Distracted Driving Behavior Recognition Based on Improved MobileNetV2

Xuemei Bai,[a*] Jialu Li,[a] Chenjie Zhang,[a] Hanping Hu,[b] Dongbing Gu[c]

[a] Changchun University of Science and Technology, School of Electronic Information Engineering, Weixing Road, Changchun, China, 130000

[b] Changchun University of Science and Technology, School of Computer Science and Technology, Weixing Road, Changchun, China, 130000

[c] University of Essex, School of Computer Science and Electronic Engineering, Colchester, UK

Abstract. In recent years, research on distracted driving behavior recognition has made significant progress, with an increasing number of researchers focusing on deep learning-based algorithms. Aiming at the problems of the existing distracted driving recognition algorithm, such as its oversized model and difficulty in adapting to low computing environments, a lightweight network MobileNetV2, is chosen as the backbone network and improved to design a distracted driving behavior detection method that is both accurate and practical. The Ghost module is employed to replace point-by-point convolution to reduce the computation, the Leaky ReLU function helps mitigate the problem of dead neurons, as it prevents gradients from becoming zero for negative inputs. Finally,the channel pruning algorithm is used to further reduce the model parameters.The experiment results on the State Farm dataset show that the model's test accuracy can reach 94.66% and the number of parameters is only 0.23M. The improved model in this paper has significantly fewer parameters than the baseline model, which demonstrates the effectiveness and applicability of the method.

Keywords: Channel Pruning,Deep Learning,Distracted Driving,Ghost Module,MobileNetV2.

***First Author,** E-mail: baixm@cust.edu.cn

1 Introduction

The World Health Organization estimated that 1.3 million people worldwide died in traffic-related incidents every year, and another 20–50 million got non-fatal injuries as a result. Individuals, families, and entire nations suffered significant economic losses as a result of traffic accidents, which accounted for 3% of the gross domestic product in the majority of nations. The analysis of causes and prevention of traffic accidents have been a prominent area of research by specialists to reduce the frequency of traffic accidents. Since most traffic accidents are caused by automobiles, reducing the number of traffic accidents caused by automobiles is a current problem that has to be addressed. According to statistics, most traffic casualties are caused by improper driving behavior or irregularities of drivers. Drivers may be distracted by glancing at their mobile phones or chatting on the phone while driving due to a lack of safety awareness, which raises the possibility of traffic accidents. Drivers who use mobile phones are about four times more likely to be involved in a collision than those who do not use them. According to research conducted by the Road Traffic Safety Research Center of the Ministry of Public Security, drivers who are distracted while driving can experience a significant lag in their reactions. For instance, a driver's reaction time to an unexpected scenario is 0.3–1 second in a normal state, but it can be up to three times longer in a distracted state, which significantly reduces the driver's capacity to manage risk. Thus, one of the main reasons for road accidents is distracted driving[1-3].

As you can see by this article[4],the author evaluated different deep learning approaches for driver distraction recognition. The results showed that deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), outperformed traditional machine learning methods, providing empirical evidence for the superiority of deep learning algorithms. On the other hand,The State Farm Distracted Driver Detection Kaggle competition, sponsored by State Farm, aimed to develop models for identifying driver distraction from images. The large participation and success of deep learning-based approaches in achieving top leaderboard positions highlight the effectiveness and popularity of deep learning techniques in this domain. These studies and statistics have shown the increasing interest and adoption of deep learning-based research methods in distracted driving behavior recognition. Researchers have begun to use deep learning approaches to recognize distracted driving behavior as a result of the rapid growth of deep learning.Additionally, many scholars have begun to create their datasets. Deep learning-based research methods have gained popularity among academics. High identification rates can be achieved by using in-vehicle cameras to capture the driver's driving process and pre-trained neural network models to detect and recognize the captured images.However, most research on distracted driving behavior identification has concentrated on recognition accuracy, with little attention paid to the lightweight nature of distracted driving behavior recognition models. To solve the above problem, the Ghost module and Leaky ReLU function are added to MobileNetV2, and channel pruning is then carried out.Compared with the initial model, the improved model parameters were significantly reduced and the computational efficiency of the neural network was improved, enabling end-to-end driver distracted driving behavior detection[5].

The organization of this paper is as follows. Section 2 focuses on the work of distracted driving. In Section 3, the model and improvement scheme chosen for the study are presented, including the introduction of the Ghost module、 the Leaky ReLU function and the channel pruning algorithm used for the improved model. Section 4 describes the experimental environment, the dataset and the experimental results. Section 5 provides the conclusion.

2 Related Work

2.1 Detection Methods

Distracted driving includes activities like talking or texting on the phone, eating or drinking, conversing with passengers, and adjusting audio or navigation systems, which divert attention from safe driving.For the sake of this study, distracted driving can be divided into three categories: cognitive distractions, visual distractions and motor distractions[6].

There are numerous methods for detecting distracted driving, and most researchers have used cameras to identify distractions while driving. Information such as head posture, mouth movements and eye direction can be used to assess the driver's attention at the current moment. Researchers have also used microphones to detect driver attention and exhaustion,as well as Electroencephalogram(EEG), Electrocardiogram(ECG) and other comparable physiological sensors to evaluate the driver's physiological and emotional condition and determine the driver's current distraction.

2.2 Convolutional Neural Networks and Deep Learning

Convolutional neural networks were proposed as a result of research into the visual mechanisms of living creatures, and it is a well-known deep learning architecture[7-8]. The development of the AlexNet framework in 2012 resulted in a breakthrough in image recognition, followed by ResNet, VGGNet, GoogleNet and other network structures as computing power increased. Convolutional neural networks have increasingly becoming more utilized in image classification since then.Deep learning algorithms have been increasingly popular and acknowledged in recent years, owing to their practicality, and almost all deep learning methods are based on CNNs. Deep learning can solve complicated issues using massive volumes of data, and raising the model's complexity can improve its problem-solving capabilities[9].

2.3 Lightweight Network Models

Due to their restricted hardware capabilities, embedded and mobile devices cannot handle computation-intensive and storage-intensive deep neural networks. At the same time, however, there is an increasing demand for deep learning in mobile scenarios such as mobile phones and cars. Therefore, it becomes crucial to compress and accelerate deep neural networks so that they can be deployed on mobile devices. Model compression and lightweight model design are the most common solutions to the aforementioned difficulties.

The core of lightweight networks is to lighten the network in terms of both size and speed while maintaining accuracy as much as possible. The enormous number of classic convolutional neural network model parameters, the deeper layers, and the difficulty in training limit the applicability of convolutional neural networks. In 2016, the Squeeze Net stack was proposed by Iandola, which used the Fire module to attain the approximate accuracy of Alex Net on the Imagenet dataset , but with 50 times fewer parameters. Google suggested MobileNetV1 in 2017, which could be implemented on mobile devices. In the same year, Zhang et al. proposed ShuffleNet, which introduced grouped convolution in deep separable convolution,greatly improving network performance[10-13].

MobileNetV2 is chosen as the backbone network model in the paper because it can assure excellent recognition accuracy while lowering the number of parameters.

2.4 Model Compression Methods

With the growing popularity of deep learning techniques, the demand for deep learning models has increased, because of an increased interest in models with a small memory and low computing resource requirements while maintaining a high level of accuracy. Model compression and acceleration for deep learning using neural network redundancy have sparked considerable interest in academia and industry[14-18].

Deep learning model compression and acceleration refer to a simplified model with fewer parameters and more simplified structure with the redundancy of neural network parameters and network structure, without affecting the degree of task completion. The compressed model requires less computing resources and memory, and it can meet a wider range of application requirements than the original model.

To achieve this effect, various methods have been proposed to compress network models. Two major categories may be used to categorize compression techniques,one is compression of existing networks such as tensor decomposition, model pruning and weight quantization,the other is construction of smaller networks such as knowledge distillation and compact structre design. To reconstruct and compress deep neural networks in this study, a pruning strategy relevant to image classification issues is adopted, which can drastically minimize compute and memory usage.

Neural network pruning first appeared in the 1990s,and LeCun's Optimal Brain Damage employed information-theoretic theories to remove irrelevant content from the network to improve its learning and classification performance. In 1993, Hassibi proposed the optimal brain surgery method, a second-order method for neural network pruning to remove the unimportant parts of the network. Song et al. proposed the well-known Deep Compression method,which effectively combined pruning, neural network quantization and Hoffman coding of the network model. In 2016, Li proposed a new pruning method that first identified the filters in the convolutional neural network that had a low impact on the final output accuracy, and then pruned the unimportant filters identified. Deleting irrelevant filters and their corresponding feature maps could significantly reduce network size and achieve network lightweight[19-23].

3 Network Framework

3.1 Improvements of MobileNetV2

The majority of algorithmic research on recognizing distracted driving behavior in recent years has mainly focused on increasing accuracy rates, ignoring the significance of lightweight. The goal of this study is to create a distracted driving detection algorithm that is highly accurate, highly efficient and memory-efficient. The lightweight network model MobileNet V2 is selected as the driving behavior recognition model,and the point-by-point convolution is replaced with the Ghost module to reduce a large number of floating point operations[24-26]. The Leaky ReLU function takes the place of the original activation function to prevent neuronal death. Table 1 illustrates the general organization of the improved model.

Table 1 Improved model structure

Input	Operator	t	c	n	s
224^2×3	Conv2d	-	32	1	2
112^2×32	Improved bottleneck	1	16	1	1
112^2×16	Improved bottleneck	6	24	2	2
56^2×24	Improved bottleneck	6	32	3	2
28^2×32	Improved bottleneck	6	64	4	2
14^2×64	Improved bottleneck	6	96	3	1
14^2×96	Improved bottleneck	6	160	3	2
7^2×160	Improved bottleneck	6	320	1	1
7^2×320	Conv2d	-	1280	1	1
7^2×1280	Avgpool	-	-	1	-
1×1×1280	Conv2d	-	k	-	-

Note: "-" means the value does not exist there and "k" is any positive integer value. A series of one or more identical layers that are repeated n times is described by each line. "c" means output channels.

3.1.1 MobileNetV2

MobileNetV2 is a lightweight neural network, which is improved from MobilenetV1. Its main features are as follows.

1) The advantages of MobileNetV1 are followed, and depth-separable convolution is employed instead of conventional convolution to reduce computing cost and the amount of model parameters;

2) The main structure of the model consists of a bottleneck layer of reverse residuals, which deepens the network structure and enhances the expression of features;

3) The normal ReLU activation function is replaced with the ReLU6 nonlinear activation function since it is more reliable in the low-precision calculation. To lessen the loss of information about low-dimensional features, the linear bottleneck layer is added to the point-by-point convolution.

The reverse residual bottleneck layer's primary job is to extract features of the input image. To begin, point-by-point convolution (PW) is used to up-dimension the input image, followed by depth convolution (DW) with a convolution kernel of 3*3 for high-dimensional feature extraction, and finally one-dimensional convolution is used to downscale the output, which can increase the depth of the network and better extract useful information without increasing the number of operations and parameters. Fig. 1 depicts the inverted residual structure.

According to Fig. 1, when the depth convolution step is 1, the output feature size is the same as the input one, and the input and output elements are summed by the inverse residual connection; when the depth convolution step is 2, the downsampling operation is performed without the residual connection.

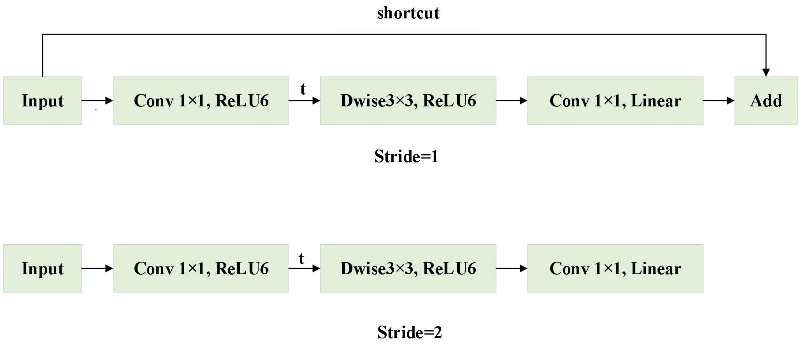


Fig. 1 Initial inverted residual structure
Note: "t" is the expansion factor.

3.1.2 Ghost module

The Ghost module is a lightweight neural network implementation method that enables deep neural networks to be transferred to some mobile devices with relatively small processing capacity while maintaining algorithmic expressiveness. The Ghost module splits the original convolutional layer into two parts. The first part is an ordinary convolutional operation, but their number will be strictly controlled. The second part is a series of simple linear operations for generating more feature maps based on the feature maps from the first part. In this way, without changing the size of the output feature maps, the Ghost module reduces the number of parameters required and the computational complexity with respect to traditional convolutional neural networks.The Ghost module is able to generate more features while meeting resource constraints, thus improving the efficiency and performance of the network. The schematic diagram of the Ghost module is shown in Fig. 2, where Φ represents the cheap operation, which has a considerably lower computational cost per channel than standard convolution.

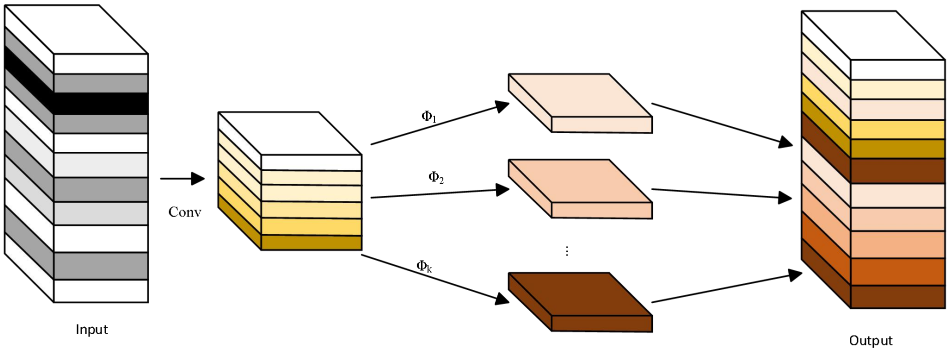


Fig. 2 Schematic of Ghost Module

Although the computational complexity and model size of MobileNetV2 is reduced with the deep separable convolution, the point-by-point convolution still requires a significant amount of computing power due to the high channel count. To address the problem of intensive computation of the point-by-point convolution, this study uses the Ghost module to replace the point-by-point convolution in MobileNetV2,and the structure of the inverse residuals with the ghost module is shown in Fig. 3.

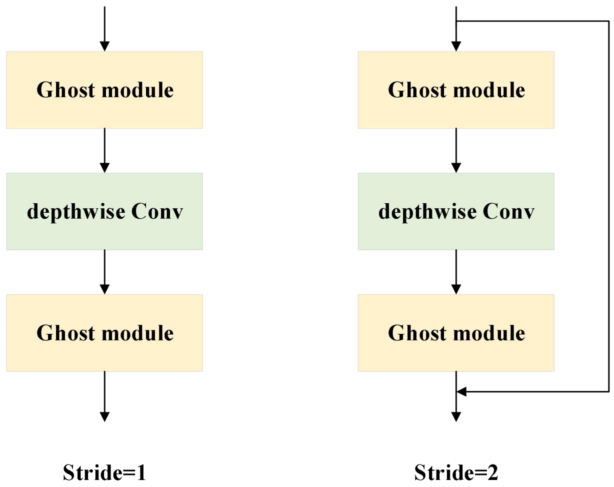


Fig. 3 Bottleneck structure with Ghost module added

3.1.3 Leaky ReLU function

In neural networks, the activation function is used to incorporate nonlinear elements and enhance the model's expressiveness. The ReLU function's equation is displayed in (1). It is an activation function that is frequently used for convolutional neural networks. In the x>0 region, gradient saturation and gradient disappearance will not occur and the computational complexity is low. However, when x ≤ 0, the gradient is 0, and the gradient of this neuron and the subsequent neurons is always 0, and no longer responds to any data, resulting in the corresponding parameters never being updated, that is, the neuron is necrotic. The Leaky ReLU function introduces α as the gradient when x ≤ 0 on the basis of the ReLU function with the formula shown in (2). It can simultaneously prevent neuronal necrosis and enhance the gradient.Fig. 4 depicts the overall structure with the Leaky ReLU funciton. Leaky ReLU can be used behind each convolutional layer of MobileNetV2 to replace the original ReLU function. Leaky ReLU accelerates the convergence of the neural network compared to the ReLU function.

$$ReLU(x) = \max(0, x) \ \#(1)$$

$$LeakyReLU(x) = \begin{cases} x, \ x > 0 \\ \alpha x, \ x \leq 0 \end{cases} \#(2)$$

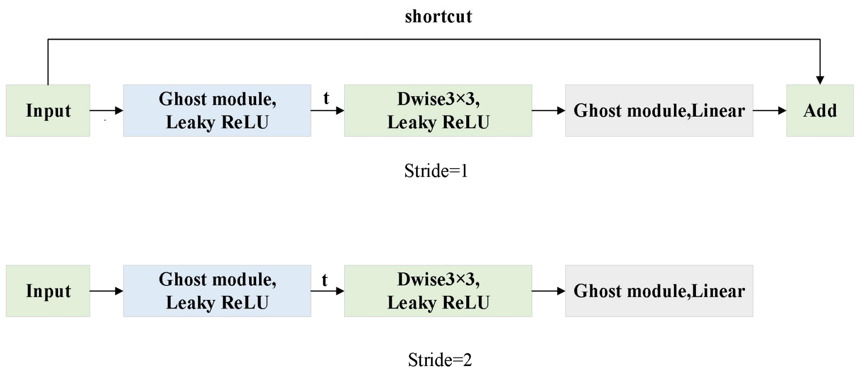


Fig. 4 Improved MobileNetV2 bottleneck structure

3.2 Pruning Algorithm

The steps of the pruning algorithm employed in this paper are as follows. First, to find the unimportant channels in the network, the modified MobileNetV2 with sparse scaling coefficients is trained to obtain a model with sparse scaling coefficients. Then, in the pruning stage, the channels corresponding to the scaling factors below the threshold are pruned to obtain the pruned network, and the pruned network is retrained to compensate for the accuracy loss caused by pruning.Fig. 5 depicts the channel pruning implementation flow.

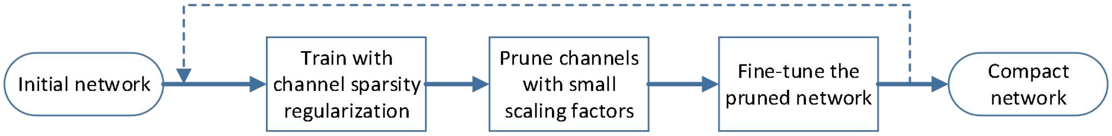

Initial network
Train with channel sparsity regularization
Prune channels with small scaling factors
Fine-tune the pruned network
Compact network

Fig. 5 Channel pruning algorithm flow

3.2.1 Sparse training

The channel sparsity of the deep model facilitates channel pruning and obtains the number of channels with low importance that are likely to be pruned. According to the channel pruning principle, each channel in each convolutional layer is given a scaling factor γ as an important basis for channel pruning. This scaling factor is multiplied with the input of that channel to produce various effects on the extracted features of each channel in each layer, and the absolute value of the scaling factor indicates the importance of the channel. Assume Z_{in} and Z_{out} denote the input and output of the BN (Batch Normalization) layer; B denotes the current mini-batch, and the transformation of the BN layer is as (3), where μ_B and σ_B are the mean and standard deviation of B, respectively;ε and β are the trainable hyperparameters of the BN layer.

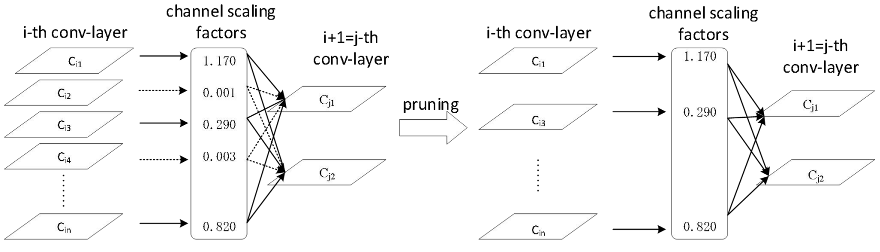
$$\hat{z} = \frac{z_{in} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}; \ z_{out} = \gamma\hat{z} + \beta\#(3)$$

Meanwhile, the penalty term related to γ is introduced based on the variation of parameters in the network, and the sparse training loss function is shown in Equation (4), where (x, y) denotes the input and output of the network, w denotes the weights, the first term denotes the loss during training of the original network, the second term denotes the L1 regularization concerning γ, g(γ) denotes the sparse induction penalty of the scaling factor, λ denotes the hyperparameter that balances the ratio of the normal training loss to the loss of the channel scaling factor penalty term, and Γ denotes the set of values of the scaling factor γ.

$$L = \sum_{(x,y)} l(f(x, W), y) + \lambda \sum_{\gamma\in \Gamma} g(\gamma) \#(4)$$

3.2.2 Channel pruning

Fig. 6 illustrates the pruning process of the standard convolutional layer of the network. The channel scaling factors obtained after sparse training are shown in Fig. 6(a). The current scaling factor γ exhibit sparsity, and the scaling factors of channels 2 and 4 in the i-th convolutional layer are approximately 0. This means that the trained model believes that the features extracted by the channel have little effect on target recognition and classification. After pruning, the channel is deleted, and the convolution kernel and its parameters will not be saved, which achieves the effect of lightening the model. The model after pruning is shown in Fig. 6(b).



(a) initial network

(b) compact network

Fig. 6 Schematic diagram of channel pruning principle

4 Experimental results and analysis

4.1 Experimental Environment and Parameter Configuration

Table 2 Experiment environment configuration

Type	**Specific parameter**
CPU	AMD EPYC 7543 32-Core Processor
GPU	RTX 3090
Cuda Version	11.3
Python Version	3.8
Torch Version	1.10.0
Torchvision Version	0.11.1

Table 2 depicts the experimental setting. The Windows 10 Professional operating system was used, and all experiments were built to the same specifications to ensure that the experimental comparisons were fair. The batch_size was set to 16, the number of iterations was set to 100, Adam was used as the optimizer, the initial learning rate was set to 0.0001, CrossEntropyLoss was used as the loss function, and CrossEntropy was used to calculate cross-entropy in Pytorch.

4.2 Dataset Introduction and Evaluation Metrics

To evaluate the performance of this study in detecting distracted driving behavior, the experiment used the CIFAR-10 dataset and the State Farm dataset. The CIFAR-10 dataset consists of 10 categories, each of which has 6000 photos, for a total of 60,000 color RGB images in the entire dataset. The CIFAR-10 dataset is divided into training and test sets, with 50,000 photos serving as training images and 10,000 images serving as testing images.

The State Farm dataset contains 10 categories of actions.They are normal driving, left-handed texting, left-handed phone call, right-handed texting, right-handed phone call, operating the radio, drinking water, turning the body back, finishing the face and talking to the passenger. In Fig. 7, the images for each category are displayed. This is a competition dataset on the platform Kaggle and is the first publicly downloadable dataset for distracted driving behavior recognition.It contains 22,424 annotated photographs with an image size of 480 × 460. The training set and the test set are divided by 8:2. The size of this dataset is uniformly modified to 224×224 in this study.

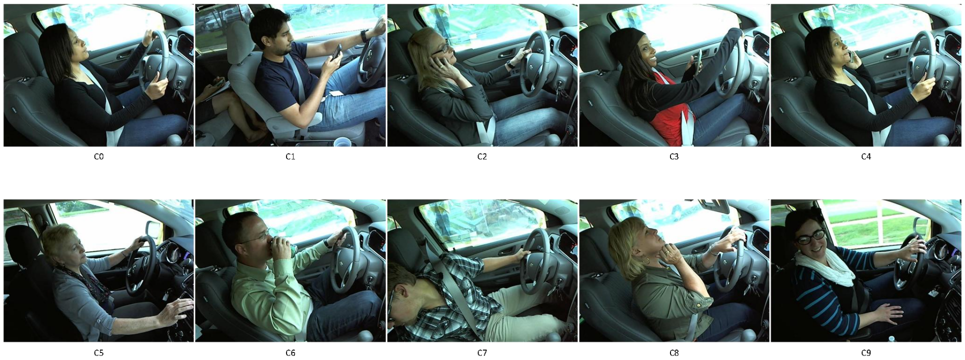


Fig. 7 Images of different categories of the State Farm dataset

The evaluation metrics in this study include accuracy, the number of model parameters, weight file size and the number of floating-point operations.The complexity of the model is determined by the number of its parameters. Typically, the more parameters, the more complicated and computationally intensive the model is. To some extent, the amount of floating-point operations can indicate how quickly the model can conclude. The more operations, the slower the model can conclude. Smaller-weight files require less storage on embedded systems and are less expensive to utilize since the memory of embedded devices is restricted. The weight file size is in megabytes (MB), and the parameter count is in millions (M).

4.3 Data Preprocessing

Before the images were fed into the network model, they need to be preprocessed. First, the images were randomly cropped and resized to 224 × 224 pixels to match the input requirements of the model, then horizontally flipped. Next, the images were transformed into a Pytorch tensor and finally normalized by subtracting the mean [0.485, 0.456, 0.406] and dividing by the standard deviation [0.229, 0.224, 0.225] to normalize the image tensor.

4.4 Analysis of Experimental Results

4.4.1 Comparison of model recognition performance

To reflect the superiority of the MobileNetV2 backbone network, a variety of popular neural networks were selected for experimental comparison. ResNet, VGG and other well-known models that were trained on the CIFAR-10 dataset are included in the experimental models. To maintain track of the models' training conditions and make sure that each model finishes the training in a converged state, the test set accuracy and training loss values of the models were recorded for each training cycle during the training process.

Table 3 displays the training outcomes for each model on the dataset. Although the MobileNet v2 network's accuracy was marginally lower than the traditional network's, it had the least size of weight files and parameters, which was significantly better than the traditional network's and had a very good training effect, demonstrating the performance benefit of selecting such a light network.

Table 3 Performances of different networks on the CIFAR-10 dataset

Model	**Weighting file/MB**	**Acc/%**	**Params/M**
VGG	152.9	92.36	138.36
Resnet	81.3	92.5	21.80
Densenet	27.1	89.2	7.98
MobileNetV2	8.8	91.59	2.24

4.4.2 Ablation experiments

To prove that each improvement scheme of the algorithm had a gain effect on the original algorithm, the ablation experiment was designed to analyze and compare the algorithm. In this experiment, the model was trained for 100 iterations, the size of the weight file and the model's accuracy in the test set were used as assessment indicators, and the following specific experimental variables were controlled:

Experiment 1:Initial MobileNetV2 network.

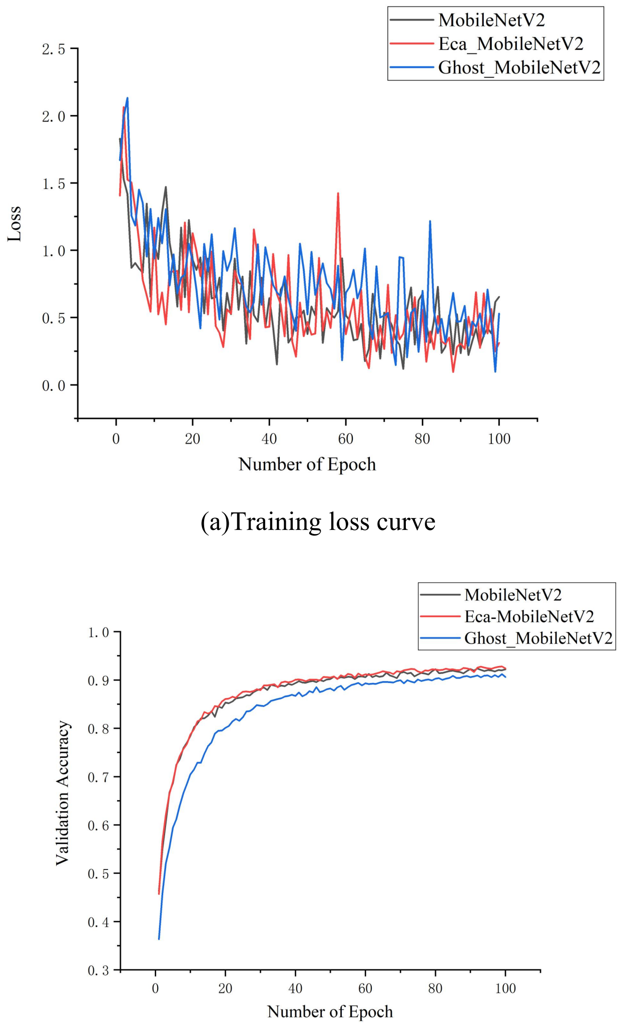
Experiment 2: ECA (Efficient Channel Attention) was added to MobileNetV2.

Experiment 3:The model using the improved scheme of this paper.

Table 4 shows the comparative results of the above ablation experiments.It can be observed that adding the Ghost module and Leaky ReLU function to MobileNet v2 can compromise some accuracy to speed up the model's inference. The accuracy of the model was reduced by 0.43%, but the number of model parameters was only 85% of the original one. The experimental training process is shown in Fig. 8.

Table 4 Comparison results of ablation experiments

Model	**ECA**	**Ghost**	**Leaky ReLU**	**Acc/%**	**Weighting file/MB**
MobileNetV2	-	-	-	90.8	8.8
ECA-MobileNetV2	√	-	-	90.05	8.8
Ghost_MobileNetV2	-	√	√	91.23	7.5



(a)Training loss curve

(b)The accuracy vaiation curve

Fig. 8 Model training process diagram

4.4.3 Model pruning experiments

This section verified the impact of the chosen channel pruning algorithm on the VGG, Resnet, MobileNet and other networks to better understand the impact of structured pruning on various networks. The results of the above networks' pruning are shown in Table 5. As can be seen, the VGG and Resnet networks' accuracy reached after pruning to 92% and 92.1%, respectively, but their floating point numbers remained high. In contrast, Ghost_MobileNetV2's accuracy decreased from the original model by 1.11%, which was within acceptable bounds, and their floating point numbers decreased by 95.22. The number of parameters was 23% of the original model. As a result, we can observe from the experimental data that the modified model using the pruning technique is effective.

Table 5 Parameters of the pruned model

Model	Acc/%	FLOPs/M	Params/M
VGG	92.0	152.81	1.67
Resnet	92.1	975	6.00
MobileNetV2	91.5	126.5	0.98
Ghost_MobileNetV2	90.39	31.28	0.23

4.4.4 Performance on State Farm dataset

In order to prove the superiority of this algorithm over other algorithms, this experiment was conducted in the same environment to compare MobileNetV2 and the final improved algorithm in this paper. The performance of the improved algorithm and the initial algorithm is shown in Table 6.When the improved approach described in this paper was applied to the Statefarm dataset, the findings revealed that it outperforms MobileNetV2. As can be seen by comparing with the initial model, although the accuracy has decreased by 3.55%, it still met the design goals of this paper for the driving behavior recognition task. Both the floating point number and the number of parameters decreased substantially, with the number of parameters decreasing by 2.01 and the floating point number decreasing by 176, which was better than the common structure.Under the condition of a slight drop in accuracy, the improved MobileNetV2 had a large decrease in both the number of parameters and the floating point count. The comparison of the test accuracies is shown in Fig. 9.

Table 6 Performances of the State Farm dataset

Model	Acc/%	FLOPs/M	Params/m
MobileNetV2	98.01	313.5	2.24
Slimming_Ghost_ MobileNetV2	94.66	137.5	0.23

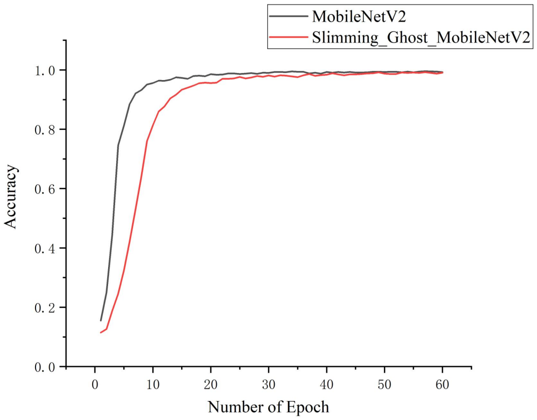


Fig. 9 Comparison of test accuracy

5 Conclusion

In this paper, MobileNetV2 is employed as the backbone model to implement driving behavior detection, and this algorithm is improved. From the perspective of lightweight, the introduction of the Ghost module and Leaky ReLU functions in the trunk reduce the calculation amount without affecting the accuracy, to enhance the model’s ability to extract useful feature information. The channel pruning algorithm considerably reduced the number of model parameters and made it possible to deploy the model on edge devices with low processing power. The performance of MobileNetV2 depends heavily on the dataset used for training and evaluation. If the dataset is insufficient or the sample distribution in the dataset does not match the actual application scenarios, the performance of the algorithm may be limited. The algorithm's ability to generalize to different driving scenarios and behaviors can be improved through reasonable data enhancement techniques and migration learning methods.The improved algorithm on the State Farm dataset achieved low consumption and high accuracy in the detection of distracted driving behavior, which had academic significance and practical application value.

Acknowledgement

This work was supported by the Department of Science and Technology of Jilin Province, China, project number: 20220402025GH

Code, Data, and Materials Availability

The code used in this study is not public due to the project requirement.

References

1. Wilson, Fernando A., and Jim P. Stimpson. "Trends in fatalities from distracted driving in the United States, 1999 to 2008." American journal of public health 100.11 (2010): 2213-2219. [doi: 10.2105/AJPH.2009.187179]
2. Chan, Michelle, and Anthony Singhal. "Emotion matters: Implications for distracted driving." Safety science 72 (2015): 302-309. [doi:10.1016/j.ssci.2014.10.002]
3. Sherif, Bassel, Hatem Abou-Senna, and Essam Radwan. "Distracted driving effects on headways at signalized intersections." *Transportation research record* 2677.3 (2023): 738-756.
4. Li, Wanjun, Konstantina Gkritza, and Chris Albrecht. "The culture of distracted driving: Evidence from a public opinion survey in Iowa." Transportation research part F: traffic psychology and behaviour 26 (2014): 337-347.[doi:10.1016/j.trf.2014.01.002]
5. Sahoo, Goutam Kumar, Santos Kumar Das, and Poonam Singh. "A deep learning-based distracted driving detection solution implemented on embedded system." *Multimedia Tools and Applications* 82.8 (2023): 11697-11720.[doi:https://doi.org/10.1007/s11042-022-13450-6]
6. Zou, Zhengxia, et al. "Object detection in 20 years: A survey." arXiv preprint arXiv:1905.05055 (2019). [doi:10.48550/arXiv.1905.05055]
7. Liu, Li, et al. "Deep learning for generic object detection: A survey." International journal of computer vision 128.2 (2020): 261-318.

8. Xiao, Youzi, et al. "A review of object detection based on deep learning." Multimedia Tools and Applications 79.33 (2020): 23729-23791. [doi:https://doi.org/10.1007/s11042-020-08976-6]
9. Mohan, Anuj, Constantine Papageorgiou, and Tomaso Poggio. "Example-based object detection in images by components." IEEE transactions on pattern analysis and machine intelligence 23.4 (2001): 349-361. [doi:10.1109/34.917571]
10. Megat-Johari, Nusayba, et al. "Evaluation of enforcement and messaging campaign focused on reducing cell phone-related distracted driving." *Transportation research record* 2677.1 (2023): 1741-1752.
11. Lu, Dengsheng, and Qihao Weng. "A survey of image classification methods and techniques for improving classification performance." International journal of Remote sensing 28.5 (2007): 823-870. [doi:10.1080/01431160600746456]
12. Ping, Peng, et al. "Distracted driving detection based on the fusion of deep learning and causal reasoning." *Information Fusion* 89 (2023): 121-142.
13. Sánchez, Jorge, et al. "Image classification with the fisher vector: Theory and practice." International journal of computer vision 105.3 (2013): 222-245. [doi:https://doi.org/10.1007/s11263-013-0636-x]
14. Rawat, Waseem, and Zenghui Wang. "Deep convolutional neural networks for image classification: A comprehensive review." Neural computation 29.9 (2017): 2352-2449. [doi:10.1162/neco_a_00990]
15. Chen, Junde, Defu Zhang, and Yaser Ahangari Nanehkaran. "Identifying plant diseases using deep transfer learning and enhanced lightweight network." Multimedia tools and applications 79.41 (2020): 31497-31515. [doi:https://doi.org/10.1007/s11042-020-09669-w]
16. Chen, Huiqin, et al. "Towards Sustainable Safe Driving: A Multimodal Fusion Method for Risk Level Recognition in Distracted Driving Status." *Sustainability* 15.12 (2023): 9661.
17. Calvert, Kenneth L., James Griffioen, and Su Wen. "Lightweight network support for scalable end-to-end services." ACM SIGCOMM Computer Communication Review 32.4 (2002): 265-278.

18. Kozlov, Alexander, Vadim Andronov, and Yana Gritsenko. "Lightweight network architecture for real-time action recognition." Proceedings of the 35th Annual ACM Symposium on Applied Computing. 2020. [doi:10.48550/arXiv.1905.08711]
19. Fukui, Hiroshi, et al. "Attention branch network: Learning of attention mechanism for visual explanation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019. [doi:10.48550/arXiv.1812.10025]
20. Li, Wei, et al. "Object detection based on an adaptive attention mechanism." Scientific Reports 10.1 (2020): 1-13. [doi:10.1038/s41598-020-67529-x]
21. Cheng, Yu, et al. "Model compression and acceleration for deep neural networks: The principles, progress, and challenges." IEEE Signal Processing Magazine 35.1 (2018): 126-136. [doi:10.1109/MSP.2017.2765695]
22. Liu, Zhuang, et al. "Rethinking the value of network pruning." arXiv preprint arXiv:1810.05270 (2018). [doi:10.48550/arXiv.1810.05270]
23. Jiang, Yuang, et al. "Model pruning enables efficient federated learning on edge devices." IEEE Transactions on Neural Networks and Learning Systems (2022). [doi:10.48550/arXiv.1909.12326]
24. Ma, Hui, Turgay Celik, and Heng-Chao Li. "Lightweight attention convolutional neural network through network slimming for robust facial expression recognition." Signal, Image and Video Processing 15.7 (2021): 1507-1515. [doi:10.1007/s11760-021-01883-9]
25. Liu, Jing, et al. "Discrimination-aware network pruning for deep model compression." IEEE Transactions on Pattern Analysis and Machine Intelligence (2021). [doi:10.1109/TPAMI.2021.3066410]
26. Sarıgül, Mehmet, Buse Melis Ozyildirim, and Mutlu Avci. "Differential convolutional neural network." Neural Networks 116 (2019): 279-287. [doi:10.1016/j.neunet.2019.04.025]

Xuemei Bai received her PhD from Changchun University of Science and Technology, Changchun, China, in 2009. She is currently a professor at School of Electronic Information

Engineering, Changchun University of Science and Technology. Her current research interests include intelligent information processing and pattern recognition.

Jialu Li is a master student in the school of Electronic Information Engineering,Changchun University of Science and Technology.Her research interest includes Signal and Information Processing.

Chenjie Zhang received her master's degree from Changchun University of Science and Technology in 2008. She is currently an associate professor at School of Electronic Information Engineering, Changchun University of Science and Technology. Her current research interests include intelligent information processing and pattern recognition.

Hanping Hu received his PhD from Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, China, in 2015. He is a laboratory teacher in the School of Computer Science and Technology, Changchun University of Scence and Technology. His current research interests include deep learning and pattern recognition.

Dongbing Gu received his PhD from University of Essex, UK in 2004. He is currently a professor in the School of Computer Science and Electronic Engineering. His main research interests include robotics, autonomous systems, machine learning, multi-robot systems, cooperative control, SLAM, UAVs, and process automation.

Caption List

Table 1 Improved model structure

Table 2 Experiment environment configuration

Table 3 Performances of different networks on the CIFAR-10 dataset

Table 4 Comparison results of ablation experiments

Table 5 Parameters of the pruned model

Table 6 Performances of the State Farm dataset

Fig. 1 Initial inverted residual structure

Fig. 2 Schematic of Ghost Module

Fig. 3 Bottleneck structure with Ghost module added

Fig. 4 Improved MobileNetV2 bottleneck structure

Fig. 5 Channel pruning algorithm flow

Fig. 6 Schematic diagram of channel pruning principle

Fig. 7 Images of different categories of the State Farm dataset

Fig. 8 Model training process diagram

Fig. 9 Comparison of test accuracy