Journal of Applied Data Sciences
Vol. 3, No. 3, September 2022, pp. 102-109

Embedded Image Recognition System for Lightweight Convolutional Neural Networks

Jie Fang [1,*], Xiangping Zhang [2]

[1] Wuhan Qingchuan University, Wuhan, Hubei 430204, China
[2] Hubei Business College, Wuhan, Hubei 430079, China
[1] fangjie@qcuwh.cn*
* corresponding author

(Received: April 23, 2022; Revised: May 12, 2022; Accepted: July 18, 2022; Available online: September 30, 2022)

Abstract

In this paper, we design and implement an embedded image recognition system based on STM32 for the problem of limited storage space of embedded systems to run convolutional neural networks efficiently, and for the loading of lightweight convolutional neural network and the hook-up requirement of the quadrotor. The system hardware adopts the idea of modular design to improve the compatibility of the system, and the system software adopts the training of handwritten image recognition based on convolutional neural network, lightweight processing of the convolutional neural network, and transplanting the trained network to the embedded system. Finally, the system can finish the recognition of handwritten images stably and efficiently. This system can provide a low-cost and highly integrated solution for such image processing systems. The lightweight target detection model CED-Det is designed by combining CED-Net and dense feature pyramid network, which firstly performs feature extraction by CED-Net, then performs feature fusion by stacking two layers of dense pyramid network, and finally, the fused feature maps are used for classification prediction and position prediction by two 3×3 convolutions, respectively. CED-Det is used in VOC and Experimental results on COCO datasets show that CED-Det is more suitable for embedded platforms in terms of accuracy, inference speed, and a total number of parameters compared with other target detection models.

Keywords: Lightweight Convolutional Neural Network, Embedded, Image Recognition System

1. Introduction

With the application scenarios, the lightweight of algorithmic models has become a hot research topic in recent years [1]. At the beginning of the development of convolutional neural networks, researchers have proposed many pruning and compression algorithms to reduce the redundant parameters in the network and speed up the model inference. Pruning compression algorithms are mainly applied to perform feature extraction in convolutional neural networks, by measuring the importance of the weight parameters in the network and pruning the less important ones. For example, the early pruning compression algorithms work well for VGG convolutional neural networks without jump connections, but not for ResNet with residual structure. In contrast, it is more efficient to design lightweight convolutional neural networks directly by improving the structure of convolutional operations and combining them with advanced target detection frameworks, such as MobileNet v2 based on deep separable convolution combined with SSDLite, which can efficiently perform target detection tasks in mobile [2]. The disadvantages of traditional target detection algorithms are obvious due to the limitations of embedded devices and computing resources on the mobile side, which consumes a lot of hardware resources in a multitasking scenario running in parallel. Therefore, it is very important to design efficient lightweight target detection models directly. By designing lightweight convolutional neural network models that conform to embedded devices, we combine them with target detection algorithms, thus improving model accuracy and efficiency under resource constraints

Tang et al. proposes a generalized construction method for convolutional neural networks on an embedded field-programmable gate array (FPGA) platform for edge computing [3]. By designing a gas pedal core that can be reused among network layers in the convolutional neural network function, the performance-optimized convolutional neural network hardware is achieved with a small number of hardware resources. Dang et al. proposes a method of chunking two-dimensional direct memory access to compensate for the limited storage resources and an optimization

strategy that trade-offs the use of digital signal processing units and lookup tables to compensate for the lack of computational resources [4]. Also, converting floating-point numbers to fixed-point numbers is an optimization scheme.

As a classical model of the deep neural network, the convolutional neural network has become a hot spot in the field of image recognition because of its good learning ability and generalization ability for high-dimensional features, which is excellent in solving problems such as image classification, speech recognition, and pedestrian detection. Since deep neural networks generally rely on the powerful computing power of graphics processors (GPUs), this greatly limits the use of deep neural networks on mobile devices. To meet the demand of miniaturization, high integration, and intelligence, the embedded image processing system integrating image acquisition, processing, transmission, and display has become one of the development branches of the image processing systems. With the popularity of smart mobile devices and the unprecedented development of portable mobile devices, the application of deep neural networks to embedded devices to improve the user experience and expand the scope of application has shown great engineering appeal. The limited storage resources and computational power are the problems that need to be solved for applying convolutional neural networks in embedded systems. In addition, to be mounted on UAV platforms, it is also necessary to reduce the size and weight of the image processing system. In this regard, the embedded image processing system is optimized from the perspective of hardware and software by taking the quadrotor mount device as the design requirement, and the training and transplantation of the convolutional neural network are completed by taking handwriting as an example, and finally, a lightweight embedded image processing system capable of handwriting recognition is realized.

2. Address Embedded Image Recognition System Design for Lightweight Convolutional Neural Networks

2.1. Analysis of Lightweight Convolutional Neural Networks

As the research of deep learning continues to progress, convolutional neural networks, as the most classical network structure of deep learning, are widely used in the fields of image recognition, target detection, and semantic segmentation, etc. Compared with traditional artificial neural networks, convolutional neural networks have more excellent performance in image recognition. Firstly, the development of convolutional neural networks is described, then the basic structure of convolutional neural networks is introduced, and finally, the classical target detection algorithms are summarized [5]. The convolutional neural network structure is a kind of deep artificial neural network specially designed for processing the input two-dimensional data. The basic network structure is composed of several two-dimensional planes stacked without functions, and each plane contains numerous independent neuron nodes, and the neuron nodes in two adjacent layers before and after are partially connected, while the neurons in the same layer are independent of each other. The structural design of the convolutional neural network is inspired by the earlier time-delay neural network, which reduces the complexity of network training by sharing parameter weights in the time dimension and is more conducive to the processing of one-dimensional speech data and time-series data. Convolutional neural networks use a weight-sharing network structure to make them more like biological neural networks, while the specification of the model can be adjusted by changing the depth and breadth of the network, which is also highly robust to natural images. Compared with the fully connected neural network, the convolutional neural network with local connections can use fewer network connections and weight parameters for training, which effectively reduces the learning cost of the network model and thus improves the training efficiency of the model. The basic structure of the convolutional neural network is shown in Figure 1.

The basic structure of a typical CNN usually consists of three main parts: a data input port, a feature extraction part consisting of alternating convolutional and pooling layers, and finally an output part consisting of one or more fully connected layers. In some cases, the fully connected layers can also be replaced by a global average pooling layer. The arrangement of CNN components plays a fundamental role in designing new architectures and obtaining enhanced performance. This section briefly discusses the role of the following seven components in the CNN architecture.

ISSN 2723-6471

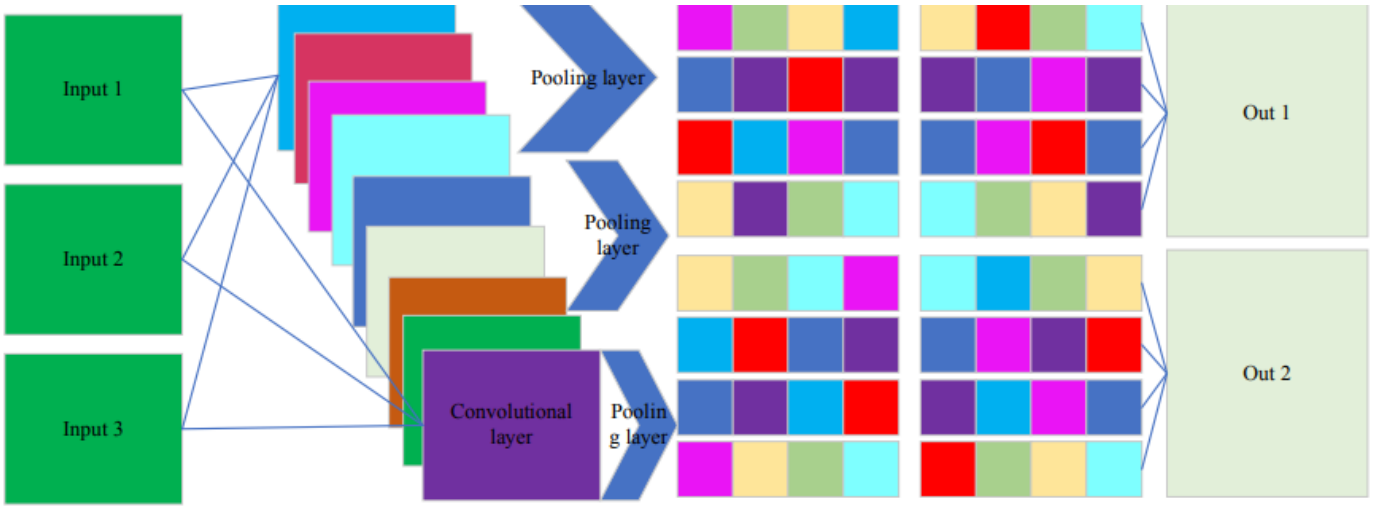

Figure. 1. Convolutional neural network basic structure diagram

The role of convolutional layers is to perform feature extraction. For an input image, each convolution layer contains multiple convolution kernels, and each convolution kernel can be convolved with the input image to produce a new image [6, 7]. The feature map of the current layer is a representation of the features in a small area covered by the convolution kernel. The feature map of the current layer is a combination of multiple feature maps extracted from the previous layer. Such a complex network composition would be very complex to learn directly by the back propagation algorithm and requires a proper design to learn effectively. The most important idea behind the design of the convolutional layer is sparse connectivity and weight sharing. Sparse connectivity means that each pixel of the output feature map is associated with only a small region of the previous feature map. The purpose of this design is to ensure that the features are translation invariant, which is very important in the field of image recognition; weight sharing means that the same convolutional kernel is used to traverse the entire input image each time, which can greatly reduce the number of parameters and play a role in regularization.

The size of the output feature map can be calculated according to the following steps: let the size of the convolution kernel be k, the sliding step of convolution, the size of the input feature map, the number of convolution kernels be m, the zero-fill parameter, and the size of the output feature map.

$$h_{out} = \frac{(h+k-2xp)}{s} - 1 \qquad (1)$$

$$w_{out} = \frac{(h+k-2xp)}{s} - 1 \qquad (2)$$

The design of the activation layer is based on a human neural network, simulating the reflex function of neurons. The above-mentioned convolutional and pooling layers operate on images with linear transformations, such that the expressiveness of the features is not good enough for the classifier to distinguish. Therefore, a nonlinear mapping of the extracted feature maps can be done by a nonlinear activation function, which enhances the expression capability of the features. A 256-channel feature map is an output using 256 convolution kernels, and the feature map contains 256 attributes of the input data. The disadvantage of standard convolution is that since the number of convolution kernels is the same as the number of channels based on the input image if the number of extracted image attributes needs to be increased, the number of convolution kernels needs to be increased. The computation process of standard convolution is that each convolution kernel is computed for each channel, and each convolution kernel stores the parameters of all channels, and the total number of parameters is shown in equation (3).

$$Parameter = k + k^{*}n - m^{*}n \qquad (3)$$

where k is the convolution kernel size, n is the number of channels of the input image, and m is the number of convolution kernels. The deep separable convolution divides the standard convolution into two steps, and each

convolution kernel corresponds to only one channel for computation. The input image scale is 12 × 12 × 3. First, three 5 × 5 depth convolutions are used to convolve each of the three channels to obtain a feature map with a scale of 8 × 8 × 3; then the number of channels of the feature map is increased by 256 1×1 point convolutions; finally, the output feature map with a scale of 8 × 8 × 256 is obtained. The total number of parameters is shown in equation (4).

$$Parameter = k - k^{*}n + m^{*}n \quad (4)$$

After the network generates the a priori frames, it will match with the real frames according to two principles: first, it will find the matching a priori frame with the largest IoU value for each target in the input image; second, it will find the matching a priori frame with the real frame whose IoU value is larger than the threshold (usually 0.5) for the remaining unmatched a priori frames to ensure that each a priori frame has its matching real frame. All matched prior frames are filtered by the NMS algorithm for redundant bounding boxes and then the border coordinates and category scores of each target are output.

In target detection, the positive and negative samples are determined based on the match between the a priori frame and the real frame, however, the proportion of targets in an image is often very small, which means that the number of negative samples is much larger than that of positive samples. In addition, most of the negative samples play a very small role in the training phase of the model, and most of these samples are easily distinguished, so their loss values are small, and many simple samples in the training phase of the model will make the model converge too slowly. To better displace the effect of the background on the recognition target, the model trains the negative samples, i.e., the background, as a separate category together with other targets, and mines all the negative samples for the hard-to-classify samples.

2.2. Embedded image recognition system design

For the hardware system design, the STM32H750 is used as the main control, with a CPU processing speed of 480 MHz; the FPU floating-point unit is designed to facilitate fast and high-precision floating-point operations; a special DSP library is available for fast matrix operations. The camera is an OV5640 module with up to 500 W pixels, which helps to extract the features of the pictures. The driver design includes SCCB, USB communication driver, SD card, file system, DCIM data center infrastructure management, screen display FMC driver, external flash, external SDRAM expansion, etc. The application-oriented functional software design is written with pre-processing algorithms such as transforming the format of the captured images, which can realize functions such as recognizing specific color or shape modules, recognizing 2D codes or bar codes, patrolling lines, and aerial photography. QT is used to write the upper computer, which transmits data through the communication module and can display information such as the location and frame rate of the identified target in real-time on the upper computer, and can also be used to debug each module and switch module functions quickly and easily, as shown in Figure 2.

ISSN 2723-6471

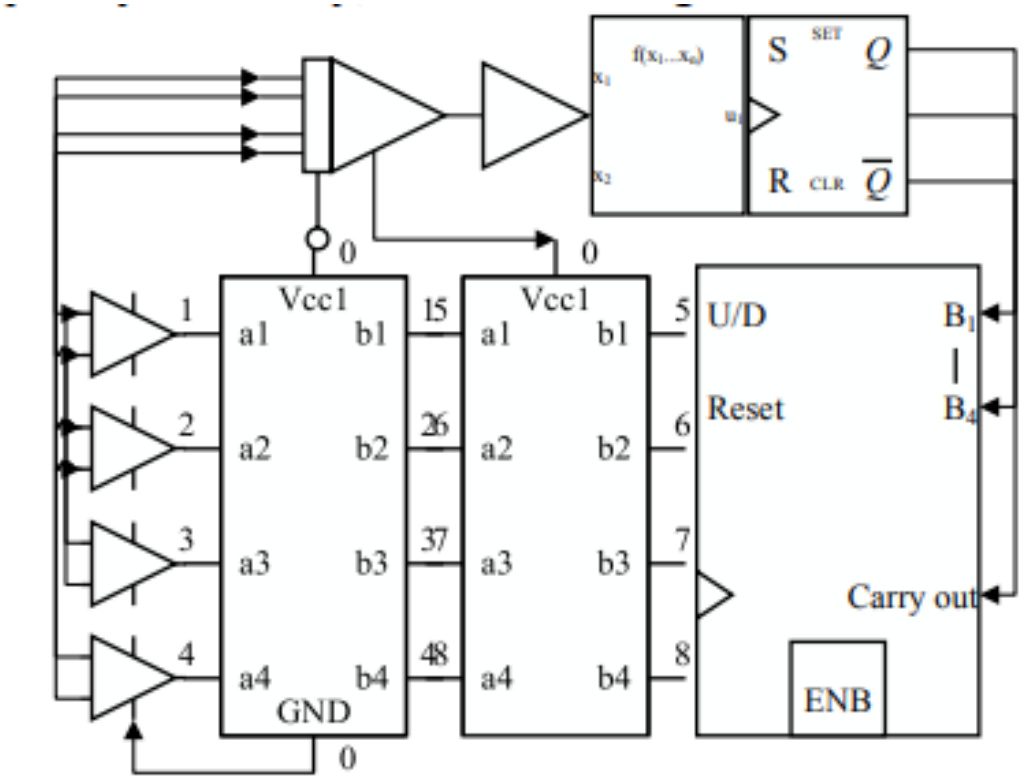

Figure. 2. Hardware framework of the embedded image processing system

Most of the network models that perform best in computer vision tasks mostly use feature information learned in pre-training tasks and are better able to transfer the learned useful information to another learning task. In recent years, most target detection algorithms have been trained by pretraining feature extraction models using large-scale datasets such as ImageNet and then fine-tuning the full model on the target detection dataset [8, 9]. This training scheme has achieved state-of-the-art results not only in target detection tasks, but also in image segmentation, key point recognition, and other vision tasks. However, the necessity of this training method is questioned, as comparable results using pre-trained networks can be achieved by randomly initializing the network on the COCO dataset by retraining it, and the training results were found to be comparable using only a tenth of the data. Therefore, the training process of the network can be completely abandoned from the tedious pretraining step to simplify the training process and avoid unnecessary resource loss.

The Keras open-source artificial neural network tool library is used for the design, debugging, evaluation, and application of deep learning models. With handwritten digit recognition as the target, the neural network structure with the best fit is found by testing on a handwritten digit dataset, consisting of three convolutional layers, two pooling layers, and two fully connected layers. The convolutional neural network relies on the large-scale data processing and storage capability of the hardware system. To adapt to the ARM-based embedded system platform, the convolutional neural network needs to be lightened. Firstly, all images are converted to grayscale format to reduce the number of channels; secondly, fewer layers are used to complete the set task; and pooling layers are added in the middle of successive convolutional layers to reduce the parameters and computation while retaining the main features to prevent overfitting and improve the generalization ability of the model. In this way, the light-weighting of the convolutional neural network is completed [10].

3. Analysis of Results

The experimental environment and setup of this chapter are the same as those in Chapter 4, but this chapter does not optimize the hyperparameters, but simply uses the embedded slice module to replace the fit module in ZDNet. In this

paper, we present the image classification results of SFNetV2 on Cigar and SVHN datasets. The experimental results show that the embedded bottleneck structure proposed in this paper can not only significantly reduce the computational complexity of the network but also maintain the same or even slightly higher performance than the standard convolution. The training set is 16551 images from VOC 2007 and VOC 2012, and the test set is 4952 images from VOC 2007. The input image scale is 300×300, and the data enhancement strategy is the same as SSD. The convergence process of the four normalization methods, BN, LN, IN, and GN, is compared with the original SSD and DSOD pre-trained with VGG, based on the construction of the four network models. Both SSD and DenseNet+DFPN models use the modified CIoU loss as the loss function, the network is trained with Adam optimizer, the initial learning rate is 0.001, the batch size is 32, and a total of 100 training rounds are performed, and the average precision (mAP) commonly used for target detection as well as the total number of parameters and inference delay are used as the evaluation criteria of the target detection model , as shown in Figure 3.

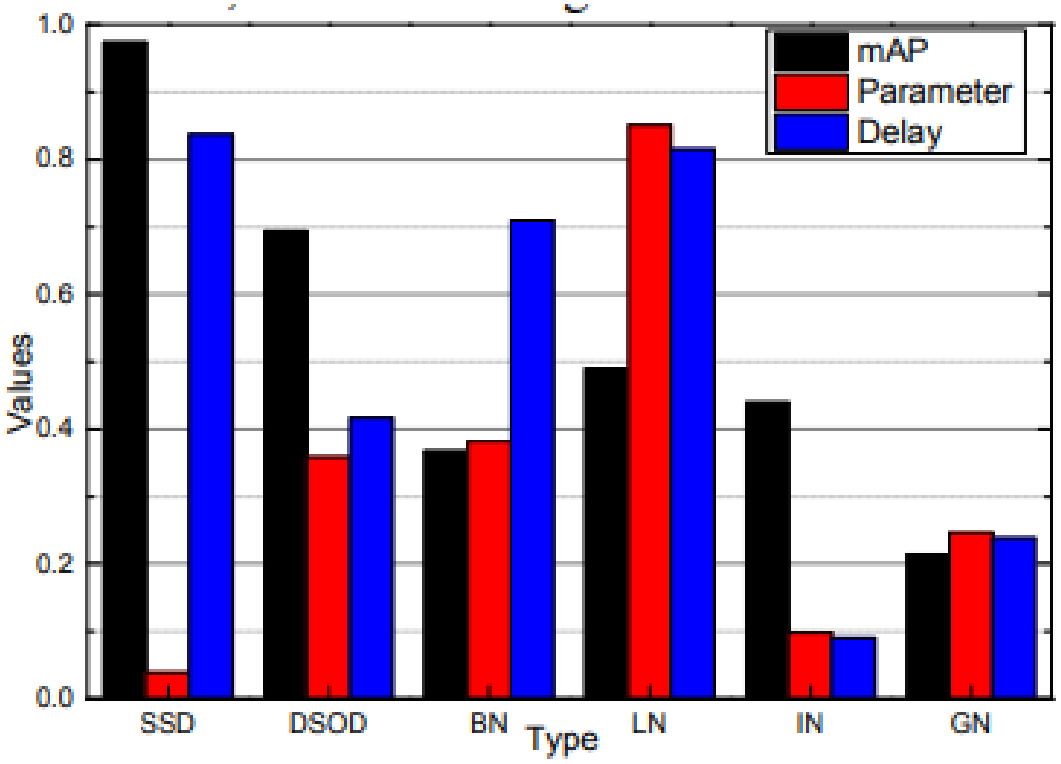

Figure. 3. Experimental results of the data set

Data set 12 denotes the training set with the sum of the training images of VOC2007 and 2012, and the test image of VO2007 as the test set. From the experimental results, the four models trained with random initialization embedded in the normalization layer all have improved mAP over the SSD using the pre-trained network, with the highest mAP improvement of 79.8% for GN over the SSD using the pre-trained network by 4.0%. In addition, compared with DSOD, DFPN can improve the mAP by 2.1% with less parameter cost.

The core idea of SFNetV2 lies in the embedded bottleneck structure and the embedded cutoff module proposed in this chapter. In this subsection, we evaluate the embedded bottleneck structure proposed in this chapter separately. To better highlight the performance of the embedded bottleneck structure, we modify MobileNet and MobileNetV2 by replacing their use of deep separable convolution with the embedded bottleneck structure. We also compare SFNetV2 with SFNet, as shown in Figure 4.

ISSN 2723-6471

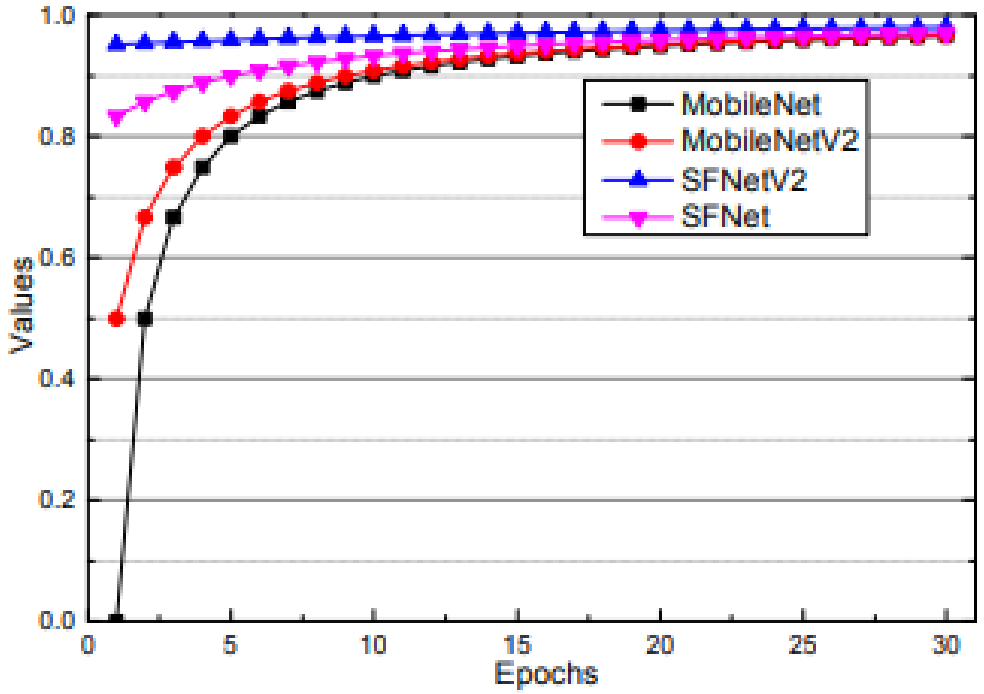

Figure. 4. Correct classification rate of each network before and after using the embedded bottleneck

As can be seen in Figure 4, replacing the deeply separable convolutions in MobileNet and SFNet with the embedded bottleneck structure reduces the number of parameters by more than 41% and the computational effort by more than 36%, with only a slight reduction in the correct classification rate. In MobilenetV2, only a 12.5% reduction in computational cost is obtained by using linear bottlenecks, mainly because MobilenetV2 already uses a normal bottleneck structure with a reduction factor of 6, so the reduction in computational cost is not significant. Experimentally, we can see that replacing the deeply separable convolution with an embedded bottleneck structure can reduce the parameters significantly and maintain almost the same performance as the original network. A new lightweight convolutional neural network building block, the cut module, is proposed. By combining group convolution with a multi-branch structure, this module increases the ability of the network to extract features while reducing the complexity of the network. With the same number of channels, the network complexity of the tangent module is smaller than that of the current state-of-the-art deep separable convolution.

4. Conclusion

An embedded handwritten image recognition system for lightweight convolutional neural networks is designed and implemented, which can be mounted on a quadcopter by optimizing the hardware and software design. The system achieves handwritten digit recognition based on a convolutional neural network, and the speed and efficiency of the recognition can well meet the application requirements. The system designed in this project can provide a low-cost and highly integrated solution for such image processing systems. The handwritten image recognition system is designed to be mounted on a quadcopter by optimizing the hardware and software. The system achieves handwritten digit recognition based on a convolutional neural network, and the speed and efficiency of recognition meet the application requirements well. The system designed in this project can provide a low-cost and highly integrated solution for such image processing systems. In practice, the image recognition speed, i.e., the frame rate, is up to 10 fps, and the recognition frame rate is calculated by counting the number of images processed by the system in 1 s. From the test results, the numbers can be recognized well with the right camera angle, and the system has very good real-time performance and stable recognition results, which can efficiently detect numbers and output recognition results.

References

[1] Guo B, Nyman L, Nayak A R, et al. Automated plankton classification from holographic imagery with deep convolutional neural networks. Limnology and Oceanography: Methods, 2021, 19(1): 21-36.

[2] Pandey V. Light weighted Convolutional Neural Network for License Plate Recognition. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 2021, 12(12): 4119-4125.

[3] Tang Y, Teng Q, Zhang L, et al. Layer-wise training convolutional neural networks with smaller filters for human activity recognition using wearable sensors. IEEE Sensors Journal, 2020, 21(1): 581-592.

[4] Dang X K, Truong H N, Nguyen V C, et al. Applying convolutional neural networks for limited-memory application. TELKOMNIKA (Telecommun. Comput. Electron. Control), 2021, 19(1): 244-251.

[5] Park J, Woo S, Lee J Y, et al. A simple and light-weight attention module for convolutional neural networks. International Journal of Computer Vision, 2020, 128(4): 783-798.

[6] Wang Y, Yan J, Yang Z, et al. GIS partial discharge pattern recognition via lightweight convolutional neural network in the ubiquitous power internet of things context. IET Science, Measurement & Technology, 2020, 14(8): 864-871.

[7] Fooladgar F, Kasaei S. Lightweight residual densely connected convolutional neural network. Multimedia Tools and Applications, 2020, 79(35): 25571-25588.

[8] Yu Y, Zhao T, Wang K, et al. Light-OPU: An FPGA-based overlay processor for lightweight convolutional neural networks. Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2020: 122-132.

[9] Guo H, Bai H, Zhou Y, et al. DF-SSD: a deep convolutional neural network-based embedded lightweight object detection framework for remote sensing imagery. Journal of Applied Remote Sensing, 2020, 14(1): 014521.

[10] Kyrkou C, Theocharides T. EmergencyNet: Efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2020, 13: 1687-1699.