Edge Intelligence Based Garbage Classification Detection Method

Ruijia Zhu[1], Yiwen Liu[1,2,3], Yanxia Gao[1,2,3](✉), Yuanquan Shi[1,2,3], and Xiaoning Peng[1,2,3]

[1] School of Computer and Artificial Intelligence, Huaihua University, Huaihua 418000, China
2877464155@qq.com

[2] Key Laboratory of Wuling-Mountain Health Big Data Intelligent Processing and Application in Hunan Province Universities, Huaihua 418000, China

[3] Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province, Huaihua 418000, China

Abstract. To address the problem that the classification and cleaning of garbage in city streets is always ineffective nowadays, the paper proposes a garbage detection method based on edge intelligence. The edge intelligence not only reduces the computational load of the cloud and speeds up the data transmission, but also greatly reduces the data transmission cost. First, images of city streets are collected and uploaded to the edge device via mobile devices in various locations in the city. Then, the edge server is used to temporarily store the image information, and the PeleeNet model deployed on it is used to identify and classify various kinds of garbage, and then visualize the information of each street. Finally, the street garbage information is transmitted to the cloud, which provides a detailed picture of the city's garbage situation and facilitates city management.

In this paper, the PeleeNet model is compared with ResNet, DenseNet and MobileNet models. The results show that the edge devices equipped with PeleeNet model not only have the fastest computation speed and the highest accuracy, but also occupy the least memory. It is fully demonstrated that the method studied in the paper can be applied to the problem of litter detection in urban streets.

Keywords: Edge Intelligence · Waste Classification · Convolutional Neural Network · Smart City

1 Introduction

With the further development of smart cities, the issue of urban street trash cleaning has once again come into focus. Traditionally, street cleaning has required manual intervention at various levels [1], and in many cities, garbage is taken by city citizens to designated garbage collection points and then collected by sanitation workers. There are also cities that install cameras at street intersections to observe the presence of trash within this area. However, these are not able to keep the city streets clean for a long time, and they do not know the status of the city's garbage in real time. Therefore, some

Z. Xiao et al. (Eds.): ICECI 2022, LNICST 478, pp. 128–141, 2023.
https://doi.org/10.1007/978-3-031-28990-3_10

scholars have studied an automated system based on edge computing [2], which collects information from edge devices, stores and simply processes the street image information using edge servers, transfers the data to a cloud center and then classifies and counts the garbage using a neural network model in deep learning. However, with the rapid development of 5G networks, network-dependent and inflexible cloud computing is no longer able to efficiently process massive amounts of data, and deploying neural network models to cloud centers for data processing can no longer yield analysis results in a very short period of time.

Based on this, this paper proposes a garbage classification detection method based on edge intelligence. It not only reduces the pressure of data transmission between edge devices and cloud center, improves the accuracy and intelligence of garbage classification, reduces the time delay of data transmission, but also protects the privacy and security of users. Images of city streets are first collected through cameras and portable mobile devices installed at street intersections. The data is then stored on an edge server, and the PeleeNet model on which it is based is used to identify and visualize the garbage images. Finally, the data is transmitted to the cloud center through the city network, and finally the garbage visualization information of the whole city is obtained. It is convenient for city managers to manage the city, arrange cleaning staff and coordinate waste management.

The rest of the paper is divided into the following parts: Sect. 2 contains a review of related work. In Sect. 2.3, the PeleeNet model is described specifically. In Sect. 3, the experimental approach is described, including the model, the image database, and the specific methods. In Sect. 4, the results of the experiments are presented. The study is concluded in Sect. 5.

2 Related Work

2.1 Edge Intelligence Research

Edge Intelligence (EI) [3] refers to the combination of endpoint intelligence, EC and AI. This new intelligence paradigm is also known as mobile intelligence [4]. Zhang et al. define edge intelligence as the ability to enable edge devices to execute AI algorithms [5]. Ken Li et al. define EI as an open platform that incorporates the core capabilities of networking, computing, storage, and applications [6]. Compared with traditional cloud-based smart end devices that upload generated data to cloud centers, edge intelligence processes and analyzes data locally, which can effectively protect user privacy, reduce response time, and save bandwidth resources [7]. Deploying intelligence on edge devices can provide intelligent services to users faster and better.

To solve the computational speed problem and resource problem in edge devices and edge networks, Hu [8] et al. proposed an algorithm that can maximize computational efficiency and optimize quality allocation. Zeng et al. proposed a resource allocation algorithm that eliminates cross-layer interference and reduces latency [9], and also proposed a solution on how to maximize the total rate at optimal power [10]. Jiang et al. proposed a solution to the task offloading and resource allocation problem in mobile edge computing [11], and to achieve the above results with limited resources, they also proposed a framework that can improve the energy efficiency of the network [12] and a method to guarantee the accuracy [13]. Liu et al. also proposed a covalent organic framework that can significantly improve the energy efficiency [14] and practical network protocols that can achieve concurrency and low power consumption [15], and also give very good approaches for transmission stability [16] and how to optimize the data transmission throughput in the network [17].

For the problem of poor signal in some areas, Zhu et al. proposed how to get accurate information under noise interference [18]. Hu et al. proposed a method to use radar signals for data enhancement [19]. In order to optimize the user experience and reduce the energy consumption at the user side, Qian [20] et al. presented the results of their study. Finally, in order to count various information, Talal [21] et al. gave very excellent statistical methods.

Finally, there are many applications that apply the edge intelligence paradigm to real life, proving the feasibility of edge intelligence. It is applied in various aspects such as industrial technology [22, 23], precision agriculture [24, 25], smart healthcare [26, 27], and smart home [28, 29].

2.2 Smart City Research

The term "smart city" has attracted worldwide attention since its emergence, and there is a boom in the construction of urban information technology with it as the core in China. However, there has been no research on urban street garbage in the construction of smart cities, which is an important reason for us to study garbage classification based on edge intelligence.

The British Standards Institute describes smart cities as "the effective integration of physical, digital and human systems in the built environment to provide a sustainable, prosperous and inclusive future for citizens" [30]. In China, there are already successful examples of smart cities as well. Alibaba's cloud computing project "City Brain" used data collected from video feeds of traffic signals to alleviate traffic congestion in Hangzhou, China, where traffic management was 92% accurate in identifying traffic violations. It helped emergency vehicles reach their destinations 50% faster than before and increased traffic speeds by 15% [31]. City government leaders and planners can also use the city brain to overcome other pressing problems, such as alleviating the problem of diminishing water supplies.

2.3 Convolutional Neural Network Model

With the continuous development of deep learning, garbage classification has gradually become intelligent, and the use of deep learning to classify garbage has become a key research direction in academia and industry. As the authoritative network structure in deep learning, convolutional neural network model has made great achievements in the field of image processing, so this paper uses convolutional neural network for urban street garbage classification processing.

The first convolutional neural network was proposed by Wei Zhang and was successfully applied in the field of medical image detection [32]. Subsequently, Yann LeCun proposed LeNet, a model that has made remarkable contributions in the field of computer vision [33]. Based on this, Yann et al. successfully solved the problem of handwritten digit recognition in 1998 [34]. Since then the research area of convolutional neural networks has become the focus. Medical image classification [35], handwritten digital image [36], coronary virus detection [37] are all successful applications of convolutional neural networks. The following is a brief description of each neural network model that will be used in this paper.

ResNet

With the exploration of the majority of researchers in the field of convolutional neural networks, we find that improving the performance of the network can increase the depth of the network. However, many experiments have proved that simply increasing network depth in a certain depth range cannot effectively improve network performance. If the number of neural network layers is within 20, increasing the number of network layers will bring improvement of network performance, and the category accuracy will decrease. This is because the deepening of the network will cause the problem of gradient explosion and gradient disappearance. In order to make a deeper network training good results, in 2016, HE [38] and others proposed a 34-layer residual network. The problem. However, deep neural networks still face the dilemma of gradient disappearance and degeneration, and network training has become a difficult problem.

DenseNet

To address this problem, the DenseNet model [39] was born. The two main advantages of the DenseNet model are the implementation of dense connectivity and feature reuse. These advantages allow DenseNet to achieve better performance with few parameters and lower cost. DenseNet also has very good generalization performance, which is especially suitable for training small data set applications. At this time, the performance of the network has been greatly improved. However, new issues arise, one is whether the device can successfully store our model, and the other is the time consumption of the model for computation. Therefore, in order to meet the practical application criteria, the only options are to optimize the processor performance or to reduce the amount of computation. Due to the time problem, reducing the amount of computation becomes the main technical tool. Solving the above problems, neural network models can be widely used in mobile.

MobileNet

Therefore, Google proposed MobileNet, a lightweight neural network model, in 2017 [40]. It is mainly deployed on resource-limited mobile and embedded devices. MobileNet requires fewer parameters, the whole model is very lightweight and computationally fast, easily meeting the requirements of these portable devices. But it also has a fatal problem of being very dependent on deep separable convolutions, which prevents it from being deployed in most frameworks.

PeleeNet

Therefore, researchers from the University of Western Ontario, Canada, have proposed PeleeNet, which requires only ordinary convolution for real-time operation on mobile devices [41]. PeleeNet has the same excellent performance as MobileNet and is applicable to almost all frameworks. It has a very good performance in terms of accuracy, speed, and power consumption. We will also introduce the PeleeNet model in detail in later sections.

In summary, the PeleeNet model is chosen as the training network in this paper.

3 Experimental Model

The PeleeNet model borrows the cascade model and architecture from the DenseNet model. On its basis, it solves the problem of limited both storage capacity and computational power. The following are the main improvement features of PeleeNet and the general structure of the PeleeNet model.

3.1 Stem Block

The Stem Block structure is the method used for downsampling in PeleeNet. This module ensures a good feature representation of the model and reduces a large number of parameters (Fig. 1).

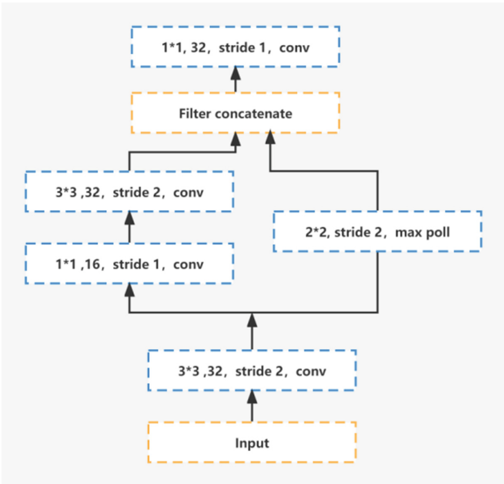

Fig. 1. The convolution operation is performed by Stem Block to enrich the feature layers and reduce the number of weight parameters.

The structure first performs a 3×3 convolution operation on the input image, the main purpose of which is to change the number of channels of the feature map. Then the network structure is divided into two branches, and the feature map is divided into two parts. One part of the feature map is pooled for maximum value, and the other part of the feature map is convolved by a 1×1 operation to halve the number of channels, followed by a 3×3 convolution with a step size of 2 to achieve the second downsampling. The outputs of the two branches are stitched together in the channel dimension, and finally the number of channels is reduced by another 1×1 convolution. Compared with the original convolution operation, the main operation of Stemblock structure to reduce the number of parameters is to introduce a bottleneck layer in one branch to reduce the number of channels before downsampling, and the other branch to pool the original input to the maximum value before stitching, in order to pass some of the information in the input and ensure that the final result still has enough semantic information based on the reduced number of parameters, without excessive loss of information. The purpose is to pass some of the information in the input to ensure that the final result still has enough semantic information on the basis of the reduced number of parameters, without excessive loss of information. In general, the Stem Block not only enriches the feature layer, but also greatly reduces the number of weight parameters.

3.2 Two-Way Dense Layer

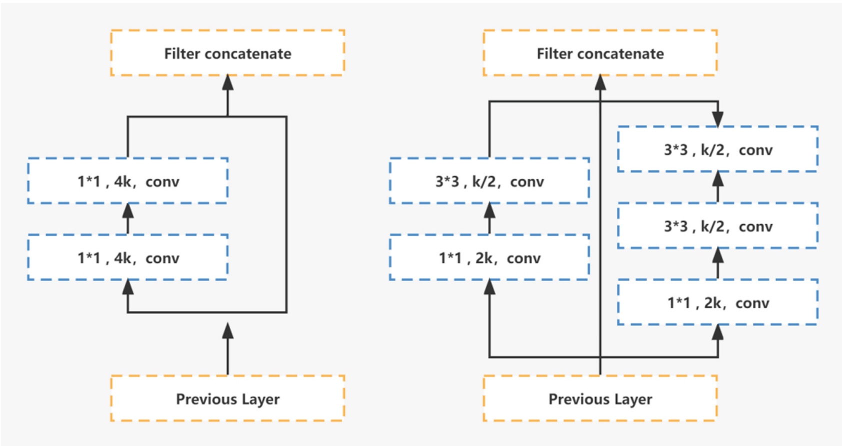

Fig. 2. The filter is convolved in two ways by Two-Way Dense Layer to achieve the effect of taking into account the size of the target.

The left (a) diagram above shows the basic module designed in DenseNet, where k and 4k represent the number of filters. The right (b) figure represents the basic module designed in PeleeNet, in addition to halving the filter of the original backbone branch (the perceptual field of the backbone branch is 3×3), a new branch is added, in which two 3×3 convolutions are used, and the perceptual field of this branch is 5×5. This ensures that the extracted features are not just single-scale, but can take into account both Small and large targets.

3.3 Transition Layer Without Compression

In DenseNet, the transition layer is used to reduce the spatial resolution of the feature map, and the number of channels in the transition layer will be smaller than the number of channels in the previous layer. However, experiments show that the compression factor proposed by DenseNet can damage the feature representation. And setting the number of channels in the transition layer and the previous layer to the same value in PeleeNet can solve this problem.

3.4 Dynamic Number of Channels of Bottleneck Layer

In DenseNet, the number of channels in the bottleneck layer of each Dense Block is fixed, but this causes the number of channels in the bottleneck in some layers at the beginning of the network to far exceed the number of input channels, resulting in an increase in computation. In PeleeNet, the dynamic setting of the number of channels in the bottleneck layer solves this problem.

3.5 Composite Function

Also, to increase the speed, the PeleeNet model uses the combination of conv+bn+relu (instead of the pre-activation combination (conv+relu+bn) in DenseNet). In post-activation, all BN layers can be merged with convolutional layers in the inference phase, which can facilitate model inference speedup. Finally, to compensate for the negative impact of this change on accuracy, PeleeNet uses a shallower and wider network structure, i.e., a 1×1 convolutional layer is added after the last sense block, as a way to obtain stronger feature representation. The structure is calculated as follows:

$$\begin{aligned} Y &= \frac{X - AVE_{BN}}{\sqrt{VAR_{BN} + \varepsilon}} * SCALE + W \\ &= \frac{X * SCALE}{\sqrt{VAR_{BN} + \epsilon}} + W - \frac{SCALE * VAR_{BN}}{\sqrt{VAR_{BN} + \varepsilon}} \end{aligned} \tag{1}$$

$$Z_{BN} = \frac{SCALE}{\sqrt{VAR_{BN} + \varepsilon}} * A_{BN} = W - \frac{SCALE * VAR_{BN}}{\sqrt{VAR_{BN} + \varepsilon}} \tag{2}$$

Substituting Eq. (2) into Eq. (1) yields:

$$Y = Z_{BN} * X + A_{BN} \tag{3}$$

$$X = Z_{Conv} * X_{conv} + A_{Conv} \tag{4}$$

Combining Eq. (3) with (2) yields.

$$Y = Z_{BN} * (Z_{Conv} * X_{Conv} + A_{Conv}) + A_{BN} \tag{5}$$

The final results are as follows:

$$Weighting : Z = Z_{BN} * Z_{Conv}$$

$$Offset : A = Z_{BN} * Z_{Conv} + A_{BN}$$

The input X is normalized in the BN layer. $\mathrm{AVE_{BN}}$ is the mean of the input, $\mathrm{VAR_{BN}}$ is the variance of the input, SCALE is the scaling, W is the displacement, the convolution kernel size is K, the weight is $\mathrm{Z_{Conv}}$, the bias is $\mathrm{A_{Conv}}$. And the number of channels is $\mathrm{B_{Conv}}$.

3.6 General Structure of the PeleeNet

(See Fig. 3).

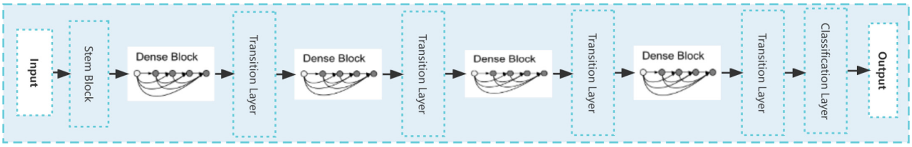

Fig. 3. The structure of each level of the PeleeNet model

4 Process Design

4.1 Method Overview

The architecture of the method model consists of three parts: the first part is the data collection from the edge devices, the second part is the analysis and processing of the data by the edge server, and the third part is the cloud server presenting the whole city garbage information status.

When the garbage images taken by edge devices are transmitted to the edge server, the edge server uses the PeleeNet model deployed on it to analyze and process the image data temporarily stored in the server, delete the images with garbage and classify and count the garbage in them in order to present the city's street garbage data quickly and accurately in the cloud server center. In the local management, the edge data center not only stores and uploads the garbage classification results accurately processed by the PeleeNet model, but also stores the initial data from the edge servers locally for a long time, which not only prevents data loss due to the failure of the edge devices, but also protects user privacy. City administrators can allocate human resources according to the visualization results of garbage and realize intelligent garbage disposal. Figure 2 shows the process of building a garbage detection model for city streets (Fig. 4).

4.2 Data Collection

The main source of city street spam information images is from edge devices such as cameras, sensors, smartphones, etc. spread all over the city, as well as information from local management centers.

For the information uploaded by edge devices, the edge server needs to set certain criteria: 1) fixed image resolution; 2) fixed number of photos taken by each edge device; 3) fixed shooting area.

For the local data center, it needs to transmit the garbage classification data to the city administrator at a fixed time, while the city administrator makes timely responses through the reports from the data center and arranges the cleaning staff to clean it in time.

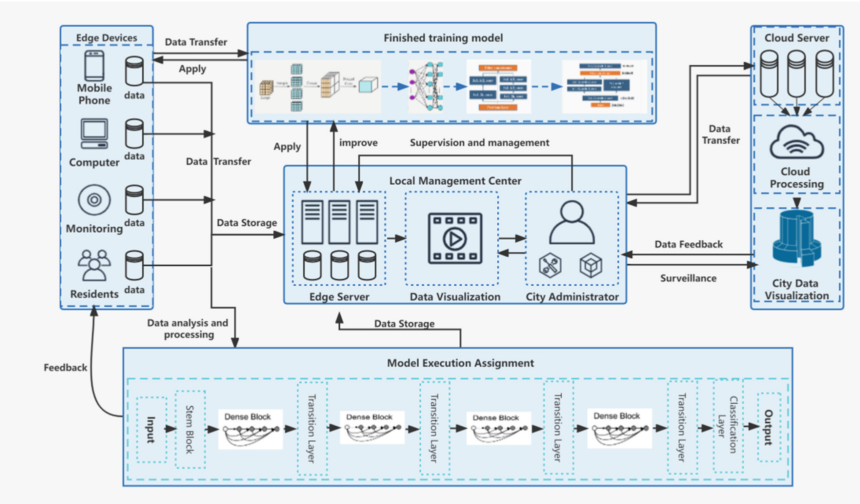

Fig. 4. Overall framework of edge intelligence-based waste sorting monitoring model

4.3 Mobile Edge Processing

Sometimes, the information uploaded by the edge devices may be useless information. For example, in one of the pictures collected by an edge device, there are obstacles such as cars and houses blocking causing incomplete information of garbage in the captured garbage images, or garbage does not appear in the captured photos. These photos are obviously useless, and the edge data processing layer is essential in order to reduce the workload and time consumption of the edge server. This layer accepts the information of garbage images from the edge devices and filters out the complete photos containing garbage images before analyzing and processing them. Figure 5 shows the basic architecture of mobile edge computing, which mainly consists of the following parts.

Edge devices: Every camera and sensor in the city streets are edge devices that can collect information, while every resident is able to take pictures of the streets with his or her own device, and this data is transmitted to the edge server.

Edge server: It is an edge device that can provide a data transmission channel and establishes a reliable connection with nearby mobile devices. Its role is to process service requests from mobile devices and store the information uploaded by mobile devices, and analyze and process the data through the neural network model mounted on it.

Cloud Center: Processes service requests from the edge servers and stores the information uploaded by the servers, and visualizes the final city street waste information.

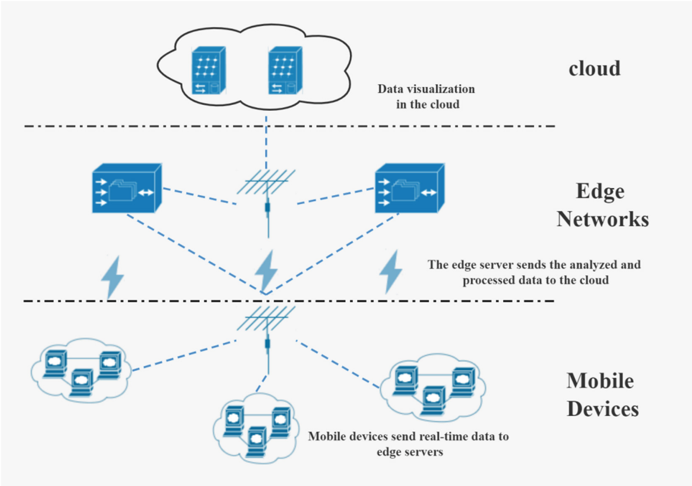

Fig. 5. Foundational Framework for Edge Computing

4.4 Image Detection Based on Pelee Model

The experiments are based on Window10 environment and simulated using iFogSim, a virtual simulation platform for edge computing.

Data Preparation

In this training, we collected more than 20,000 garbage images. These images were divided into four major categories, namely food waste, recyclable waste, hazardous waste, and other waste. In total, there are forty sub-categories, including scraps of left-overs, newspapers, shopping paper bags, foam boxes, batteries, pharmaceuticals, etc. A total of 15,000 images were used as the training set and 5,000 images were used as the test set with a ratio of (7:3).

Model Construction and Training

First, a convolutional layer is built to perform feature extraction and then normalization. Next, an activation function is chosen to introduce nonlinearities to enable the model to solve nonlinear problems. Then the first Stem Block module is built to preserve the feature representation and reduce the parameters. Then the most critical Two-Way Dense Layer is built, which makes the model very flexible and can detect both large and small shapes. Then the Two-Way Dense Layer is repeated to form the Dense Block, and each Dense Block is directly built with a transition layer for dimensionality reduction. After repeating these operations, we finally add a classification layer to complete the model construction.

Evaluation Model

In this paper, we use the following performance metrics to measure the performance of a model: Computational Cost, Speed, Accuracy, Memory Usage, Computational Complexity.

5 Conclusion

Table 1. Various performance comparisons of PeleeNet model with other models

Model	Computational Cost	Speed (320 × 320)	Accuracy	Memory Usage	Computational Complexity
ResNet-101	7864M	118.1M	77.5%	818M	50M
ResNet-152	11832M	70M	78.1%	906M	53M
DenseNet-161	4145M	60.3M	77.6%	774M	36M
DenseNet-201	3680M	58.8M	77.1%	724M	32.3M
MobileNet-v1	569M	75.7M	70.1%	648M	5.3M
MobileNet-v2	321M	68.9M	71.8%	635M	5.2M
PeleeNet	507M	129.2M	74.3%	362M	2.8M

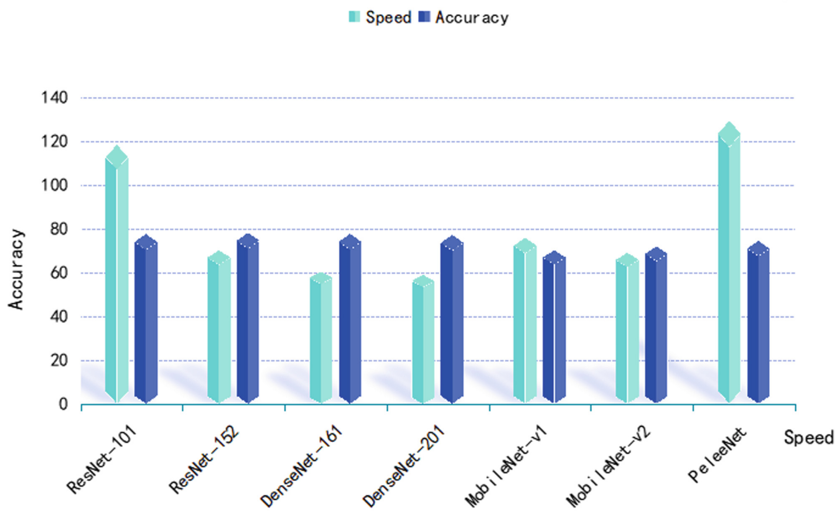

Fig. 6. Comparison of speed and accuracy of each model on edge devices

Putting each model on the virtual simulation platform iFogSim (Table 1 and Figs. 6, 7).

From the above tables and charts, we can clearly observe that PeleeNet, with its lightweight architecture, not only occupies little memory and has low computational complexity, but also has good accuracy and computational speed. It is well proven that it is suitable for deployment on edge devices.

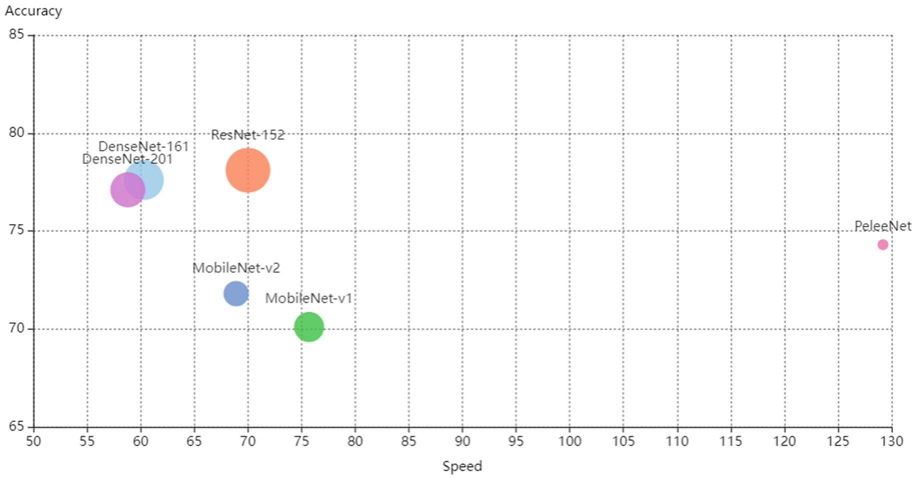

Fig. 7. Comparison of computational complexity with accuracy and speed of different models

Acknowledgment. This work was supported in part by the Scientific research projects funded by the Department of education of Hunan Province (No. 21C0628, No. 20C1472 and No. 22C0668), the Huaihua University Double First-Class initiative Applied Characteristic Discipline of Control Science and Engineering (No. ZNKZN2021-10) ,the National Natural Science Foundation of China (No. 62172182), the Hunan Provincial Natural Science Foundation of China (No. 2020JJ4490), the Project of Hunan Provincial Social Science Foundation (NO.21JD046), the Huaihua University Project (No. HHUY2019-25), the Philosophy and Social Science Achievement Evaluation Committee of Huaihua (No. HSP2022YB40) and the Science and Technology Innovation 2030 Special Project Sub-Topics (No. 2018AAA0102100).

References

1. Balchandani, C., Hatwar, K., Makkar, P., et al.: A deep learning framework for smart street cleaning. In: IEEE Third International Conference on Big Data Computing Service and Applications, pp. 112–117. IEEE, Redwood City (2017)
2. Begur, H., Dhawade, M., Gaur, N., et al.: An edge-based smart mobile service system for illegal dumping detection and monitoring in San Jose. In: IEEE International Conference on Smart City and Innovation, pp. 1–6. IEEE, San Francisco (2017)
3. Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., Chen, M.: In-edge AI: intelligentizing mobile edge computing, caching and communication by federated learning. IEEE Netw. **33**(5), 156–165 (2019)
4. Wang, Z., Cui, Y., Lai, Z.: A first look at mobile intelligence: architecture, experimentation and challenges. IEEE Netw. **33**, 120–125 (2019)
5. Zhang, X., Wang, Y., Lu, S., et al.: Open EI: an open framework for edge intelligence. In: IEEE 39th International Conference on Distributed Computing Systems, pp. 1840–1851 (2019)
6. Li, K.L., Liu, C.B.: Edge intelligence: state-of-the-art and expectations. Big Data Res. 69–75 (2019)
7. Khelifi, H., et al.: Bringing deep learning at the edge of information-centric internet of things. IEEE Commun. Lett. **23**(1), 52–55 (2018)

8. Hu, Z., Zeng, F., Zhu, X., Fu, B., Jiang, H., Chen, H.: Computation efficiency maximization and QoE-provisioning in UAV-enabled MEC communication systems. IEEE Trans. Netw. Sci. Eng. **8**(2), 1630–1645 (2021)
9. Zeng, F., Li, Q., Zhu, X., Havyarimana, V., Bai, J.: A price-based optimization strategy of power control and resource allocation in full-duplex heterogeneous Macrocell-Femtocell networks. IEEE Access **6**, 42004–42013 (2018)
10. Zeng, F., et al.: Resource allocation and trajectory optimization for QoE provisioning in energy-efficient UAV-enabled wireless networks. IEEE Trans. Veh. Technol. **69**(7), 7634–7647 (2020)
11. Jiang, H., Dai, X., Zhu, X., Iyengar, A.: Joint task offloading and resource allocation for energy-constrained mobile edge computing. IEEE Trans. Mob. Comput. (2022). https://doi.org/10.1109/TMC.2022.3150432
12. Jiang, H., Zhu, X., Li, Z., Xu, J., Zeng, F., Wang, D.: An energy-efficient framework for internet of things underlaying heterogeneous small cell networks. IEEE Trans. Mob. Comput. **21**(1), 31–43 (2022)
13. Jiang, H., Cao, H., Liu, D., Xiong, J., Cao, Z.: SmileAuth: using dental edge biometrics for user authentication on smartphones. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **4**(3), 84:1–84:24 (2020)
14. Liu, D., Cao, Z., He, Y., Ji, X., Hou, M., Jiang, H.: Exploiting concurrency for opportunistic forwarding in duty-cycled IoT networks. ACM Trans. Sens. Networks **15**(3), 31:1–31:33 (2019)
15. Liu, D., Hou, M., Cao, Z., He, Y., Ji, X., Zheng, X.: COF: exploiting concurrency for low power opportunistic forwarding. In: ICNP, pp. 32–42 (2015)
16. Liu, D., Wu, X., Cao, Z., Liu, M., Li, Y., Hou, M.: CD-MAC: a contention detectable MAC for low duty-cycled wireless sensor networks. In: SECON, pp. 37–45 (2015)
17. Liu, D., Cao, Z., Hou, M., Rong, H., Jiang, H.: Pushing the limits of transmission concurrency for low power wireless networks. ACM Trans. Sens. Netw. **16**(4), 40:1–40:29 (2020)
18. Zhu, X., et al.: Toward accurate vehicle state estimation under non-Gaussian noises. IEEE Internet Things J. **6**(6), 10652–10664 (2019)
19. Hu, J., et al.: BlinkRadar: non-intrusive driver eye-blink detection with UWB radar. In: Proceedings of IEEE ICDCS (2022)
20. Qian, C., Liu, D., Jiang, H.: Harmonizing energy efficiency and QoE for brightness scaling-based mobile video streaming. In: IWQoS, pp. 1–10 (2022)
21. Talal, A., Zhu, X., Sun, J., Seyedali, M., Vincent, H., Jiang, H.: Optimal design of IIR wideband digital differentiators and integrators using salp swarm algorithm. Knowl.-Based Sys. **182**(15), 104834 (2019). https://doi.org/10.1016/j.knosys.2019.07.005
22. Morán, A., et al.: Hardware-optimized reservoir computing system for edge intelligence applications. Cogn. Comput. 1–9 (2021)
23. Foukalas, F., Tziouvaras, A.: Edge artificial intelligence for industrial internet of things applications: an industrial edge intelligence solution. IEEE Industr. Electron. Mag. **15**, 28–36 (2021)
24. Liu, J., et al.: Boost precision agriculture with unmanned aerial vehicle remote sensing and edge intelligence: a survey. Remote. Sens. **13**, 4387 (2021)
25. Friha, O., et al.: Internet of things for the future of smart agriculture: a comprehensive survey of emerging technologies. IEEE/CAA J. Autom. Sinica **8**, 718–752 (2021)
26. Kamruzzaman, M. M.: New opportunities, challenges, and applications of edge-AI for connected healthcare in smart cities. In: IEEE Globecom Workshops, pp. 1–6 (2021)
27. Amin, S.U., Hossain, M.S.: Edge intelligence and internet of things in healthcare: a survey. IEEE Access **9**, 45–59 (2021)
28. Yar, H., et al.: Towards smart home automation using IoT-enabled edge-computing paradigm. Sens. (Basel Switz.) **21** (2021)

29. Yuan, D., et al.: Intrusion detection for smart home security based on data augmentation with edge computing. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), pp. 1–6 (2020)
30. Smart Cities Overview—Guide, BSI Standards Publication. http://shop.bsigroup.com/upload/Shop/Download/PAS/30313208-PD8100-2015.pdf. Accessed 7 Aug 2020
31. Hasija, S., Shen, Z.J.M., Teo, C.P.: Smart city operations: modeling challenges and opportunities. Manuf. Serv. Oper. Manage. **22**, 203–213 (2020)
32. Zhang, W.: Shift-invariant pattern recognition neural network and its optical architecture. In: Proceedings of Annual Conference of the Japan Society of Applied Physics, pp. 2147–2151 (1988)
33. LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**, 541–551 (1989)
34. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**, 2278–2324 (1998)
35. Jiang, Y., Chen, L., Zhang, H., Xiao, X.: Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module. PloS One **14**(3) (2019)
36. Xu, X., et al.: 11 TOPS photonic convolutional accelerator for optical neural networks. Nature **589**, 44–51 (2021)
37. Narin, A., Kaya, C., Pamuk, Z.: Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. Pattern Anal. Appl. **24**(3), 1207–1220 (2021). https://doi.org/10.1007/s10044-021-00984-y
38. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
39. Huang, G., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
40. Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 21–26 (2017)
41. Mahmud, R., Buyya, R.: Modelling and simulation of fog and edge computing environments using iFogSim toolkit. Fog Edge Comput. Principles Paradigms 1–35 (2019)