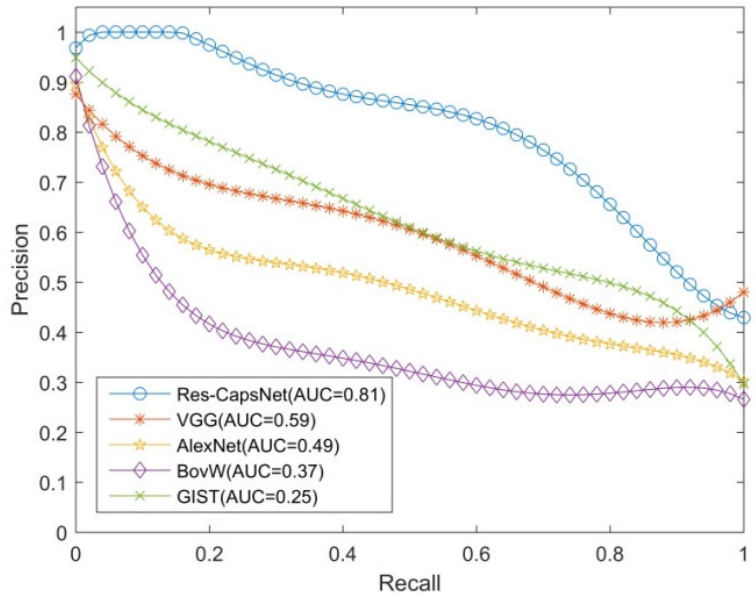


Figure 10. Precision–recall curves of the day-right vs. night-right datasets.

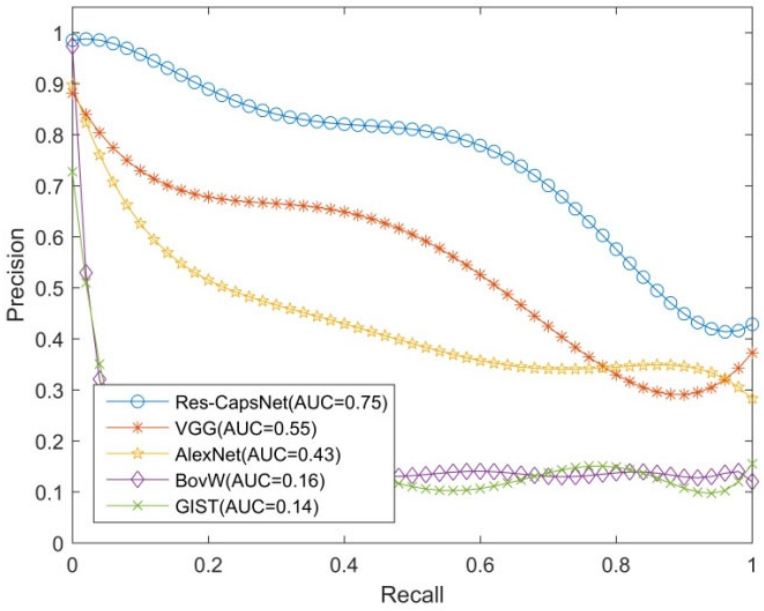


Figure 11. Precision–recall curves of day-left vs. night-right datasets.

Figure 9 shows the loop closure detection experimental results of Gardens Point dataset under the same illumination with a changing perspective, testing the robustness of the proposed method for changing perspective. The AUC of the loop closure detection precision–recall curve based on Res-CapsNet was 0.97, and the average accuracy was also the highest. The AUC values for loop closure detection based on VGG, AlexNet, BoVW, and GIST were 0.95, 0.97, 0.84, and 0.63, respectively. When the recall rate was 80%, the precision of loop closure detection based on Res-CapsNet was 96.49%, while the precision values based on VGG, AlexNet, BoVW, and GIST were 94.97%, 96.18%, 73.28%, and 48.69%, respectively, in the scene with a changing perspective. The effects of loop closure detection based on VGG and AlexNet were similar, and the precision was higher than the loop closure detection based on BoVW and GIST, indicating that feature extraction based on the convolutional neural network model has good robustness for scenes with changing perspectives. The loop closure detection method based on Res-CapsNet maintained a high precision under a high recall rate.

Figure 10 shows the loop closure detection experimental results of the Gardens Point dataset with the same perspective and changing illumination, testing the robustness of the proposed method for changing illumination. The AUC of the loop closure detection precision–recall curve based on Res-CapsNet was 0.81, and the average accuracy was also the highest. The AUC values for loop closure detection based on VGG, AlexNet, BoVW, and GIST were 0.59, 0.49, 0.37 and 0.25, respectively. With the increase in the recall rate, the precision rate decreased gradually. When the recall rate was 80%, the accuracy of the Res-CapsNet method was 65.55%, while the closed-loop detection accuracy values based on VGG, AlexNet, BoVW, and GIST were 43.73%, 37.64%, 27.80%, and 49.89%, respectively. Res-CapsNet had a high precision under a high recall rate.

Figure 11 shows the loop closure detection experimental results of the Gardens Point dataset, which were used to test the robustness of the proposed method in a scenario with illumination changes and view changes. With the illumination changes and view changes of the environment, the performance values of all methods were degraded. The AUC of the loop closure detection precision–recall curve based on Res-CapsNet was 0.75, and the average accuracy was also the highest. The AUC values for loop closure detection based on VGG, AlexNet, BoVW, and GIST were 0.55, 0.43, 0.16 and 0.14, respectively. When the recall rate was 80%, the precision of loop closure detection based on Res-CapsNet was 57.53%, while the precision values based on VGG, AlexNet, BoVW, and GIST were 32.99%, 34.47%, 13.72% and 14.84%, respectively, in the scene with illumination changes and view changes. BoVW and GIST were less robust, suggesting that traditional features are susceptible to changes in illumination and perspective. Due to the image features extracted by the convolutional neural network, spatial details are lost. Therefore, the accuracy values of the AlexNet and VGG closed-loop detection methods were not greatly improved. Under the condition of a high recall rate, the loop closure detection precision based on Res-CapsNet was the highest.

The CMU visual localization dataset consists of multiple visual image sequences [40]. The image sequence was acquired by two monocular cameras mounted on a car. The car drove along the same route in Pittsburgh in different seasons. The image sequences belong to spring, summer, autumn, and winter, respectively, including light, weather, green vegetation, and visual changes produced by dynamic objects, as shown in Figure 12.

Figure 12. The same place during different seasons of a year from the CMU dataset.

The CMU visual localization dataset contains seasonal, weather, light, green vegetation, and visual changes produced by dynamic objects for contrast experiments.

Figure 13 shows the precision–recall curves of the four seasons for the CMU visual localization dataset (spring vs. summer) by PCANet, ResNet-18, ResNet-50, Faster R-CNN, and Res-CapsNet, reflecting the impact of different seasons on loop closure detection. Figure 14 shows the precision–recall curves of the four seasons for the CMU visual localization dataset (summer vs. winter). The weather conditions vary from season to season. Since snow increases the difficulty of identification, summer and autumn are the easiest to match, and autumn and winter are the most difficult to match. Res-CapsNet can maintain a high recall rate with a high precision, and it has better robustness for visual place changes such as seasons.

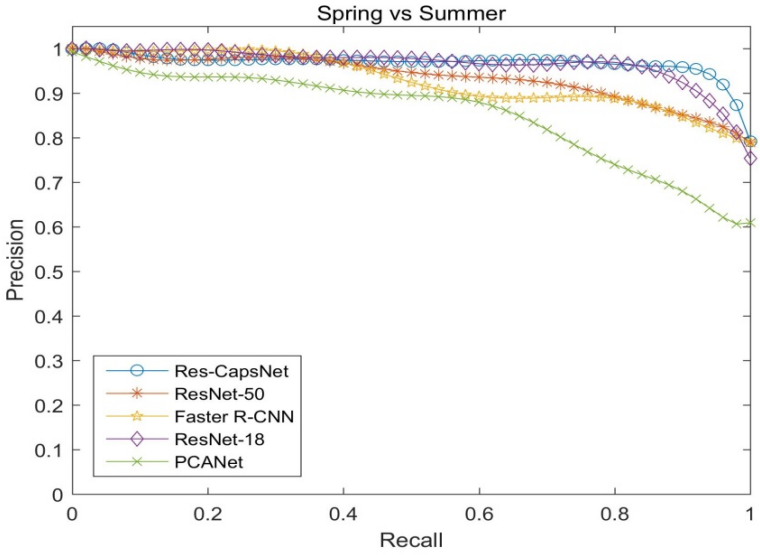


Figure 13. Precision–recall curves of four seasons comparison (spring vs. summer).

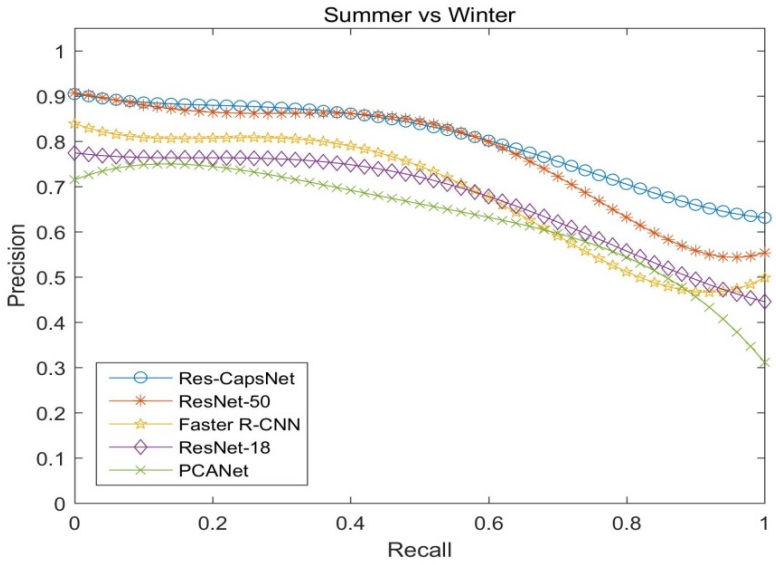


Figure 14. Precision–recall curves of four seasons comparison (summer vs. winter).

4.3. SLAM System Experimental Results and Analysis

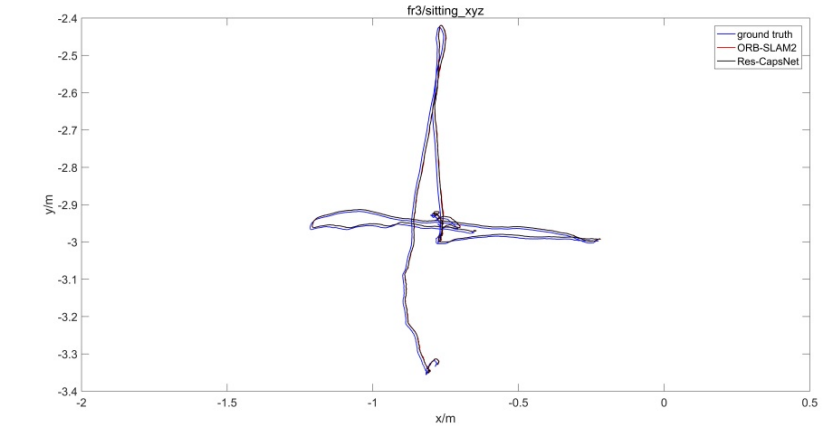
The TUM dataset was collected indoors at the Technical University of Munich, Germany [41]. The dataset was collected by Kinect, including an RGB color map and a depth map, image size 640 × 480, and the real pose trajectory file of the camera. The datasets contain dynamic and large-scale scenes that are targeted to motion blur, rotation, structure, texture, and loop closure situations to meet different testing needs. The parameters are shown in Table 2.

The "sitting" image sequence contains small movements of the human body, which is a low-dynamic scene. The "walking" image sequence contains pedestrians walking dynamically, which is a highly dynamic scene. The "office" image sequence contains the office scene with a track of more than 18 m, which belongs to the large-scale scene. The image was preprocessed, and the color image size was compressed to 224 × 224 × 3 by the scaling function as the input of the feature extraction network (ResNet50), where 224 was the image size and 3 referred to the three RGB channels. The proposed algorithm was compared with the classical ORB SLAM2, and the absolute trajectory error (ATE) was used to evaluate the accuracy of the SLAM system.

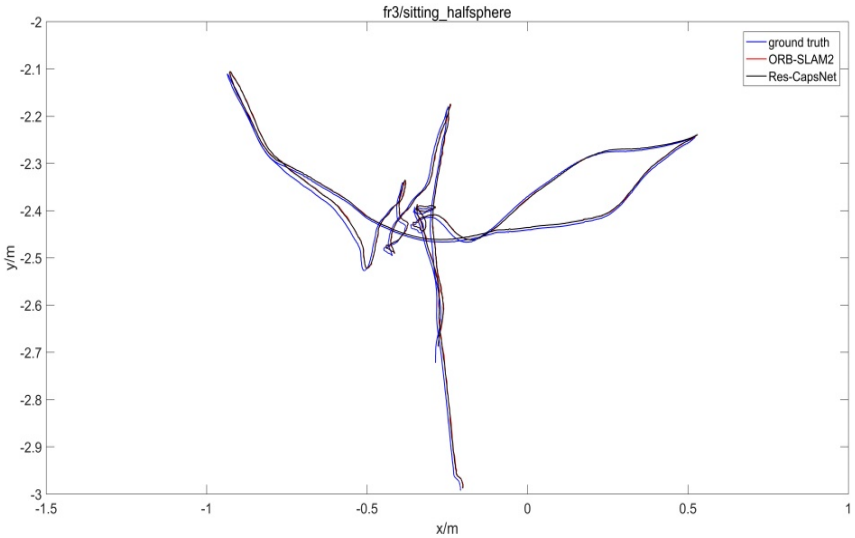
Table 2. The parameters of TUM dataset.

Dataset Attributes	Dataset	Duration (s)	Average Speed of Movement (m/s)	Mean Angular Velocity (deg/s)	True Trajectory Length (m)
Low-dynamic	fr3/sitting-xyz	42.50	0.132	3.562	5.496
	fr3/sitting-halfsphere	37.15	0.180	19.094	6.503
High-dynamic	fr3/walking-xyz	28.83	0.208	5.490	5.791
	fr3/walking-halfsphere	35.81	0.221	18.267	7.686
Large-scale	fr3/long-office	87.09	0.249	10.188	21.455
	fr2/desk	99.36	0.193	6.338	18.880

The TUM dataset provided the actual camera pose, and the accuracy of pose estimation for the SLAM system was evaluated by comparing the estimated pose with the real pose. Figure 15 is the comparison of the estimated trajectory and real trajectory for the TUM dataset. The black curve denotes the real trajectory, the red curve denotes the ORB-SLAM2-estimated trajectory, and the blue curve denotes the Res-CapsNet-estimated trajectory.

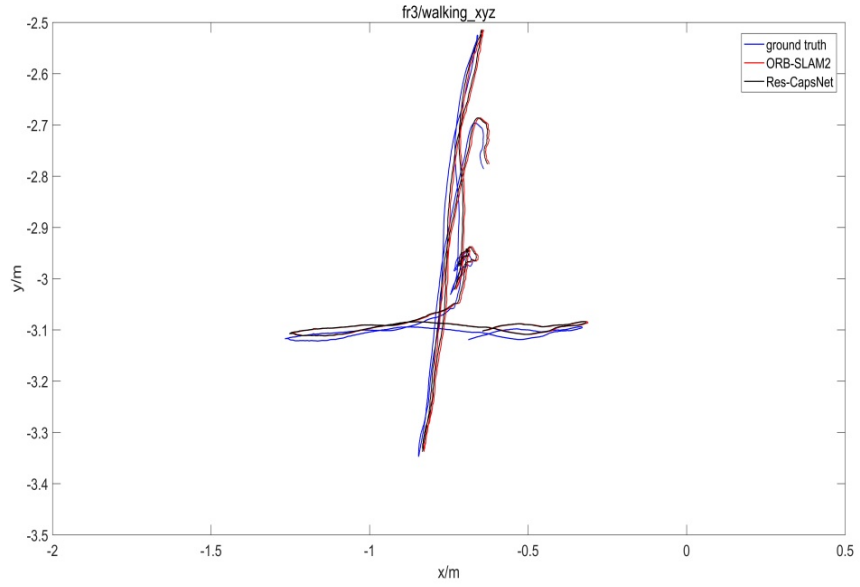


(**a**)

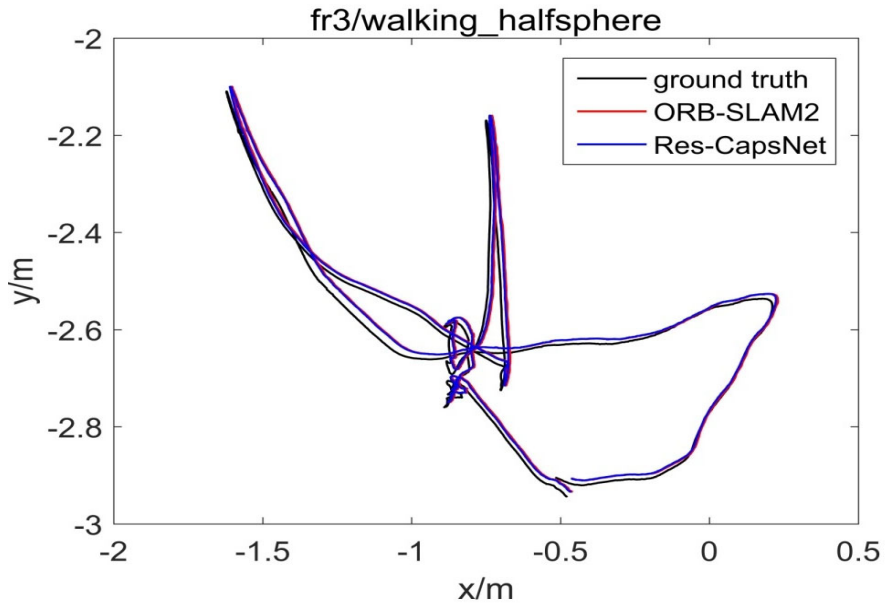


(**b**)

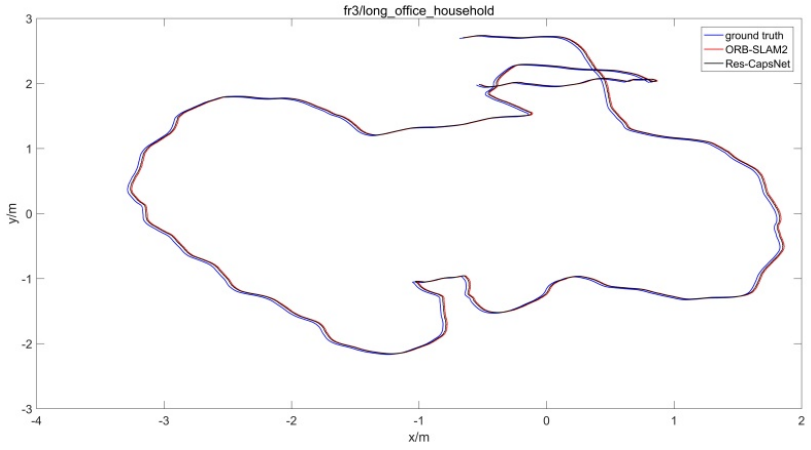
Figure 15. *Cont.*



(**c**)

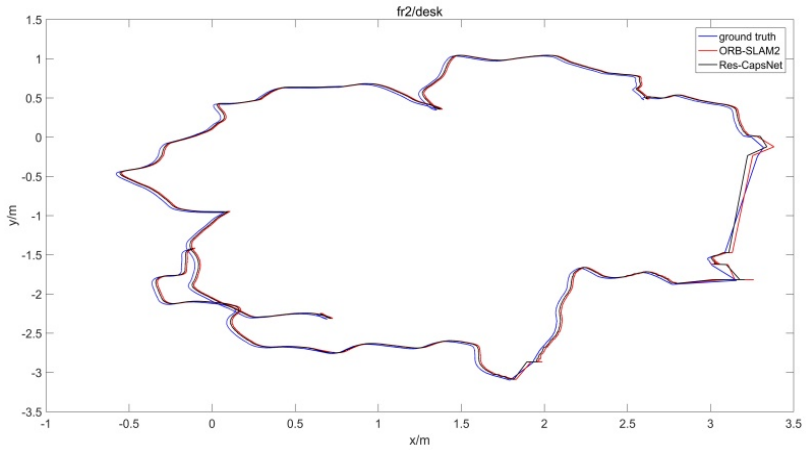


(**d**)



(**e**)

Figure 15. *Cont*.



(**f**)

Figure 15. Absolute trajectory error comparison for SLAM. (**a**) fr3/sitting-xyz, (**b**) fr3/sitting-halfsphere, (**c**) fr3/walking-xyz, (**d**) fr3/walking-halfsphere, (**e**) fr3/long-office, (**f**) fr2/desk.

Figure 15 shows the SLAM system trajectory comparison results of Res-CapsNet and ORB-SLAM2 in low-dynamic, high-dynamic, and large-scale scenarios. The trajectory estimation results of the Res-CapsNet and classical ORB-SLAM2 methods were close to the real trajectory in the low-dynamic scenarios, as shown in Figure 15a,b. The results show that both Res-CapsNet and ORB-SLAM2 have good localization accuracy in low-dynamic scenarios. ORB-SLAM2 has a large error in trajectory estimation in a high-dynamic scene, as shown in Figure 15c,d. Under the condition of violent camera shaking or rapid movement, ORB-SLAM2 cannot accurately distinguish static or dynamic features in the scene, and the accuracy of pose estimation is reduced due to the influence of dynamic features. The estimated trajectory of Res-CapsNet was closer to the real trajectory, with a higher accuracy. Compared with ORB-SLAM2, Res-CapsNet maintained a higher accuracy in large-scale scenarios, as shown in Figure 15e,f. To sum up, Res-CapsNet maintains high accuracy and robustness in complex scenarios.

Table 3 shows the absolute trajectory error (ATE) results of ORB-SLAM2 and Res-CapsNet between the estimated trajectory and the real trajectory in the TUM dataset. Since the RANSAC algorithm of ORB-SLAM2 can eliminate the interference of outside-point motion in low-dynamic scenes, the accuracy of ORB-SLAM2 and Res-CapsNet were similar, and the performance of SLAM was not significantly improved. Compared with ORB-SLAM2, the ATE of Res-CapsNet decreased significantly in high-dynamic and large-scale scenarios. The Res-CapsNet method improved performance by 72.68%, 60.73%, 20.88%, and 27.91%, respectively, in the fr3/walking-xyz, fr3/walking-halfsphere, fr3/long-office, and fr2/desk sequences. This shows that the SLAM based on Res-CapsNet has higher localization accuracy and better robustness in complex scenarios.

Table 3. Comparison of absolute trajectory error.

Sequence	ORB-SLAM2/(cm)	Our Method/(cm)	Performance Improvement
fr3/sitting-xyz	0.95	0.87	8.42%
fr3/sitting-halfsphere	7.75	6.62	14.58%
fr3/walking-xyz	73.80	20.16	72.68%
fr3/walking-halfsphere	49.93	19.61	60.73%
fr3/long-office	10.44	8.26	20.88%
fr2/desk	0.86	0.62	27.91%

Table 4 shows the time consumption of the feature extraction algorithms. The feature extraction time of Res-CpasNet was lower than the ResNet and VGG16 methods and higher

than the AlexNet and Faster R-CNN methods. The real-time performance of Res-CapsNet can guarantee the real-time requirement of the SLAM system in complex scenarios.

Table 4. Time consumption of feature extraction algorithms.

Feature Extraction Algorithms	Network Models	Dataset Tasks	Calculation Rate (ms/Frame)
AlexNet-CONV3	AlexNet [42]	ImageNet classification [44]	167
AlexNet-POOL5			147
VGG-M1024	VGG-M1024 [43]		249
VGG16	VGG16 [20]		1379
FastRCNN-CaffeNet	Fast R-CNN [45]	PASCAL VOC	23
FastRCNN-VGG-M1024			36
FastRCNN-VGG			128
Faster RCNN-ZF	Faster R-CNN [46]	Object detection [47]	49
Faster RCNN- VGG16			157
SqueezeNet	SqueezeNet [48]	ImageNet classification [44]	245
ResNet	ResNet [25]		1517
Res-CapsNet	Res-CapsNet	SLAM [41]	256

5. Conclusions

We proposed a loop closure detection method based on optimized ResNet and CapsNet. ResNet was used to extract the deep features of images, and GhostVLAD feature coding was introduced to achieve image feature clustering, which solves the problems of network gradient disappearance and network degradation and improves the network convergence speed. The optimal truncation distance was solved by optimizing the minimum value of entropy, the dynamic routing mechanism of the capsule network was improved by using the peak value of entropy density, and the relative spatial location information between features was extracted. Combined with global feature descriptors and feature vectors extracted from CapsNet, the deep network's ability to recognize and describe image features was improved, and the differences and correlations among features were retained, thus improving the overall performance of the network. The experimental results show that the average accuracy of Res-CapsNet is the highest, which effectively realizes the loop closure detection of a mobile robot in complex scenes, such as illumination changes, view changes, weather changes, and dynamic and large-scale scenes; reduces the cumulative error of the visual odometer; realizes the establishment of a global consistent environment map; and improves the accuracy and robustness of mobile robot SLAM.

Author Contributions: Conceptualization, X.Z. and L.Z.; Methodology, X.Z. and Z.T.; Software and validation, X.Z. and S.L.; Writing—Original draft preparation, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Basic Scientific Research Project of Colleges and Universities from the Educational Department of Liaoning Province (LJKZ0258); Liaoning Doctor Scientific Research Initial Fund in 2022 from Department of Science & Technology of Liaoning Province (2022-BS-187); Research support project for introducing high-level talents of Shenyang Ligong University (1010147001012); Research and innovation team building project of Shenyang Ligong University (SYLUTD202106).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments on the paper and the builders of the datasets.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localizationand Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
2. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [CrossRef] [PubMed]
3. Schneider, T.; Dymczyk, M.; Fehr, M.; Egger, K.; Lynen, S.; Gilitschenski, I.; Siegwart, R. Maplab: An open framework for research in visual-inertial mapping and localization. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1418–1425. [CrossRef]
4. Lee, S.H.; Civera, J. Loosely-coupled semi-direct monocular slam. *IEEE Robot. Autom. Lett.* **2019**, *44*, 399–406. [CrossRef]
5. Muoz-Salinas, R.; Medina-Carnicer, R. UcoSLAM:Simultaneous Localization and Mapping by Fusion of KeyPoints and Squared Planar Markers. *Pattern Recognit.* **2020**, *101*, 107193. [CrossRef]
6. Guclu, O.; Can, A. Integrating global and local image features for enhanced loop closure detection in RGB-D SLAM systems. *Vis. Comput.* **2019**, *36*, 1271–1290. [CrossRef]
7. Kuo, J.; Muglikar, M.; Scaramuzza, D. Redesigning SLAM for Arbitrary Multi-CameraSystems. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation, Paris, France, 31 May–31 August 2020; pp. 2116–2122.
8. 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**. [CrossRef]
9. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
10. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2:An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
11. Lowe, D. Distinctive image features from scale-invariant key points. *Int. J. Comput. Vis.* **2003**, *20*, 91–110.
12. Bay, H.; Tuytelaars, T.; Gool, L.V. SURF: Speeded up robust features. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
13. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
14. Neubert, P.; Schubert, S.; Protzel, P. Resolving Place Recognition Inconsistencies Using Intra-Set Similarities. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2084–2090. [CrossRef]
15. Oliva, A.; Torralba, A. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [CrossRef]
16. Labbe, M.; Michaud, F. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Trans. Robot.* **2013**, *29*, 734–745. [CrossRef]
17. Gao, X.; Zhang, T. Unsupervised learning to detect loops using deep neural networks for visual SLAM system. *Auton. Robot.* **2017**, *41*, 1–18. [CrossRef]
18. Hou, Y.; Zhang, H.; Zhou, S. BoCNF:efficient image matching with Bag of ConvNet features for scalable and robust visual place recognition. *Auton. Robot.* **2017**, *42*, 1–17.
19. Sünderhauf, N.; Shirazi, S.; Dayoub, F.; Upcroft, B.; Milford, M. On the performance of convnet features for place recognition. In Proceedings of the 2015 IEEE Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 4297–4304.
20. Simonyan, K.; Zisserman, A. Very Deep convolutional networks for large-scale image recognition. *Comput. Vis. Pattern Recognit.* **2014**, *6*, 1–14.
21. Radwan, N.; Valada, A.; Burgard, W. Vlocnet++: Deep multitask learning for semantic visuallocalization and odometry. *IEEE Robot. Autom. Lett.* **2018**, *3*, 4407–4414. [CrossRef]
22. Wang, A.; Wang, M.; Wu, H.; Jiang, K.; Iwahori, Y. A novel LiDAR data classification algorithm combined capsnet with resnet. *Sensors* **2020**, *20*, 1151. [CrossRef]
23. Xiang, H.; Huang, Y.S.; Lee, C.H.; Chien, T.Y.C.; Lee, C.K.; Liu, L.; Li, A.; Lin, X.; Chang, R.F. 3-D Res-CapsNet convolutional neural network on automated breast ultrasound tumor diagnosis. *Eur. J. Radiol.* **2021**, *138*, 109608. [CrossRef]
24. Jampour, M.; Abbaasi, S.; Javidi, M. CapsNet regularization and its conjugation with ResNet for signature identification. *Pattern Recognit.* **2021**, *120*, 107851. [CrossRef]
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
26. Rajpal, S.; Lakhyani, N.; Singh, A.K.; Kohli, R.; Kumar, N. Using Handpicked Features in Conjunction with ResNet-50 for Improved Detection of COVID-19 from Chest X-ray Images. *Chaos Solitons Fractals* **2021**, *145*, 1–9. [CrossRef]
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 630–645.
28. Jégou, H.; Perronnin, F.; Douze, M.; Sánchez, J.; Pérez, P.; Schmid, C. Aggregating Local Image Descriptors into Compact Codes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1704–1716. [CrossRef]
29. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1437–1451. [CrossRef]

30. Zhong, Y.; Arandjelovi, R.; Zisserman, A. GhostVLAD for set-based face recognition. In Proceedings of the Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; pp. 35–50.
31. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. In Proceedings of the 31st International Conference on Neural Information Processing Systems, New York, NY, USA, 4–9 December 2017; pp. 3859–3869.
32. Zhou, S.; Zhou, Y.; Liu, B. Using Siamese Capsule Networks for Remote Sensing Scene Classification. *Remote Sens. Lett.* **2020**, *11*, 757–766. [CrossRef]
33. Abra Ayidzoe, M.; Yu, Y.; Mensah, P.K.; Cai, J.; Adu, K.; Tang, Y. Gabor capsule network with preprocessing blocks for the recognition of complex images. *Mach. Vis. Appl.* **2021**, *32*, 91. [CrossRef]
34. Chang, S.; Liu, J. Multi-lane capsule network for classifying images with complex background. *IEEE Access* **2020**, *8*, 79876–79886. [CrossRef]
35. Huang, R.; Li, J.; Wang, S.; Li, G.; Li, W. A robust weight-shared capsule network for intelligent machinery fault diagnosis. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6466–6475. [CrossRef]
36. Paoletti, M.E.; Haut, J.M.; Fernandez-Beltran, R.; Plaza, J.; Plaza, A.; Li, J.; Pla, F. Capsule networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 2145–2160. [CrossRef]
37. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [CrossRef] [PubMed]
38. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
39. Glover, A. Gardens Point Walking Dataset. 2014. Available online: https://wiki.qut.edu.au/display/cyphy/Open+datasets+and+software (accessed on 9 March 2014).
40. Hernan, B.; Daniel, H.; Takeo, K. The CMU Visual Localization Dataset. 2011. Available online: http://3dvis.ri.cmu.edu/datasets/localization (accessed on 28 July 2011).
41. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, Vilamoura, Portugal, 7–12 October 2012; pp. 573–580.
42. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, SN, USA, 3–6 December 2012; pp. 1097–1105.
43. Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Return of the Devilin the Details: Delving Deep into Convolutional Nets. In Proceedings of the British Machine Vision Conference (BMVC), Nottingham, UK, 1–5 September 2014.
44. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
45. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
46. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montreal, QC, USA, 7–12 December 2015; pp. 91–99.
47. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
48. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.