International Journal of Computational Intelligence Systems (2023) 16:49
https://doi.org/10.1007/s44196-023-00223-8

RESEARCH ARTICLE

A Lightweight Neural Network for Loop Closure Detection in Indoor Visual SLAM

Deyang Zhou[1] · Yazhe Luo[2] · Qinhan Zhang[1] · Ying Xu[2] · Diansheng Chen[2] · Xiaochuan Zhang[1]

Received: 8 December 2022 / Accepted: 19 March 2023

Abstract

Loop closure detection (LCD) plays an important role in visual simultaneous location and mapping (SLAM), as it can effectively reduce the cumulative errors of the SLAM system after a long period of movement. Convolutional neural networks (CNNs) have a significant advantage in image similarity comparison, and researchers have achieved good results by incorporating CNNs into LCD. The LCD based on CNN is more robust than traditional methods. As the deep neural network frameworks from AlexNet and VGG to ResNet have become smaller while maintaining good accuracy, indoor LCD does not need robots to finish a large number of complex processing operations. To reduce the complexity of deep neural networks, this paper presents a new lightweight neural network based on MobileNet V2. We propose a strategy to use Efficient Channel Attention (ECA) to insert into Compressed MobileNet V2 (ECMobileNet) for reducing operands while maintaining precision. A corresponding loop detection method is designed based on the average distribution of ECMobileNet feature vectors combined with Euclidean distance matching. We used TUM datasets to evaluate the results, and the experimental results show that this method outperforms the state-of-the-art methods. Although the model was trained only on the indoorCVPR dataset, it also demonstrated superior performance on the TUM datasets. In particular, the proposed approach is more lightweight and highly efficient than the current existing neural network approaches. Finally, we used TUM datasets to test LCD based on ECMobileNet in PTAM, and the experimental results show that this lightweight neural network is feasible.

Keywords LCD · SLAM · ECA · MobileNet · Indoor · Lightweight neural network

Abbreviations

LCD	Loop closure detection
SLAM	Simultaneous location and mapping
ECA	Efficient channel attention
BoW	Bag of words
SE	SEnet
P–R	Precision–recall
GAP	Global average pooling
PTAM	Parallel tracking and mapping
CNN	Convolutional neural networks
ECMobileNet	Efficient Channel attention + compressed MobileNet V2
Dwise	Depthwise

✉ Xiaochuan Zhang
zxc@cqut.edu.cn

Deyang Zhou
2677806470@qq.com

Yazhe Luo
by2007124@buaa.edu.cn

Qinhan Zhang
1061327503@qq.com

Ying Xu
xuying@buaa.edu.cn

Diansheng Chen
chends@163.com

[1] School of Artificial Intelligence, Chongqing University of Technology, Chongqing 401135, China

[2] School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China

1 Introduction

With the development of indoor service robots, an increasing number of indoor robots are being introduced into our lives. Due to the low cost of visual sensors that can obtain rich scene information, visual SLAM has garnered significant attention [1]. As an essential component of indoor device robots, visual SLAM [2] not only assists the robots in navigation but also aids in obstacle avoidance [3]. LCD in SLAM [4, 5] enables a robot or observing agent to identify that it

Published online: 05 April 2023

has returned to a previously visited area [6]. Therefore, LCD is crucial for maintaining the accuracy of a machine's position during exploration. When the robot returns to its starting point after a period of movement, its position may not coincide with its original position due to posture drift. Using LCD, the estimated position can be recalibrated, effectively eliminating drift and minimizing cumulative errors, thus enabling the creation of a more accurate map [7], as illustrated in Fig. 1.

An important aspect of improving visual LCD performance is to obtain effective scene descriptions based on the input images [8]. Traditional LCD methods rely on handcrafted feature descriptors for image matching. Lowe et al. [9] designed SIFT feature which has become a commonly used image feature descriptor in the field of computer vision, but the calculation process is very time-consuming and cannot meet the real-time requirements of SLAM. Rublee et al. [10] on SIFT features improved and put forward the ORB feature. Bay et al. [11] proposed SURF feature, a feature of accelerated robustness, which greatly improves the computing speed while appropriately reducing robustness. The BoW model applied initially to text retrieval [12], and the study found that it can also recognize images. In other words, building vocabulary's visual words [13]. The BoW extracts image features by SIFT and ORB, while it belongs of traditional feature description. However, the common feature of traditional feature description is that it is unable to efficiently extract various dynamic changes in the scene, resulting in the feature extraction is not obvious or wrong. Furthermore, the size of vocabulary in BoW requires more memory [14], which is not very realistic for indoor robotics.

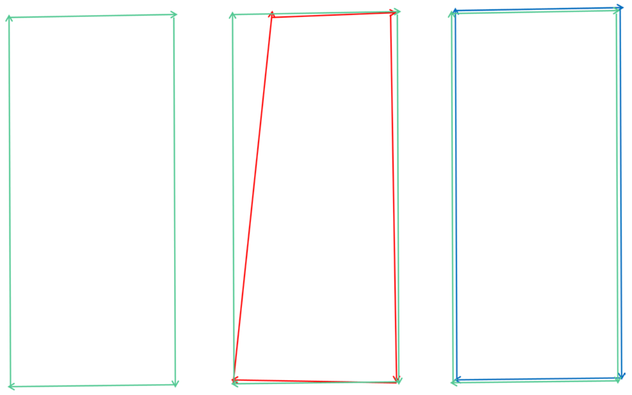
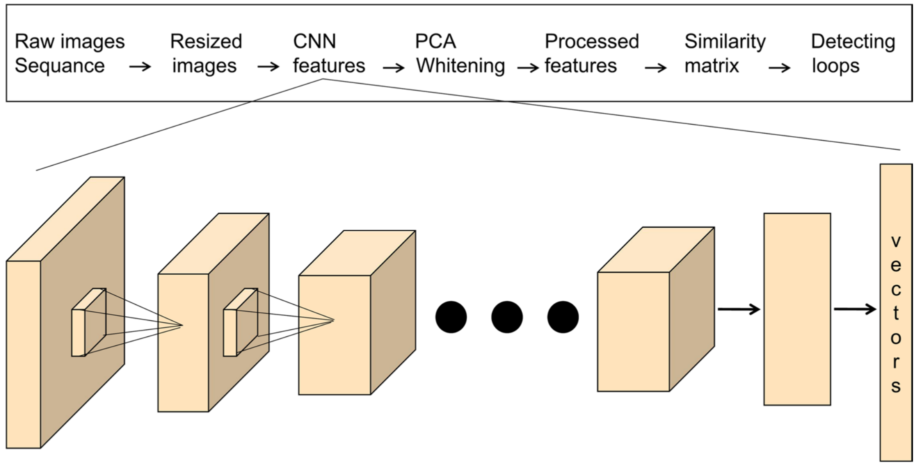


Fig. 1 The green line is the true trajectory. The red line is the drifting trajectory. The blue line indicates that the machine is back to true trajectory by LCD

In recent years, deep learning has become increasingly popular for extracting image features. Deep neural networks are particularly effective at image recognition and classification [15]. The CNN features are demonstrated which have stronger robustness to counter viewpoint changes, light conditions, and scale variations, and can effectively solve the traditional method's shortcomings [16]. Zhang X [16] has proposed a general algorithm for inserting convolutional networks into LCD, as shown in Fig. 2. This algorithm extracts image feature vectors using CNNs and compares the similarity of images to achieve LCD. Although the current common VGG and ResNet solve the traditional method's defects, the SLAM adds huge computational operations [17]. As a result, real-time SLAM with deep neural networks is not practical in many real-world scenarios. Lightweight neural networks, such as small CNNs, have recently been successfully applied in many areas. These models maintain high accuracy while using very few parameters. For example, Google's MobileNet V1 [18], proposed in 2016 and published in 2017, uses deep separable convolution and a stack of deep separable modules to achieve high accuracy on mobile devices. MobileNet V2 [19] and MobileNet V3 [20] were subsequently proposed with smaller parameters and higher accuracy. Similarly, the ShuffleNet model [21] is extremely lightweight and computationally efficient. Like MobileNet and SqueezeNet, ShuffleNet is primarily intended

Fig. 2 The image is using CNN to detect loops



for use on mobile devices. Using lightweight neural networks can reduce computational overhead and improve real-time performance in LCD.

In this paper, we propose a new lightweight neural network model ECMobileNet based on MobileNet V2 that can be integrated into LCD. We use Efficient Channel Attention (ECA) to insert MobileNet's bottlenecks and delete some bottlenecks to reduce the number of parameters. First, we train the ECMobileNet using the indoorCVPR dataset and achieve the desired results. We then use the public dataset images as input to the ECMobileNet to obtain image feature vectors. Second, we reduce the vector size and use the Euclidean Distance algorithm to calculate similarity scores for comparing images and determining whether they belong to LCD. Third, we demonstrate the feasibility of this algorithm by comparing it to the current mainstream algorithm in several aspects, including the number of parameters and the PR curve. Finally, we integrate this LCD algorithm into PTAM to verify its effectiveness using the TUM RGB-D dataset. Compared to the motion trajectory obtained using PTAM alone, the motion trajectory obtained using this method is smoother and more consistent with the real trajectory, which can effectively reduce the cumulative error of the entire system. The main contributions of this paper are summarized as follows:

(1) A new lightweight neural network model ECMobileNet is proposed.
(2) Using the specific small indoor dataset also can make get good results.
(3) We demonstrate that our approach achieves competitive recall rates at 100% precision when compared to state-of-the-art methods using four challenging public image sequences.
(4) We show that this LCD algorithm is feasible in the SLAM system.

The remainder of the paper is organized as follows. Section 2 summarizes related work in LCD. Section 3 describes the proposed approach. Section 4 presents experimental results. Finally, Sect. 5 concludes the study.

2 Related Work

2.1 Efficient Channel Attention

Qilong Wang et al. [22] propose an ECA module for deep CNN, which avoids dimension reduction and effectively captures information from local cross-channel interaction. The ECA module is shown in Fig. 3.

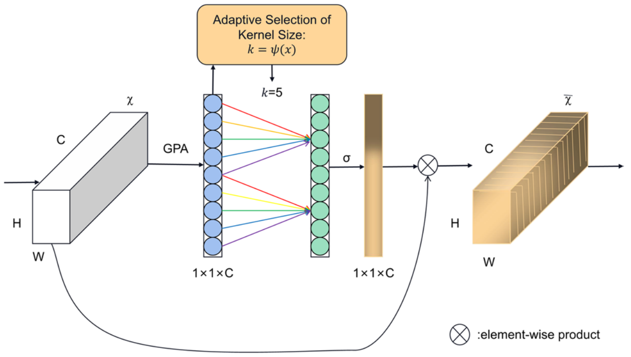


Fig. 3 Diagram of this efficient channel attention module. Given the aggregated features obtained by global average pooling (GAP), ECA generates channel weights by performing a fast 1D convolution of size k, where k is adaptively determined via a mapping of channel dimension C

To understand ECA, it is necessary to first be familiar with SENet [23] (SE). The author of ECA has experimentally evaluated the effects of dimensionality reduction and non-linear cross-channel interaction in the SE block, which motivated the proposal of the ECA module. Additionally, the author has developed a method for adaptively determining ECA parameters and ultimately demonstrates how it can be used with deep CNNs. The weights of channels in SE block can be computed as Eq. (1), where $g(\chi) = \frac{1}{WH}\sum_{i=1,j=1}^{W,H}\chi_{ij}$ is channel-wise global average pooling (GAP) and σ is a Sigmoid function

$$\omega = \sigma(f\{W_1, W_2\}(g(\chi)). \tag{1}$$

The ECA modules are described starting with Eq. (2). To verify their effectiveness, the authors compared the original SE block with three variants (SE-Var1, SE-Var2, and SE-Var3), all of which were not subjected to the dimensionality reduction operation. According to the paper, SE-Var1 with no parameter was still superior to the original network, indicating that channel attention can improve the performance of deep CNN. In addition, SE-Var2 learns the weight of each channel independently, which is slightly superior to the SE block while involving fewer parameters. This may indicate that channels and their weights need to correspond directly while avoiding dual sensitivity reduction, which is more important than considering non-linear channel dependence. Furthermore, SE-Var-3, which uses a single FC layer, performs better than two FC layers with dimensionality reduction in the SE block. All of the above results clearly show that avoiding dimensionality reduction helps to learn effective channel attention and improve accuracy. Later, the authors combined the advantages of SEVar-2 and SEVar-3 and proposed a new local cross-channel interaction method. However, the results were not satisfactory, and therefore, another method for local cross-channel interaction was proposed. This method was called ECA

$$\left\{w_1, w_2\right\}(y) = W_2\mathrm{ReLU}\left(W_1 y\right). \tag{2}$$

ECA module aims at guaranteeing both efficiency and effectiveness. Specifically, the author employs a band matrix W_k to learn channel attention, and W_k has clearly W_k in a matrix as follows:

$$\begin{bmatrix} W^{1,1} & \cdots & W^{1,k} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & W^{2,2} & \cdots & W^{2,k+1} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & W^{C,C-k+1} & \cdots & W^{C,C} \end{bmatrix}.$$

As for Eq. (1), the weight of y_i is calculated by only considering the interaction between y_i and its k neighbors, i.e., where Ω_i^k indicates the set of k adjacent channels of y_i in Eq. (3).

$$\omega_i = \sigma\left(\sum_{j=i}^{k} w_i^j y_i^j\right), y_i^j \in \Omega_i^k \tag{3}$$

As for Eq. (4) to further improve the performance, it is also possible to have all the channels share the weight information and the information interaction between channels is achieved by a 1D convolution of kernel size k. C1D stands for one-dimensional convolution. To get the k-value, the coverage of cross-channel information interactions should also be proportional to the channel dimension C

$$\omega = \sigma(\mathrm{C1}D_k(y)) \tag{4}$$

$$\mathrm{C} = \phi(\mathrm{k}) = 2^{y*k-b}. \tag{5}$$

In other words, there may be a mapping ϕ of Eq. (5) between k and C. The simplest mapping is a linear function $\varphi(k) = y * k - b$; however, the relations characterized by linear functions are too limited. Therefore, we introduce a possible solution by extending the linear function to a nonlinear function. Then, given the channel dimension C, kernel size k can adaptively determine the convolution size k formula as follows [Eq. (6)]. Finally, we compared the number of parameters about SE and ECA. We can know both the accuracy and the number of parameters ECA is better than SE that ECA is an upgrade of SE

$$k = \psi(C) = \left|\frac{\log_2 C}{y} + \frac{b}{y}\right|_{\mathrm{odd}}. \tag{6}$$

2.2 MobileNet V2

MobileNet V2 [19] is a lightweight neural network model proposed by Google, which improves on MobileNet V1 [18] by enhancing the depthwise (Dwise) separable convolution and adding an inverted residual block. The improved depthwise separable convolution block (bottleneck) is shown in Fig. 4. First, it employs 1 × 1 convolution to increase the number of channels, then uses 3 × 3 deep convolution in a high-dimensional space, followed by another 1 × 1 convolution to reduce the number of channels, and finally applies the linear activation function. When stride = 1, MobileNet V2 uses residual connections to connect the inputs and outputs. However, when stride = 2, residual connections are not necessary, since the features of the inputs and outputs differ in size. Figure 5 compares the residual block of ResNet [24] with the inverted residual block of MobileNet V2. The activation function used is ReLU6, which sets any value above 0–6; if the value bigger than 6, the value is 6, and if the value smaller than 0, the value is 0. The ReLU6 function has a value range of [0,6] and offers better representation performance at low-precision floating-point numbers. To reduce the individual convolution block size, MobileNet uses the extended convolution length, which significantly reduces the network parameters and makes the network long and narrow. This design is advantageous for mobile devices, as the CPU is the primary processor. The CPU, with cache memory, can process long and thin programs much faster than the GPU. Hence, the MobileNet network outperforms numerous other neural networks on mobile devices.

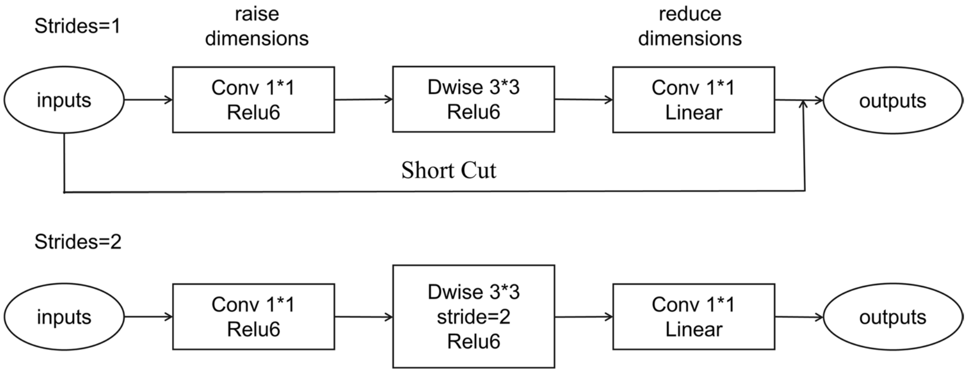


Fig. 4 The image is improved depthwise separable convolution; if strides = 1, we need to shortcut that means using residual connection

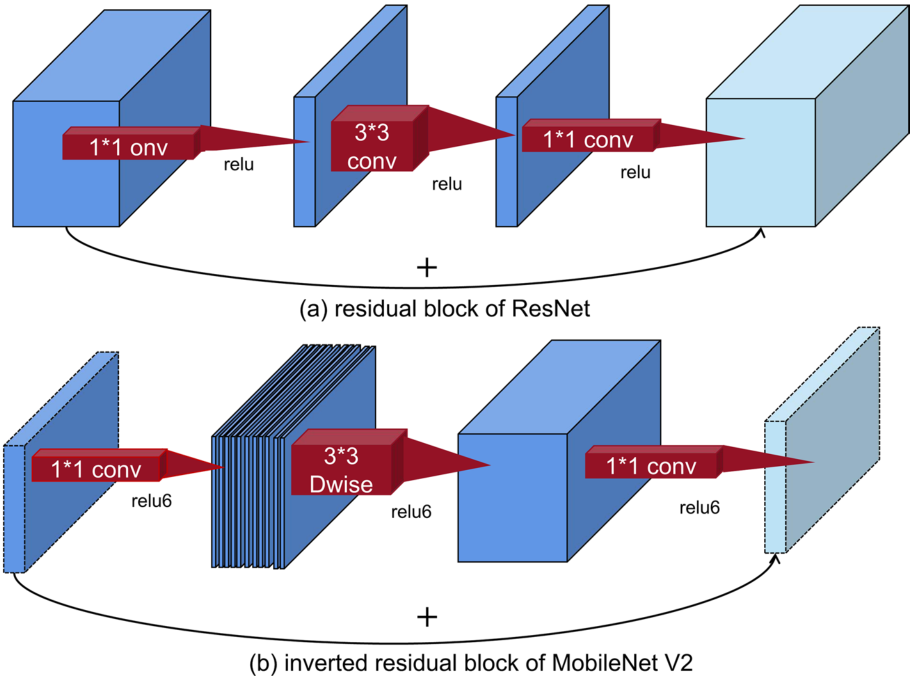


Fig. 5 ResNet first applies 1 × 1 convolution to reduce the dimensions, followed by standard convolution in the down-dimension space, and another 1 × 1 convolution to increase the dimensions. The residuals then connect the two high-dimensional parts. In contrast, MobileNet first applies 1 × 1 convolution to increase the dimensions, followed by standard convolution in the up-dimension space, and another 1 × 1 convolution to reduce the dimensions. The residuals then connect the two low-dimensional parts

2.3 SlAM System Framework

In this section, we describe how we build an SLAM system based on PTAM [25], with ECMobileNet used as the loop closure detector (LCD). PTAM comprises two threads for tracking and mapping, but does not include loop closure detection. For the tracking thread, the main task is to extract FAST features from the image, estimate the pose based on the previous frame, or initialize the pose through global relocation. It then tracks the reconstructed local map, optimizes the pose, and determines new keyframes according to set rules [26]. The mapping thread completes the construction of local maps by inserting keyframes, verifying newly generated map points, filtering them to generate new map points, and applying local bundle adjustment to remove redundant keyframes. To add loop closure detection based on ECMobileNet into PTAM, a new keyframe is defined through the tracking thread, transmitted to the mapping thread to form a local map, and sent to the LCD to judge whether a loop is formed.

PTAM is a landmark project in the field of visual SLAM. Before PTAM, the mainstream algorithm was MonoSLAM [27] based on Kalman filtering, which used a single thread to update the camera position pose and map by every frame. The computational complexity of map update was very high and achieved only real-time processing (30 Hz). MonoSLAM could only process about 10–12 most stable feature points per frame by the filtering method. The biggest contribution of PTAM is its dual-threaded architecture for tracking and mapping. The tracking thread only needs to update the camera position pose by every frame, which can be easily calculated in real time. The mapping thread does not need to update by every frame, so the original bundle adjustment (BA), which can only be used in offline structure from Motion, can also be used. This optimization algorithm can obtain higher accuracy per unit of computation time than the filtering method [28]. This multi-threaded processing is more in line with modern CPU trends, and almost all subsequent visual SLAM algorithms have followed this idea. For example, ORB-SLAM [10] incorporated loop closure detection, improving the practicality of PTAM and the accuracy of SLAM algorithms for map building, making the VSLAM system able to move longer distances. Later on, ORB-SLAM2 [29] and ORB-SLAM3 [30] were proposed.

3 Proposed Approach

3.1 Overview

The structure of the loop closure detection based on ECMobileNet which was put in the PTAM is shown in Fig. 6 for testing the effect of loop closure detection. Our proposed ECMobileNet is a lightweight neural network through MobileNet V2 that inserts the ECA and reduces some useless blocks. We designed an algorithm to extract the image features from the ECMobileNet and compute the similarity of images. Finally, setting a threshold to judge whether the current frame is or not loop.

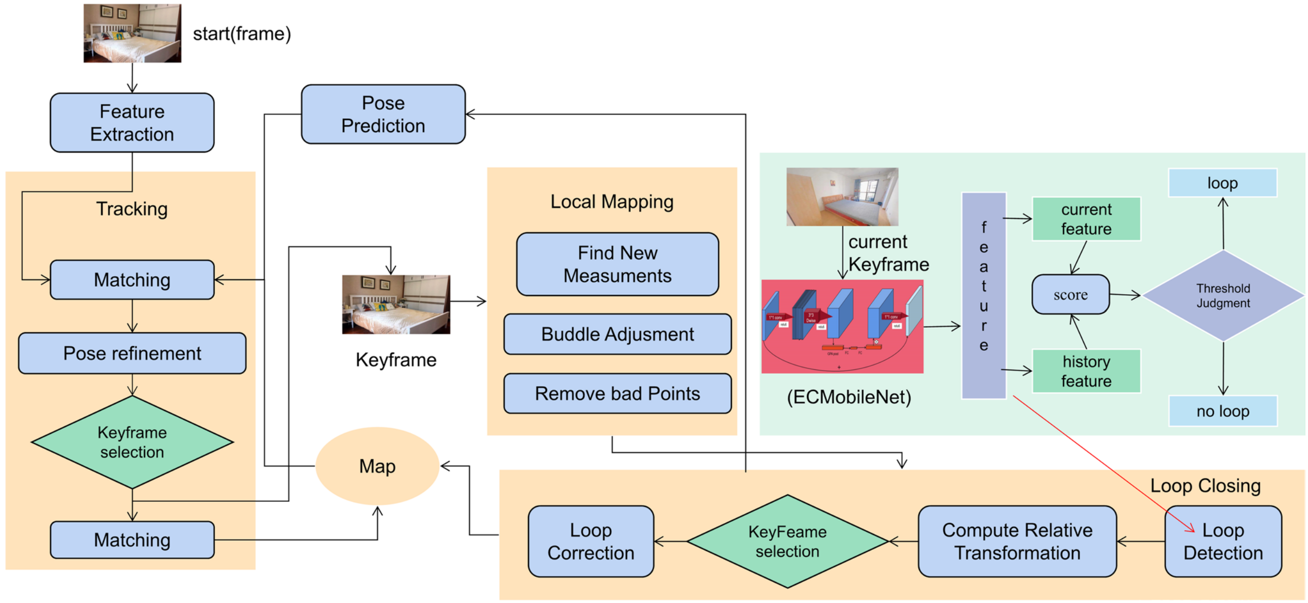


Fig. 6 In this image is the entire SLAM system including tracking, local mapping, and loop closure detection

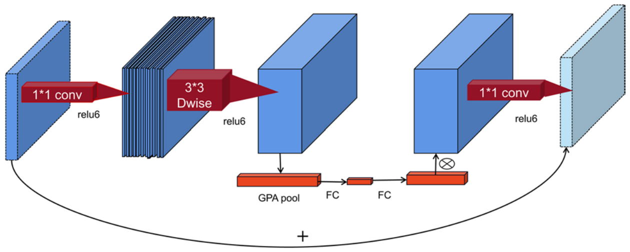


Fig. 7 This is a bottleneck, from this picture when the Dwise deals with conv using ECA and not changes the conv block

3.2 ECMobileNet

From MobileNet V2 to MobileNet V3, improvements were made by adding SE to improve precision and deleting unnecessary parts of output layers, resulting in MobileNet V3 [23]. After analyzing the SE and ECA, which belong to channel attention, we found that ECA has more efficient results with fewer parameters than SE. To incorporate ECA into the MobileNet V2 bottleneck, we designed the structure shown in Fig. 7. First, the image input is raised in dimensions using 1 × 1 convolution, and then, a standard convolution is performed in the up-dimension space. Next, the feature map data are optimized by the ECA's GPA pool and two fully connected layers. Finally, a 1 × 1 convolution is used to reduce the number of channels (using a linear activation function). When the stride = 1, the input and output feature maps have the same shape, and the input and output are connected using a residual connection. When the stride = 2 (downsampling stage), the feature of down-dimension is directly outputted. However, the insertion of ECA resulted in a bloated framework. To reduce the total model's parameters, we decided to cut some unnecessary bottlenecks and created the ECMobileNet structure shown in Table 1. We tested ECMobileNet on public datasets CIFAR-10 and CIFAR-100, achieving the same precision as MobileNet V2. We also used ECMobileNet to train the indoorCVPR dataset and obtained good results. Therefore, this model can be used in LCD.

Table 1 In this table, the t expresses the multiple of 1 × 1 convolution to raise dimensions, the c means the number of channels about 1 × 1 convolution, the n show times of bottleneck from MobileNet V2 to ECMobileNet, and the s is the stride

Input	Operator	*t*	*c*	*n*	*s*
224^2×3	conv2d	–	32	1->1	2
112^2×32	Bottleneck	1	16	1->1	1
112^2×16	Bottleneck	6	24	2->2	2
56^2×24	Bottleneck	6	32	3->2	2
28^2×32	Bottleneck	6	64	4->3	2
14^2×64	Bottleneck	6	96	3->2	2
14^2×96	Bottleneck	6	160	3->2	1
7^2×160	Bottleneck	6	320	1->1	2
7^2×320	conv2d 1 × 1	–	1280	1->1	1
7^2×1280	Avgpool	–	–	1->1	1
1 × 1 × 1280	conv2d 1 × 1	–	*k*	–	–

3.3 Loop Closure Detection Algorithm

LCD Frame [31] is composed of Input Image, Feature Extraction, Similarity Calculation, and Loop Judgment. First, the keyframe images are put into the ECMobileNet to get the pooled layer feature vectors v(1 × 1x1280), as shown in Fig. 8 and the feature vectors v are normalized through min–max normalization as follows [Eq. (7)]. Next, compressing the feature vectors v becomes 1 × 1x128. To ensure that each vector value is involved in the operation, we will every ten values of the vector group to become one value through addition and insert weight σ as follows [Eq. (8)] and put the new vector in the historical feature vectors store

$$v_i = \frac{v_i - \min(v)}{\max(v) - \min(v)} \tag{7}$$

$$a_i = \frac{\sigma\left(\sum_j^{j+10} v_j\right)}{10} j = i * 10. \tag{8}$$

When the number of historical feature vectors exceeds a certain threshold, a new feature vector is obtained from the latest keyframes using ECMobileNet. The new feature vector is then compared with the historical feature vectors using the Euclidean distance, as shown in Eq. (9), to obtain the similarity score. Finally, this score is compared with a threshold value to determine if a loop is detected. If the similarity score is greater than the threshold value, the result indicates a loop, and the SLAM system will perform graph optimization of the Essential Graph. If the similarity score is smaller than the threshold value, the system will calculate the next feature vector

$$\mathrm{d}(a_{\mathrm{now}}, a_{\mathrm{his}}) = \sqrt{\sum_{i=1}^{n} a_{\mathrm{now}} - x_{\mathrm{his}}}. \tag{9}$$

4 Experimental Results and Discussion

This section presents the experimental results and discussion on various aspects of ECMobileNet to determine if it meets our requirements. The experimental results demonstrate that ECMobileNet is a highly effective network that performs well, as expected.

4.1 Experimental Environment and Datasets

In this experiment, we utilized the Linux platform and TensorFlow deep learning framework, with Python language used for data analysis. We used the TUM indoor datasets as the test for loop closure detection, while IndoorCVPR dataset served as the training set for ECMobileNet. The IndoorCVPR dataset is a small collection of indoor scene images comprising 67 categories with a total of 15,620 images. The number of images per category varies, but each category has at least 100 images and all images are in JPG format. The TUM RGB-D datasets contain 39 sequences recorded in various indoor scenes using Microsoft Kinect sensors. The sequences cover Testing and Debugging, Handheld SLAM, Robot SLAM, Structure VS. Objects, 3D Object Reconstruction, Validation Files, and Calibration Files. Each sequence contains multiple data points that can be used to evaluate the performance of various tasks. The TUM datasets come with standard trajectories and comparison tools, making them ideal for research purposes. We used the test sets shown in Table 2 for our experiments.

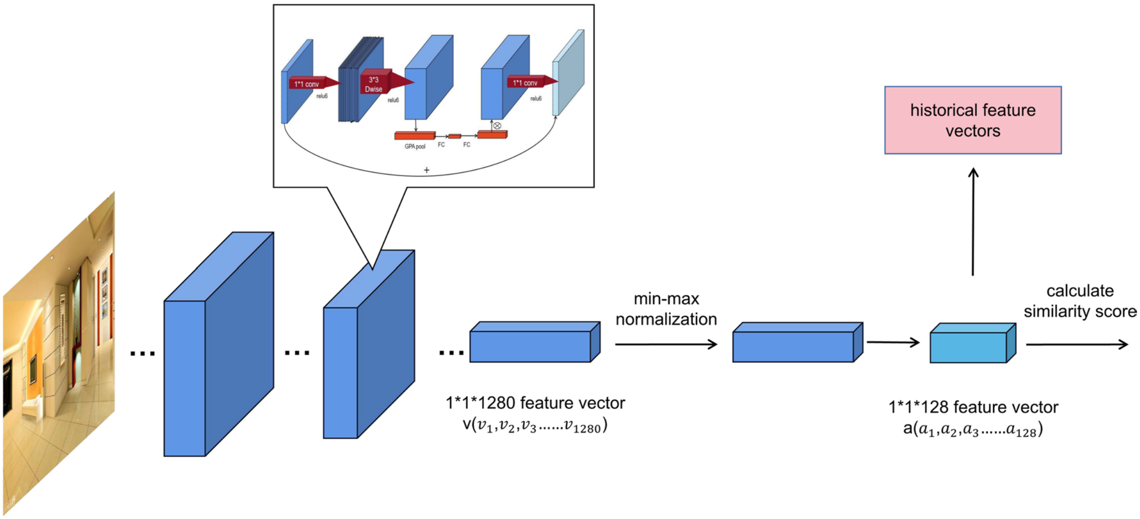


Fig. 8 The whole process of LCD based on ECMobileNet

Table 2 The pictures of TUM datasets

Datasets	Resolution	Images	Description
freiburg1_desk	640×480	608	Indoor
freiburg1_floor	640×480	1242	Indoor
freiburg2_large_with_loop	640×480	5182	Indoor
freiburg3_long_office_household	640×480	2585	Indoor

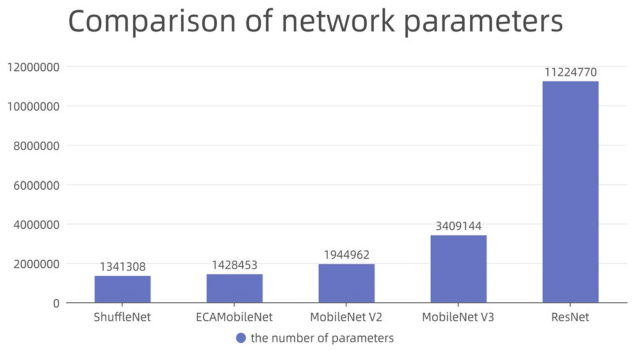


Fig. 9 This figure express the parameters of neural networks, and the parameters of ECMobileNet are very small

4.2 Neural Network Size of Comparison Between Parameters

In our quest for a neural network model with the smallest possible size, we compared the parameter counts of different neural networks for 224×224x3 images under 67 classifications, as shown in Fig. 9. It is evident that the number of parameters in ECMobileNet is smaller than that of other lightweight neural networks, such as MobileNet V2 and MobileNet V3. This proves that ECMobileNet is indeed a lightweight neural network with a clear advantage in terms of parameter count. In comparison to the improved neural network, MobileNet V2, the parameter count is reduced by 26.5%, and the parameter count of MobileNet V2 is twice that of ECMobileNet. Moreover, the parameter count of the large model ResNet18 is only one-tenth that of ECMobileNet. Hence, the number of parameters of ECMobileNet is almost the same as that of ShuffleNet.

4.3 Performance Evaluation: Precision–Recall Curve

To compare the algorithms, analyze the experimental results, and evaluate their performance, we compared the traditional ORB with ShuffleNet V1, MobileNet V2, ResNet18, and ECMobileNet based on image similarity. The evaluation method used in the experiment is the precision–recall rate curve, which is currently recognized as a standard evaluation method. In the SLAM system, precision is of higher importance than recall, because a low recall rate may result in some true loops being unrecognized, while low accuracy can lead to incorrect results in the backend optimization and ultimately result in an incorrect map construction [32]. The precision–recall rate is calculated as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{recall} = \frac{\text{TP}}{\text{FN} + \text{TP}}.$$

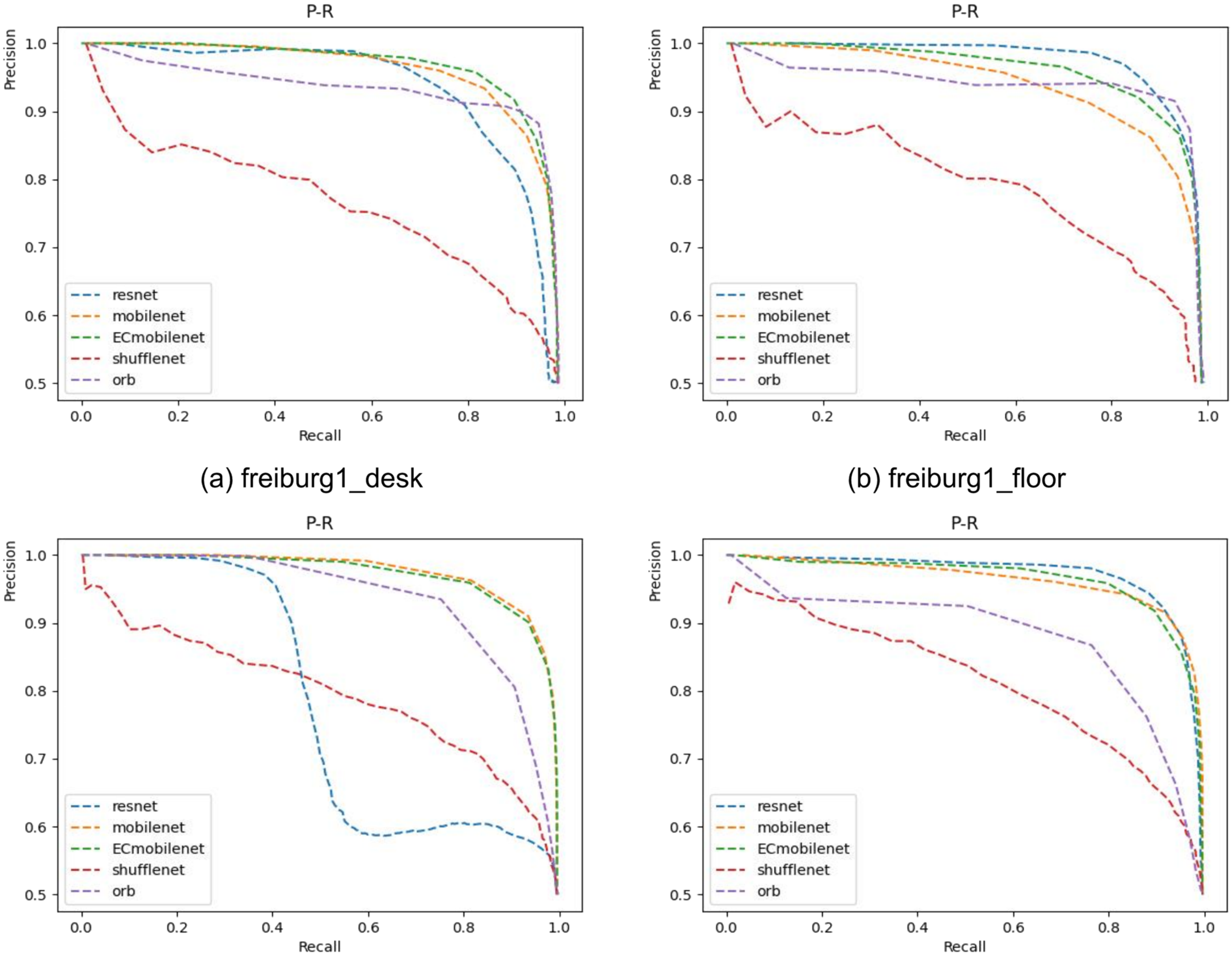
In the formula, TP represents a true positive, which means that a true loop is detected as a loop by the algorithm; FP means a false positive, and a wrong loop is detected as a loop by the algorithm; FN means a false negative. The experimental results are shown in Fig. 10. The results show that when the recall rate is 60%, the precision of the ECA algorithm in this paper is very high and stable. The shuffleNet's behavior is very bad. ResNet is although the best result but very unstable. MobileNet V2 is not as good as ECMobileNet. ORB cannot achieve 100% precision and the area of the Precision–Recall Curve is small. From an overall point of view, the algorithm curve of this paper is biased to the upper right. Experiments have proved that the algorithm proposed in this paper performs better on the same hardware conditions and test set, ensuring a certain accuracy rate.

4.4 The Performance of PTAM

The accuracy of localization is crucial for map building in a visual SLAM system. Loop closure detection plays a critical role in localization, especially for robots that need to move repeatedly. In this paper, we conducted experiments based on PTAM and set the loop closure detection threshold relatively high to ensure accuracy. Reducing the threshold could result in false loop closures and distort the maps. To demonstrate the applicability of the ECMobileNet-based loop closure detection algorithm in an SLAM system, we ran PTAM with loop closure detection on several scenarios of TUM RGB-D datasets, as shown in Fig. 11. The trajectories after loop closure detection were more stable, indicating the effectiveness of the algorithm. We also observed that even a single lap of error could cause significant drift, emphasizing the importance of loop closure detection in visual SLAM.

5 Conclusion

To reduce the parameters of the neural network while maintaining real-time LCD, we propose using ECMobileNet. Although both the ECA and SE modules are channel attention mechanisms with similar properties, we found that ECA is smaller and performs better than SE, so we chose to adopt



(a) freiburg1_desk

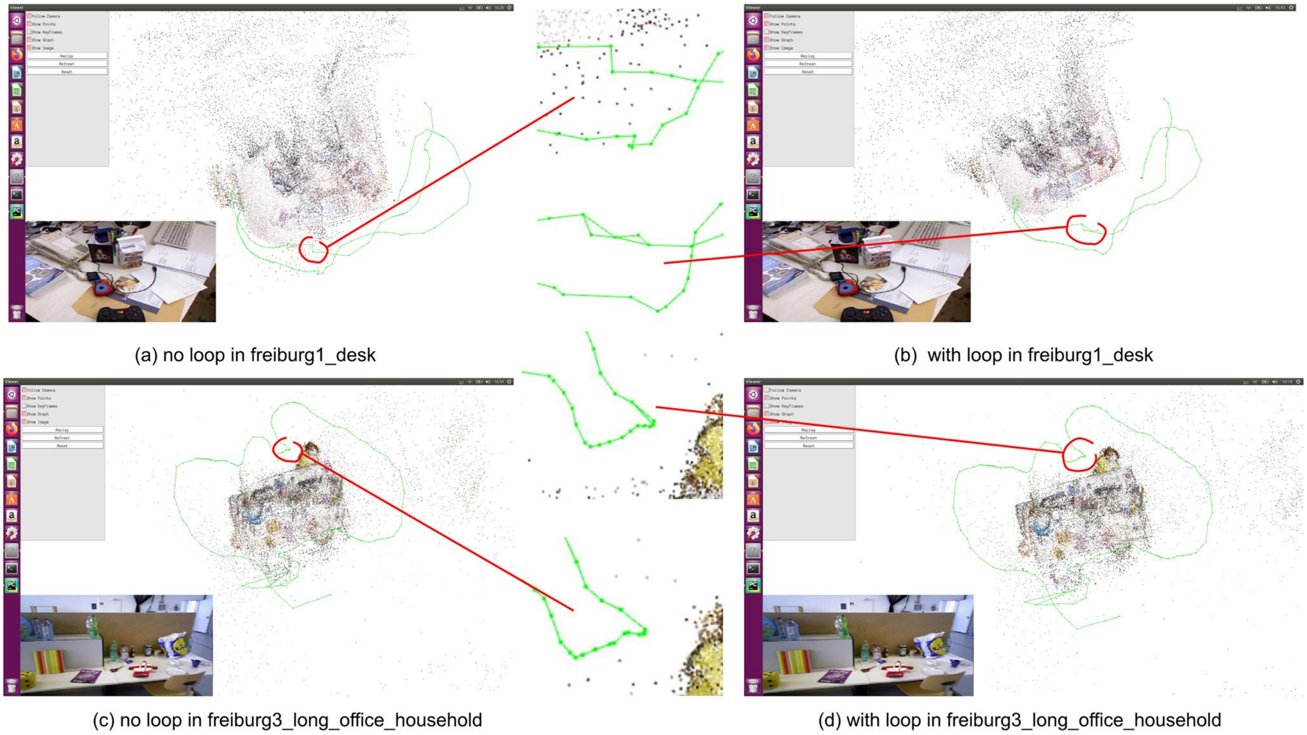
(b) freiburg1_floor

(c) freiburg2_large_with_loop

(d) freiburg3_long_office_household

Fig. 10 P–R curve about TUM datasets

ECA and achieved very good results. For our dataset selection, we focused on the indoor environment and used the indoorCVPR dataset, which we modified slightly to better target indoor environments. Our final results were as good as we hoped for the public TUM datasets. The compression of the MobileNet V2 module was just an idea that was tried and the experiments showed that the reduction of the bottleneck did not reduce the accuracy, just like MobileNet V3 removed the invalid output layer of MobileNet V2. Overall, ECMobileNet was built with a very small number of parameters and performed better than the current ResNet, which performs very well in image features. We created a loop closure detection algorithm based on ECMobileNet to verify whether it could be implemented in SLAM, and as expected, the LCD algorithm worked effectively. However, there are some shortcomings and areas for improvement in our study. First, in this paper, we used TensorFlow on a GPU for our experiments, which means that we could not take full advantage of the benefits of lightweight neural networks on a CPU. We have no way of knowing exactly how this algorithm behaves on a microprocessor. Second, the indoorCVPR dataset is relatively small, and the training effect on some networks was limited. To address these issues, in the future, we plan to rewrite all network models through the TensorFlow Lite framework and then port them to the microprocessor for experiments. We will also expand the number of indoorCVPR dataset to make it more comprehensive and have stronger generality.


(a) no loop in freiburg1_desk

(b) with loop in freiburg1_desk

(c) no loop in freiburg3_long_office_household

(d) with loop in freiburg3_long_office_household

Fig. 11 In this image using loop closure detection based on ECMobileNet work well

Acknowledgements The authors would like to thank the anonymous reviewers for providing helpful comments.

Author Contributions DZ carried out investigation, methodology, writing, and took care of project administration and neural network training. YL build an environment about Tensor Flow. QZ makes the datasets. YX provides fund support. DC carried out investigation, conceptualization, review, and editing. XZ made some constructive suggestions and ideas.

Funding This work was partially funded by National Key Research and Development Program of China under Grant No. 2020AAA0108905.

Data Availability Data will be made available on reasonable request.

Declarations

Conflict of Interest The authors declare that they have no known competing financial interests or personal relationships.

Ethical Approval and Consent to Participate Not applicable.

Consent for Publication Not applicable.

References

1. Munoz-Salinas, R., Medina-Carnicer, R.: UcoSLAM: simultaneous localization and mapping by fusion of keypoints and squared planar markers. Pattern Recogn. **101**, 107193 (2020)
2. Taketomi, T., Uchiyama, H., Ikeda, S.: Visual SLAM algorithms: a survey from 2010 to 2016. IPSJ Trans. Comput. Vis. Appl. **9**(1), 1–11 (2017)
3. Niloy, M.A.K., Shama, A., Chakrabortty, R.K., et al.: Critical design and control issues of indoor autonomous mobile robots: a review. IEEE Access **9**, 35338–35370 (2021)
4. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Robot. Autom. Mag. **13**(2), 99–110 (2006)
5. Durrant-Whyte, H.: Simultaneous localization and mapping (slam): part II. IEEE Robot. Autom. Mag. **13**(3), 108–117 (2006)
6. Ho, K., Newman, P.: Loop closure detection in slam by combining visual and spatial appearance. Robot. Auton. Syst. **54**(9), 740–749 (2006)
7. Thrun, S.: Simultaneous localization and mapping. In: Robotics and cognitive approaches to spatial mapping, pp. 13–41. Springer, Cham (2007)
8. Jin, J., Bai, J., Xu, Y., et al.: Unifying deep ConvNet and semantic edge features for loop closure detection. Remote Sens. **14**(19), 4885 (2022)

9. Warren, M., McKinnon, D., He, H., et al.: Large scale monocular vision-only mapping from a fixed-wing sUAS. In: Field and service robotics, pp. 495–509. Springer, Berlin (2014)
10. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Rob. **31**(5), 1147–1163 (2015)
11. Bay, H., Ess, A., Tuytelaars, T., et al.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
12. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval, Vol. 463. ACM press, New York (1999)
13. Sivic, A.: Zisserman video google: a text retrieval approach to object matching in videos. Proc. IEEE Int. Comput. Vis. (2003). https://doi.org/10.1109/ICCV.2003.1238663
14. Memon, A.R., Wang, H., Hussain, A.: Loop closure detection using supervised and unsupervised deep neural networks for monocular SLAM systems. Robot. Auton. Syst. **126**, 103470 (2020)
15. Naseer, M., Ruhnke, C., Stachniss, C., Spinello, L., Burgard, W.: Robust visual SLAM across seasons. Iros (2015). https://doi.org/10.1109/IROS.2015.7353721
16. Zhang, X., Su, Y., Zhu, X.: Loop closure detection for visual SLAM systems using convolutional neural network. In: 2017 23rd International Conference on Automation and Computing (ICAC), IEEE, pp. 1–6 (2017)
17. Arshad, S., Kim, G.-W.: Role of deep learning in loop closure detection for visual and lidar slam: A survey. Sensors **21**(4), 1243 (2021)
18. Howard, A. G., Zhu, M., Chen, B., et al.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
19. Sandler, M., Howard, A., Zhu, M., et al.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
20. Koonce, B.: MobileNetV3. In: Convolutional neural networks with swift for tensorflow, pp. 125–144. Apress, Berkeley (2021)
21. Zhang, X., Zhou, X., Lin, M., et al.: Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856 (2018)
22. Wang, Q., Wu, B., Zhu, P., et al.: Supplementary material for 'ECA-Net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Seattle, WA, USA, pp. 13–19 (2020)
23. Jie, H., Li, S., Gang, S.: Squeeze-and-excitation networks. In: CVPR (2018)
24. Wu, Z., Shen, C., Van Den Hengel, A.: Wider or deeper: revisiting the resnet model for visual recognition. Pattern Recogn. **90**, 119–133 (2019)
25. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality, IEEE, pp. 225–234 (2007)
26. Dong, N., Qin, M., Chang, J., et al.: Weighted triplet loss based on deep neural networks for loop closure detection in VSLAM. Comput. Commun. **186**, 153–165 (2022)
27. Davison, A.J., Reid, I.D., Molton, N.D., et al.: MonoSLAM: real-time single camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)
28. Strasdat, H., Montiel, J. M. M., Davison, A. J.: Real-time monocular SLAM: why filter? In: Robotics and Automation (ICRA), 2010 IEEE International Conference on IEEE, pp. 2657–2664 (2010)
29. Mur-Artal, R., Tardós, J.D.: Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Trans. Rob. **33**(5), 1255–1262 (2017)
30. Campos, C., Elvira, R., Rodríguez, J.J.G., et al.: Orb-slam3: an accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Trans. Rob. **37**(6), 1874–1890 (2021)
31. Dian, S., Yin, Y., Wu, C., et al.: Loop closure detection based on local-global similarity measurement strategies. J. Electron. Imaging **31**(2), 023004 (2022)
32. Zhu, M., Huang, L.: Fast and robust visual loop closure detection with convolutional neural network. In: IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC), IEEE, pp. 595–598 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.