Journal of Computers Vol. 33 No. 5, October 2022, pp. 29-38
doi: 10.53106/199115992022103305003

Application of Lightweight Neural Network in Speed Bump Recognition of Autonomous Vehicle

Zhi-Yong Yang[1*], Zhen-Ping Mou[2], Long Wang[2], Yu Zhou[3]

[1] School of Big Data and Internet of Things, Chongqing Vocational Institute of Engineering, Chongqing 402260, China
zyy@cqvie.edu.cn
[2] School of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China
muzp555@163.com, 1490011375@qq.com
[3] School of Finance and Tourism, Chongqing Vocational Institute of Engineering, Chongqing 402260, China
67752142@qq.com

Received 2 November 2021; Revised 2 February 2022; Accepted 2 March 2022

Abstract. Vibration occurs when a vehicle passes through a speed bump, which has different intensities at different sizes and speeds. The recognition of speed bump type is an important step for vehicle to adjust speed automatically in time in automatic driving, which helps to improve the safety and comfort of passengers. In this paper, we put forward the technical requirements of speed bump image acquisition in automatic driving scene, and establish the speed bump image dataset. Based on improved EfficientNet basic block, we construct a lightweight convolutional neural network integrating edge detection, which is named Edge-Efficientnet. The experimental results show that its accuracy is improved by 3.3% and the model size is reduced by 53% compared with EfficientNetB0 model. In terms of computing speed, the model meets the real-time performance requirements. The Edge-Efficientnet model can be applied to the comfortable speed adjustment of autonomous vehicles passing through speed bump.

Keywords: CNN, deep learning, image classification, automatic driving

1 Introduction

Speed bump is a traffic safety facility widely distributed in urban roads and expressways. It forces the vehicle speed to decrease by forcibly causing vehicle vibration, so as to achieve the purpose of safe driving [1]. According to relevant research, the size of speed bump and vehicle speed are the key factors affecting the strength of vibration [2-5]. Under the excitation of the same speed bump, the vehicles in different speed ranges show three vibration states: periodic vibration, quasi periodic vibration and chaotic vibration. In the periodic vibration state, the comfort of passengers is the best, in the quasi periodic vibration state, the comfort of passengers is the worst, and speed bumps of different size correspond to different comfort speed ranges [6-7]. This effect is more obvious when encountering continuous speed bump. With the rapid development of artificial intelligence technology and intelligent control technology, automatic driving technology is developing towards a higher level. How to recognize the type of the speed bump efficiently and accurately and adjust to a comfortable speed in time according to the type is the key to improve the safety and comfort [8-10].

At present, traffic sign recognition technology based on deep learning is widely used in the field of automatic driving [11-14]. VGG, GooLeNet and ResNet models have achieved good results in image classification tasks [15-17]. These models usually have deeper network layers, larger model volume and longer computing time. However, the computational performance of the vehicle is relatively low, and the speed bump recognition task requires high real-time performance. In recent years, MobileNet, ShuffleNet, EfficientNet and other lightweight network models suitable for mobile terminals have emerged [18-24]. By changing the traditional convolution operation mode, these models realize the fast and lightweight network. However, in the automatic driving scene, due to the influence of distance, height, angle and other factors, the speed bump characteristics in the image are not obvious, and the effect of using these models directly is not good. In terms of data sets, with the rapid development of automatic driving applications, many institutions have released traffic sign datasets, including a variety of common traffic signs or facilities [25-26]. However, in these data sets, there is no more detailed classification of speed bump. In this paper, we first put forward the technical requirements of speed bump image data acquisition in automatic driving scene, and construct the image dataset. Then, based on the EfficientNet basic block of Google team, an algorithm suitable for fast extraction and classification of speed bump features is designed, and

* Corresponding Author

edge detection technology is integrated to enhance the performance of image features. Finally, experiments show that the model improves the accuracy and recognition efficiency on the premise of model reduction.

2 Speed Bump Image Dataset

2.1 Material and Type

In this study, the common speed bump types include hump speed bump, spike speed bump, thermoplastic vibration marking and color anti-skid deceleration pavement. We purchased 30 different hump speed bumps from manufacturers, which have different heights, widths, cross-sectional shapes, surface textures and color distributions, as shown in Fig. 1. Hump speed bump is assembled by multiple sections, for research convenience, we only use one section with a length of 1 meter. In Fig. 2, images of other types are collected on site. Compared with hump speed bump, they have lower height and fewer subtypes.

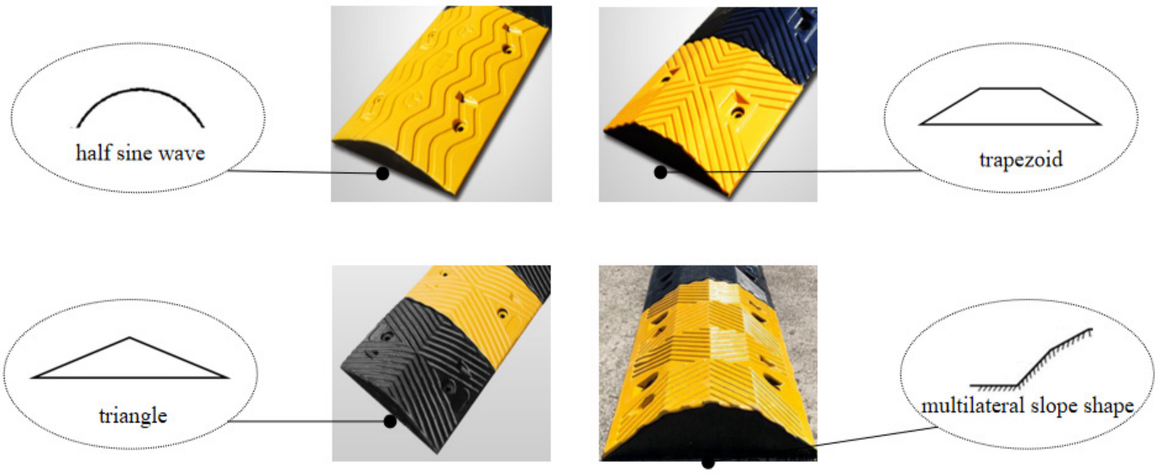


Fig. 1. Subtype of hump speed bump

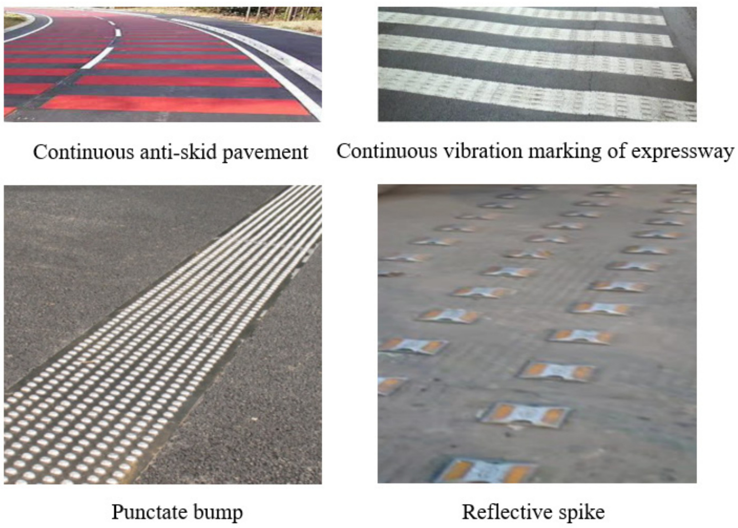


Fig. 2. Other types of speed bump except hump type

2.2 Technical Specification for Image Acquisition

The image acquisition shall conform to the automatic scene, where the direction and height of image acquisition are fixed. As shown in Fig. 3, suppose the camera height is h, the distance from the speed bump is s, and the camera depression angle is θ. The approximate relationship between the three is as follows:

$$Tan\left(90^{o} - \theta\right) = h / s. \quad \textbf{(1)}$$

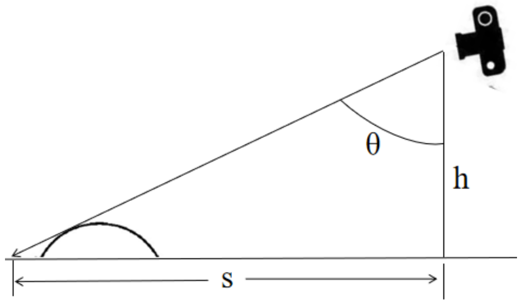


Fig. 3. Mathematical relationship among photographing distance, height and depression angle

When s becomes larger, θ will become larger, and the image lacks the contour and shape information in the vertical direction of the speed bump. On the contrary, when s is too small, there will not be enough distance for the car to adjust its speed. After field practice, the image acquisition specification is formulated by manually checking the image quality, as shown in Table 1.

Table 1. Operation specification for image acquisition of speed bump

Item	Specification
Camera pixel	5 million at least
Photographic distance	3m-9m
Camera height	1.5m
Camera angle	Straight ahead,15 degrees to the left and right

2.3 Dataset Development

We collected 5280 original images of speed bumps. All images are marked with data type by manual marking, and the dataset is divided into 12 subtypes as shown in Table 2.

Table 2. 12 types of speed bump collected so far

Type No.	Shape	Width (mm)	Height (mm)	Length (mm)	Remarks
1	Miniature	200	10	1000	hump
2	Half sine wave	300	30-40	1000	hump
3	Half sine wave	400	50-60	1000	hump
4	Triangle	300	30-40	1000	hump
5	Triangle	400	50-60	1000	hump
6	Trapezoid	300	40-50	1000	hump
7	Trapezoid	400	60-70	1000	hump
8	Multilateral slope shape	450	70	1000	hump
9	Reflective spike	400	20		
10	Punctate bump	400	5-7		
11	Continuous vibration marking of expressway		5-7		
12	Continuous anti-skid pavement		3-5		

Note: The surface textures and patterns of speed bumps with same shapes but different sizes are different.

3 CNN Model for Image Classification

3.1 Classification Principle of Speed Bump

The speed bump is classified according to the geometric features, including shape, texture, width, height and other elements. The whole project includes dataset construction, feature network design and model training. In Fig. 4, it shows the flow of speed bump classification. We design an edge detection module to enhance the edge information before entering the convolutional neural network. Then the image processed by edge detection enters the trained convolution neural network model for feature extraction, and finally connects with softmax classifier for feature matching.

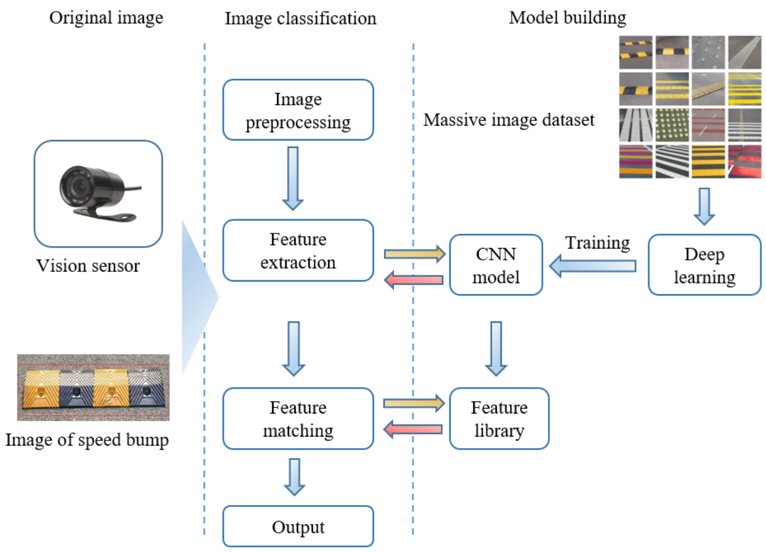


Fig. 4. Speed bump classification task flow

3.2 Design of CNN Model Combined with Edge Detection

As shown in Fig. 5, the classification algorithm mainly includes two modules: edge detection and feature extraction network.

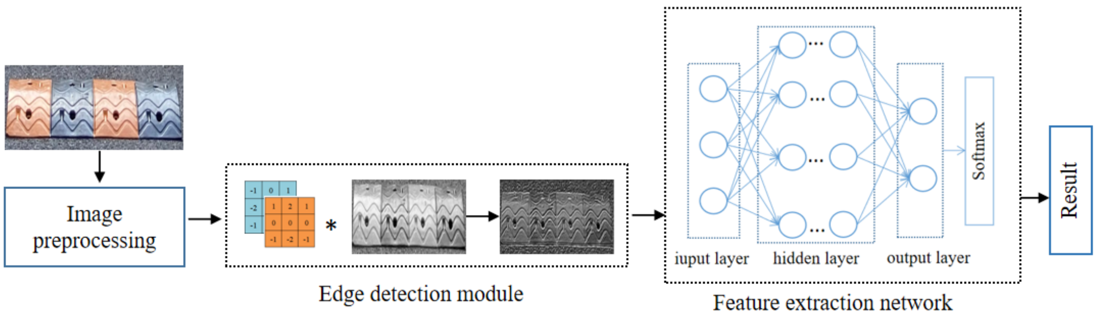


Fig. 5. Feature extraction and classification algorithm flow

3.2.1 Edge Detection Module

An edge detection module is added at the front end of the convolutional neural network to enhance the feature expression ability of the shape and texture of the speed bump image. Firstly, we preprocess the image, convert it into gray image, and resize the size to 380 * 380 pixels. Secondly, Sobel operator, an edge detection operator, is used to convolute the input matrix in X direction and Y direction. The convolution mode adopts “zero filling” and the image size remains unchanged. After the matrix is convoluted by Sobel operator, some pixels become negative numbers. In order to ensure the normal data range, it is necessary to convert them. The conversion rules are as follows:

$$\text{new\_pixel} = \begin{cases} |\text{src\_pixel}| & -255 < \text{src\_pixel} < 0 \\ 255 & \text{src}_{\text{pixel}} > 255 \text{ or } \text{src}_{\text{pixel}} < -255. \\ \text{src\_pixel} & 0 < \text{src\_pixel} < 255 \end{cases} \quad (2)$$

The converted convolution results in X and Y directions are fused according to the ratio of 50%: 50%. Finally, a sharpening filter operator is used to convolute the image matrix again. Fig. 6 shows the detailed flow of the edge detection module.

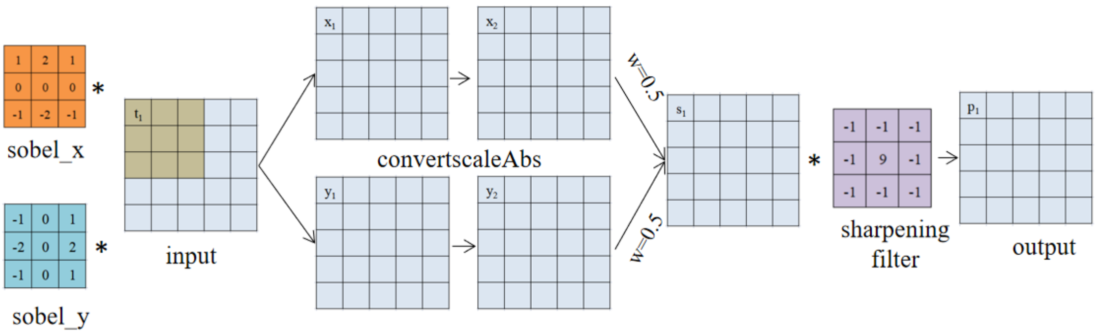


Fig. 6. Detailed flow of edge detection module

3.2.2 Structure of Feature Extraction Network

The feature extraction network adopts the EfficientNet basic block, as shown in Fig. 7, and it is improved. We change average pooling to maximum pooling to retain more shape features. Moreover, the DWConv kernel does not use 5x5 size, but all are set to 3x3 size to reduce the amount of network parameters. The DWConv convolution kernel uses the depthwise convolution mode, which is different from the traditional convolution operation. A depthwise convolution kernel is responsible for one channel, and a channel is convoluted by only one convolution kernel, which greatly reduces the amount of parameters.

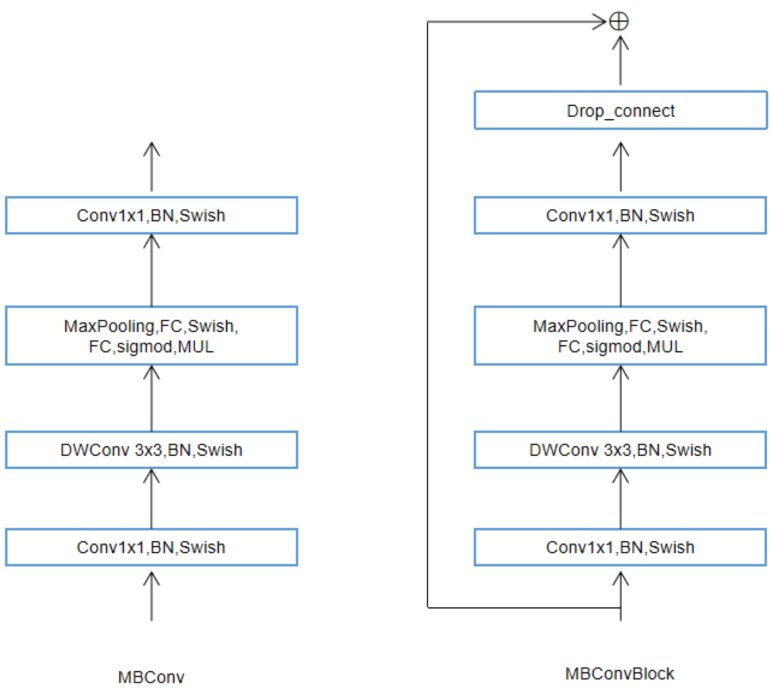


Fig. 7. Improved EfficientNet basic block

We named the feature extraction network as “Edge- EfficientNet”. The input data of the network is the output data of the edge detection module. As shown in Fig. 8, the network is composed of improved MBConv and MBConvblock stack, including 6 stages and 12 convolution layers. Stage1 contains a 3x3 size traditional convolution layer with a stride of 2, including BN and the activation function swish. Stage2 contains a MBConv with a stride of 1 and a MBConvBlock with a stride of 2. Stage3 contains an MBConv with a stride of 2 and two MBConvBlocks with a stride of 2. Stage4 contains an mbconv with a stride of 2 and three MBConvBlocks with a stride of 1. Stage5 is an MBConv with a stride of 1. Stage6 is a 1x1 size traditional convolution layer with a step of 1, including BN and the activation function swish. After stage 6, there is a pool layer and a full connection layer. Similarly, unlike EfficientNet, the last of the network uses maximum pooling instead of average pooling to retain more shape features. After the full connection layer is another full connection layer, which transforms the features extracted by neural network into feature mapping corresponding to 12 speed bump types. Finally, the softmax classifier is connected to realize the speed bump type output.

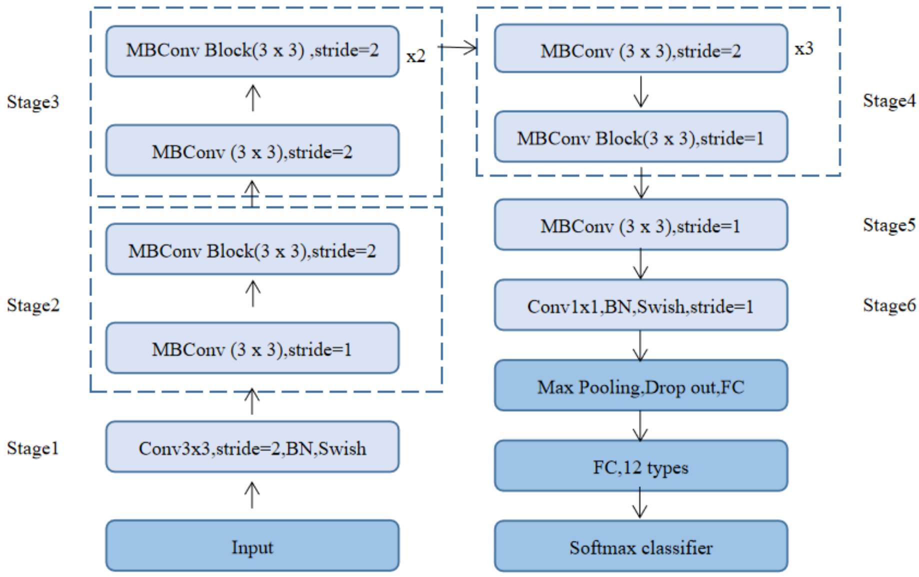


Fig. 8. Network structure of the Edge-EfficientNet

4 Experimental Verification

4.1 Training Dataset and Test Dataset

The dataset with 5280 original images is expanded to 10560 by using data enhancement techniques such as translation, chroma transformation and adding a small amount of noise. The training dataset and test dataset are divided according to the ratio of 85% : 15%, and the training dataset of 8976 images and the test dataset of 1584 images are obtained.

4.2 Model Configuration and Experimental Setup

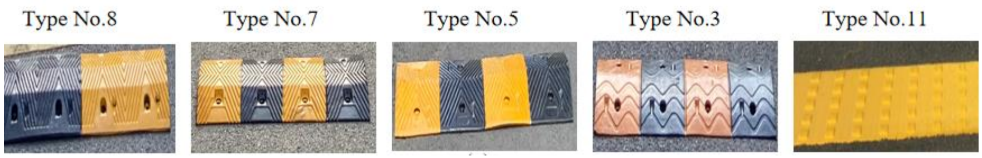
The training dataset is preprocessed such as label smoothing and normalization, so as to improve the learning ability of the model, reduce over fitting and improve the final classification accuracy. In Table 3, the configuration of the model is shown. In terms of learning rate, the model adopts warm up learning rate and CosineAnnealing method [27]. The loss function uses “cross entropy”. The padding value is set to “same”. After training, it is found that epoch of 250 and batch of 64 are better settings.

Table 3. List of hyper-parameters used in Edge- EfficientNet model

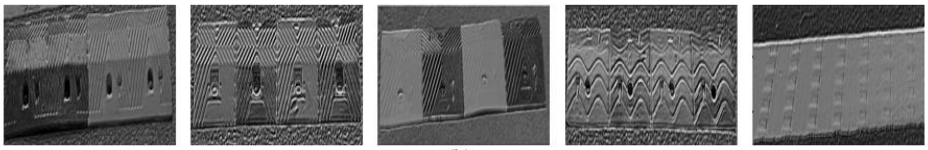
Parameters	Value
Learning rate	Dynamic learning rate
Loss-function	Cross-entropy
Epoch	250
Batch size	64
Padding	Same

4.3 Results and Analysis

On the dataset constructed in this paper, the accuracy of EfficientNetB0 model is 95.33%, and the accuracy of Edge-EfficientNet model is 98.67%. The edge detection module extracts the key features of the speed bump image. Fig. 9 shows the renderings of some examples.



(a) Original image



(b) Graphics after edge extraction

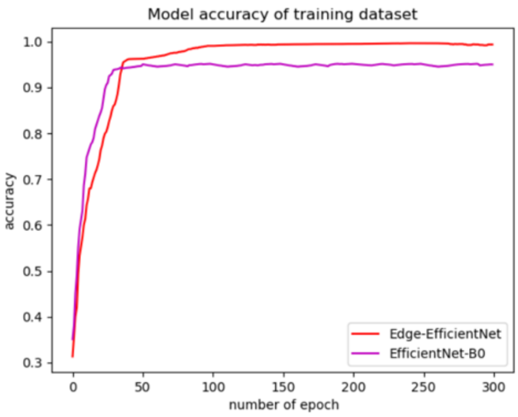
Fig. 9. Effect comparison after edge detection

The classification of speed bump is mainly based on contour, shape, texture and other features. These features are shallow features. Deep networks may cause the disappearance of back-propagation gradient and over fitting in the process of multiple extraction. Based on the improved EfficientNet basic block, we tried a variety of combinations, including different structures, convolution kernel sizes and depths, and finally determined the model of six stages through many experiments. In Table 4, the size of the trained EfficientNetB0 model is 22.1M, and the time taken to identify a picture is 64 ms. The size of the Edge-Efficientnet model is 10.3M, and the time taken to identify a picture is 26 ms, which meets the requirements of real-time.

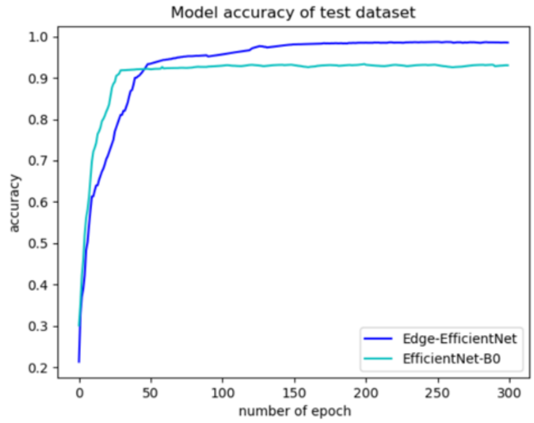
Table 4. Performance comparison between Edge-EfficientNet model and EfficientNetB0 model

CNN model	Size	Operation time	Value
EfficientNetB0	22.1M	64ms	95.33%
Edge- EfficientNet	10.3M	26ms	98.67%

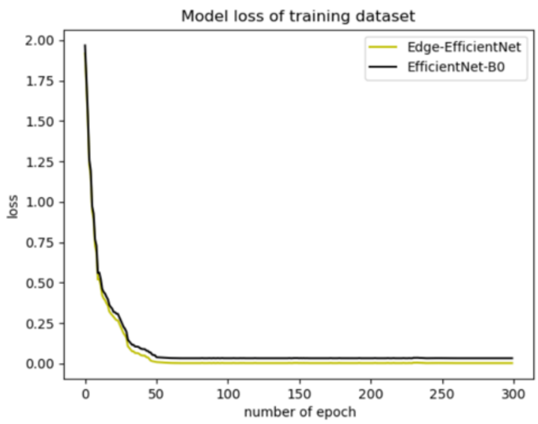
In the actual environment, the vision sensor imaging of autonomous vehicles may have some problems, such as noise, low color contrast, poor lighting conditions and so on. The image smoothing function of the edge detection module can reduce the noise interference, and the sharpening function of the module can strengthen the shape characteristics of the speed bump. Therefore, the Edge-EfficientNet model has better robustness. Fig. 10 and Table 5 show the accuracy and loss corresponding to the number of different batches used by the model, which is helpful to observe the critical point of the model training process and set the best network model parameters.



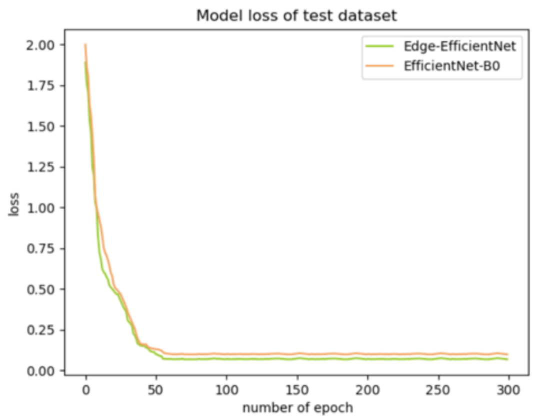
(a) Model accuracy of training dataset



(b) Model accuracy of test dataset



(c) Model loss of training dataset



(d) Model loss of test dataset

Fig. 10. Accuracy and loss of Edge-EfficientNet and EfficientNetB0 in different epoch values and phases

Table 5. Accuracy and loss of Edge-EfficientNet and EfficientNetB0 in different epoch values

Number of epoch	Edge-EfficientNet training accuracy	EfficientNetB0 training accuracy	Edge-EfficientNet test accuracy	Edge-EfficientB0 test accuracy
50	96.27%	94.70%	93.53%	94.10%
100	99.10%	97.09%	95.60%	94.89%
150	99.37%	97.11%	98.07%	94.10%
200	99.51%	97.17%	98.45%	95.33%
250	99.67%	97.10%	98.67%	95.13%
300	99.62%	97.13%	98.61%	95.24%

As is shown in Fig. 11, among the 12 types of speed bump, the recognition rates of type 7 (higher trapezoid) and type 8 (multilateral slope shape) are 96% and 91% respectively, and the recognition accuracy of other types is 100%. The main factors causing recognition errors are the speed bump height and recognition distance. When the recognition distance exceeds 7 meters, the shape information in the vertical direction of the speed bump image is incomplete. However, in the automatic driving scene, it is usually necessary to complete the target recognition as far as possible in order to provide more time for vehicle speed adjustment, which is contradictory. Therefore, if manufacturers can design unique surface texture, pattern or other obvious features, it will make the classification of speed bump more efficient and accurate.

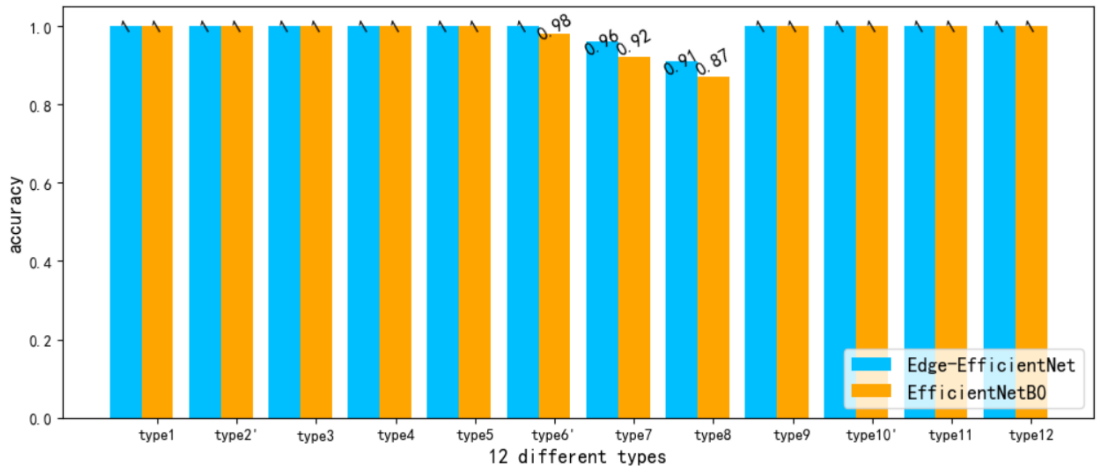


Fig. 11. The recognition accuracy of 12 types of speed bumps with Edge-EfficientNet and EfficientNetB0

5 Conclusion

In this paper, we have completed two tasks:

Firstly, the technical requirements of speed bump image acquisition in automatic driving scene are proposed, and the speed bump image dataset is constructed.

Secondly, an efficient speed bump classification network based on convolutional neural network is designed and verified by experiments.

At the front end of the feature extraction network, we improve the performance of key features such as shape, texture and contour of the image through the edge detection module. We build a network model with appropriate structure and depth based on improved EfficientNet basic block. Compared with EfficientNetB0, our model has higher accuracy, lighter model and shorter operation time. The model can be applied to the fine classification task of speed bump in automatic driving scene, which helps the vehicle to make comfort speed adjustment in time, which will promote the development of automatic driving from safety to higher comfort. At present, there are only 12 types of speed bump in the dataset. In future, we will continue to collect more samples of speed bump types, and build a standardized feature library based on Edge-EfficientNet model, which can provide data support for automatic driving.

Acknowledgments

The author is very grateful to the anonymous reviewer for his/her useful comments and suggestions, and the Fundamental Research Funds for the Program for Innovation Research Groups at Institutions of Higher Education in Chongqing (CXQT21032), and the Fundamental Research Funds for the Science and Technology Research Project of Chongqing Municipal Education Commission, China (KJQN201903402), and the Fundamental Research Funds for the Natural Science Foundation of Chongqing, China (cstc2021ycjh-bgzxm0088) for support.

References

[1] A. Khademi, S.A. Mirzapour, S.S. Hosseini, N.M. Yusof, The Best Location for Speed Bump Installation Using Taguchi and Classical Design of Experiments, Quality Engineering 26(4)(2014) 392-403.
[2] J. Li, W. Guo, L. Wang, S. Chen, Multi-objective optimization of ambulance ride comfort under speed bump, IEEJ Transactions on Electrical and Electronic Engineering 14(9)(2019) 1372-1380.
[3] P. Kokowski, R. Makarewicz, Predicted effects of a speed bump on light vehicle noise, Applied Acoustics 67(6)(2006) 570-579.
[4] F. Lu, Y. Ishikawa, H. Kitazawa, T. Satake, Effect of vehicle speed on shock and vibration levels in truck transport, Packaging Technology and Science: An International Journal 23(2)(2010) 101-109.
[5] A. Gedik, E. Bilgin, A.H. Lav, R. Artan, An investigation into the effect of parabolic speed hump profiles on ride comfort and driving safety under variable vehicle speeds: A campus experience, Sustainable cities and society 45(2019) 413-421.
[6] Z. Yang, S. Liang, Y.S. Sun, Q. Zhu, Vibration suppression of four degree-of-freedom nonlinear vehicle suspension model

excited by the consecutive speed humps, Journal of Vibration and Control 22(6)(2016) 1560-1567.
[7] S. Zhou, Y. Li, Z. Ren, G. Song, B. Wen, Nonlinear dynamic analysis of a unilateral vibration vehicle system with structural nonlinearity under harmonic excitation, Mechanical Systems and Signal Processing 116(2019) 751-771.
[8] N.I. Zainuddin, M.A. Adnan, J.M. Diah, Optimization of speed hump geometric design: Case study on residential streets in Malaysia, Journal of transportation engineering 140(3)(2014) 05013002.
[9] E. Bilgin, A.H. Lav, A. Gedik, An approach for the determination of optimal speed hump/table profiles by field tests and simulation stage, Sustainable cities and society 51(2019) 101716.
[10]M. Ososinski, F. Labrosse, Automatic Driving on Ill-defined Roads: An Adaptive, Shape-constrained, Color-based Method, Journal of Field Robotics 32(4)(2015) 504-533.
[11]S.E. Gonzalez-Reyna, J.G. Avina-Cervantes, S.E. Ledesma-Orozco, I. Cruz-Aceves, Eigen-gradients for traffic sign recognition, Mathematical Problems in Engineering, 2013(2013) 364305.
[12]X. Luo, J. Zhu, Q. Yu, Efficient convNets for fast traffic sign recognition, IET Intelligent Transport Systems 13(6)(2019) 1011-1015.
[13]W.A. Haque, S. Arefin, A.S.M. Shihavuddin, M.A. Hasan, DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements, Expert Systems with Applications 168(2021) 114481.
[14]A. Gudigar, S. Chokkadi, U. Raghavendra, U.R. Acharya, An efficient traffic sign recognition based on graph embedding features, Neural Computing and Applications 31(2)(2019) 395-407.
[15]K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. <https://arxiv.org/abs/1409.1556>, 2014 (accessed 10.04.15).
[16]C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proc. of the IEEE conference on computer vision and pattern recognition, 2015.
[17]K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. 2016 IEEE conference on computer vision and pattern recognition, 2016.
[18]M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: Proc. 2019 International conference on machine learning, 2019.
[19]A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications. <https://arxiv.org/abs/1704.04861>, 2017 (accessed 17.04.17).
[20]X. Zhang, X. Zhou, M. Lin, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proc. of the IEEE conference on computer vision and pattern recognition, 2018.
[21]F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. <https://arxiv.org/abs/1602.07360>, 2016 (accessed 04.11.16).
[22]F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proc. of the IEEE conference on computer vision and pattern recognition, 2017.
[23]V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, M. Grundmann, Blazeface: Sub-millisecond neural face detection on mobile gpus. <https://arxiv.org/abs/1907.05047>, 2019 (accessed 14.07.19).
[24]K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, Ghostnet: More features from cheap operations, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
[25]Y. Hou, J. Chen, S. Wen, The effect of the dataset on evaluating urban traffic prediction, Alexandria Engineering Journal 60(1)(2021) 597-613.
[26]C.G. Serna, Y. Ruichek, Classification of traffic signs: The european dataset, IEEE Access 6(2018) 78136-78148.
[27]I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts. <https://arxiv.org/abs/1608.03983>, 2016 (accessed 03.05.17).