

Research Article

A Novel Image Classification Approach via Dense-MobileNet Models

Wei Wang ¹, Yutao Li ¹, Ting Zou ², Xin Wang ¹, Jieyu You ³ and Yanhong Luo ³

¹School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China

²Yiyang Branch, China Telecom Co., Ltd., Yiyang 413000, China

³Hunan Children's Hospital, Changsha 410000, China

Correspondence should be addressed to Jieyu You; yjy660111@sina.com and Yanhong Luo; mfxgz123@163.com

Received 1 September 2019; Accepted 12 December 2019; Published 6 January 2020

Guest Editor: Malik Jahan Khan

Copyright © 2020 Wei Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a lightweight deep neural network, MobileNet has fewer parameters and higher classification accuracy. In order to further reduce the number of network parameters and improve the classification accuracy, dense blocks that are proposed in DenseNets are introduced into MobileNet. In Dense-MobileNet models, convolution layers with the same size of input feature maps in MobileNet models are taken as dense blocks, and dense connections are carried out within the dense blocks. The new network structure can make full use of the output feature maps generated by the previous convolution layers in dense blocks, so as to generate a large number of feature maps with fewer convolution cores and repeatedly use the features. By setting a small growth rate, the network further reduces the parameters and the computation cost. Two Dense-MobileNet models, Dense1-MobileNet and Dense2-MobileNet, are designed. Experiments show that Dense2-MobileNet can achieve higher recognition accuracy than MobileNet, while only with fewer parameters and computation cost.

1. Introduction

Computer image classification is to analyze and classify images into certain categories to replace human visual interpretation. It is one of the hotspots in the field of computer vision. Because the features are very important to classification, most of the researches on image classification focus on image feature extraction and classification algorithms. Traditional image features such as SIFT and HOG are designed manually. Convolutional neural networks have the ability of self-learning, self-adapting, and self-organizing; so, it can automatically extract features by using the prior knowledge of the known categories, and avoid the complicated process of feature extraction in traditional image classification methods. At the same time, the extracted features are highly expressive and efficient.

Deep convolutional neural network (CNN) has achieved significant success in the field of computer vision, such as image classification [1], target tracking [2], target detection [3], and semantic image segmentation [4, 5]. For example, in

the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012), Krizhevsky et al. won the championship with an AlexNet [1] model of about 60 million parameters and eight layers. In addition, VGG [6] with 16-layer, GoogleNet [7] with Inception as the basic structure, and ResNet [8] with residual blocks that can alleviate the problem of gradient disappearance have also achieved great success. However, the deep convolutional neural network itself is a dense computational model. The huge number of parameters, heavy computing load, and large number of memory access lead to huge power consumption, which makes it difficult to apply the model to portable mobile devices with limited hardware resources.

In order to apply the deep convolutional neural network model to real-time applications and low-memory portable devices, a feasible solution is to compress and accelerate the deep convolutional neural networks to reduce parameters, computation cost, and the power consumption. Denil et al. [9] proved that the parameters of deep convolutional neural network have a lot of redundancy, and these redundant

parameters have little influence on the classification accuracy. Denton et al. [10] found an appropriate low-rank matrix to estimate the information parameters of deep CNNs by singular value decompositions. The method requires high computational cost and more retraining to achieve convergence. Han et al. [11] deleted the unimportant connections in the pretrained network by parameter pruning, retrained and quantized the remaining parameters, and then encoded the quantized parameters by Hoffman coding to further reduce the compression rate. However, the method requires manual adjustment of superparameters. Chen et al. [12] used a low-cost Hash function to group the weights between the two adjacent layers into a Hash bucket for weight sharing, which reduces the storage of additional positions and realizes parameter sharing. Hinton et al. [13] compressed the network model by knowledge distillation, and extracted useful information. The useful information is migrated to a smaller and simpler network, which made the simple network and the complex network have similar performance.

In addition, many related researches have improved network models to compress networks. For example, SqueezeNet [14] is a network model based on fire module, MobileNets [15] is a network model based on depthwise separable filters, and ShuffleNet [16] is improved on the basis of residual structure by introducing group pointwise convolution and channel shuffle operation.

Compared with VGG-16 network, MobileNet is a lightweight network, which uses depthwise separable convolution to deepen the network, and reduce parameters and computation. At the same time, the classification accuracy of MobileNet on ImageNet data set only reduces by 1%. However, in order to be better applied to mobile devices with limited memory, the parameters and computational complexity of the MobileNet model need to be further reduced. Therefore, we use dense blocks as the basic unit in the network layer of MobileNet. By setting a small growth rate, the model has fewer parameters and lower computational cost. The new models, namely Dense-MobileNets, can also achieve high classification accuracy.

2. Fundamental Theory

2.1. MobileNet. MobileNet is a streamlined architecture that uses depthwise separable convolutions to construct lightweight deep convolutional neural networks and provides an efficient model for mobile and embedded vision applications [15]. The structure of MobileNet is based on depthwise separable filters, as shown in Figure 1.

Depthwise separable convolution filters are composed of depthwise convolution filters and point convolution filters. The depthwise convolution filter performs a single convolution on each input channel, and the point convolution filter combines the output of depthwise convolution linearly with $1 * 1$ convolutions, as shown in Figure 2.

2.2. Dense Connection. DenseNet [17] proposed a new connection mode, connecting each current layer of the

network with the previous network layers, so that the current layer can take the output feature maps of all the previous layers as input features. To some extent, this kind of connection can alleviate the problem of gradient disappearance. Since each layer is connected with all the previous layers, the previous features can be repeatedly used to generate more feature maps with less convolution kernel.

DenseNet takes dense blocks as basic unit modules, as shown in Figure 3. In Figure 3, a dense block structure consists of 4 densely connected layers with a growth rate of 4. Each layer in this structure takes the output feature maps of the previous layers as the input feature maps. Different from the residual unit in ResNet [8], which combines the sum of the feature maps of the previous layers in one layer, the dense block transfers the feature maps to all the subsequent layers, adding the dimension of the feature maps rather than adding the pixel values in the feature maps.

In Figure 4, the dense block only superimposes the feature maps of the previous convolution layers and increases the number of feature maps. Therefore, only the magnitude of x_l and x_{l+1} is required to be equal, and the number of feature maps does not need to be the same. DenseNet uses hyperparameter growth rate to control the number of feature map channels in the network. The growth rate k indicates that the output feature maps of each network layer is k . That is, for each convolution layer, the input feature maps of the next layer will increase k channels.

3. Dense-MobileNet

Dense-MobileNet introduces dense block idea into MobileNet. The convolution layers with the same size of input feature maps in MobileNet model are replaced as dense blocks, and the dense connections are carried out within the dense blocks. Dense block can make full use of the output feature maps of the previous convolution layers, generate more feature maps with fewer convolution kernels, and realize repeated use of features. By setting a small growth rate, the parameters and computations in MobileNet models are further reduced, so that the model can be better applied to mobile devices with low memory.

In this paper, we design two different Dense-MobileNet structures: Dense1-MobileNet and Dense2-MobileNet.

3.1. Dense1-MobileNet. MobileNet model is a network model using depthwise separable convolution as its basic unit. Its depthwise separable convolution has two layers: depthwise convolution and point convolution. Dense1-MobileNet model considers the depthwise convolution layer and the point convolution layer as two separate convolution layers, i.e., the input feature maps of each depthwise convolution layer in the dense block are the superposition of the output feature maps in the previous convolution layer, and so is the input feature maps of each deep convolution layer, as shown in Figure 5. Because depthwise convolution is a single channel convolution, the number of output feature maps of the middle depthwise convolution layer is the same

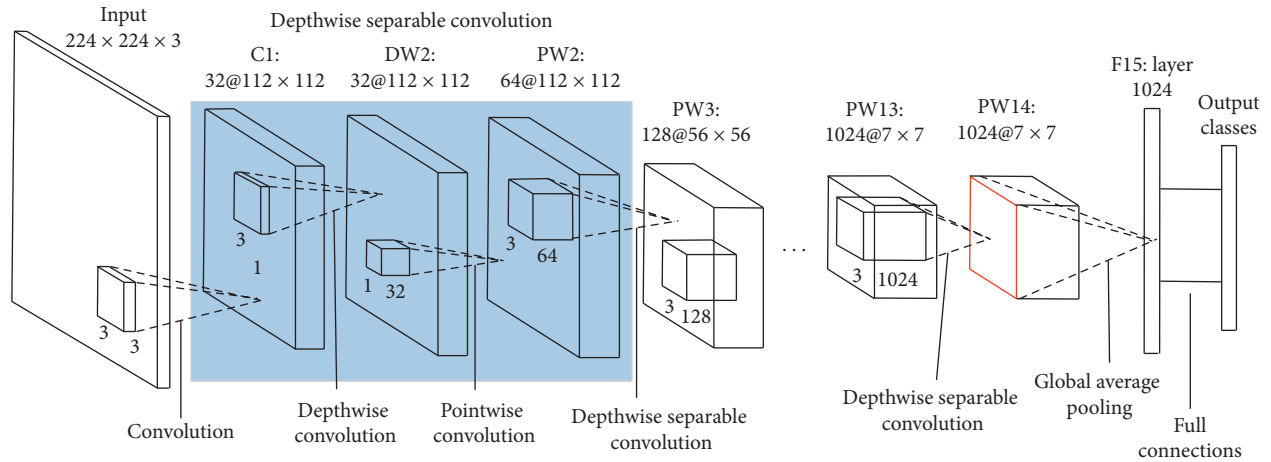


FIGURE 1: Architecture of MobileNet.

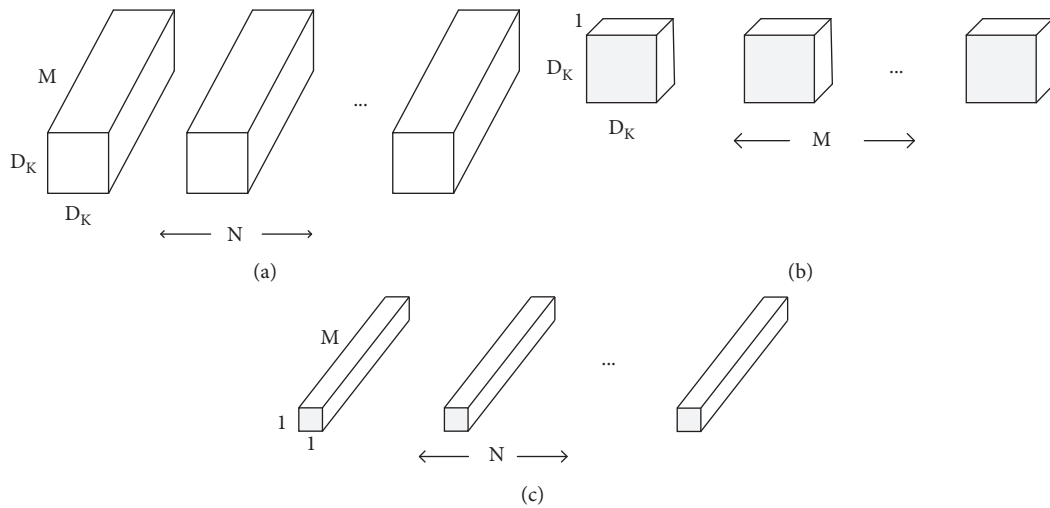


FIGURE 2: Standard convolutional filters and depthwise separable filters. (a) Standard convolutional filters, (b) depthwise convolutional filters, and (c) point convolutional filters.

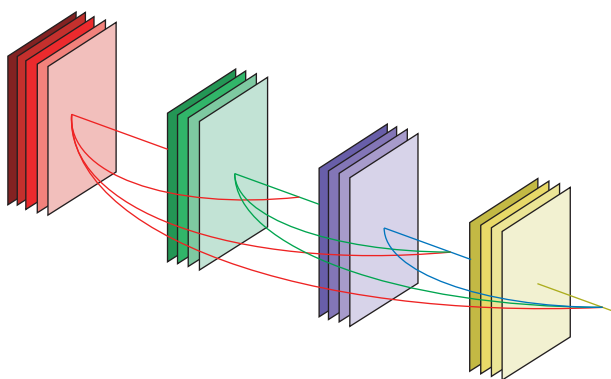


FIGURE 3: DenseNet structure.

as that of the input feature maps, which is the sum of the output feature maps of all the previous layers.

DenseNet contains a transition layer between two consecutive dense blocks. The transition layer reduces the number of input feature maps by using 1×1 convolution

kernel and halves the number of input feature maps by using 2×2 average pooling layer. The above two operations can ease the computational load of the network. Different from DenseNet, there is no transition layer between two consecutive dense blocks in Dense1-MobileNet model, the reason are as follows: (1) in MobileNet, batch normalization is carried out behind each convolution layer, and the last layer of the dense blocks is 1×1 point convolution layer, which can reduce the number of feature maps; (2) in addition, MobileNet reduces the size of feature map by using convolution layer instead of pooling layer, that is, it directly convolutes the output feature map of the previous point convolution layer with stride 2 to reduce the size of feature map.

3.2. Dense2-MobileNet. Dense2-MobileNet takes depthwise separable convolution as a whole, called a dense (depthwise separable convolution) block, which contains two point convolutional layers and a depthwise convolutional layer.

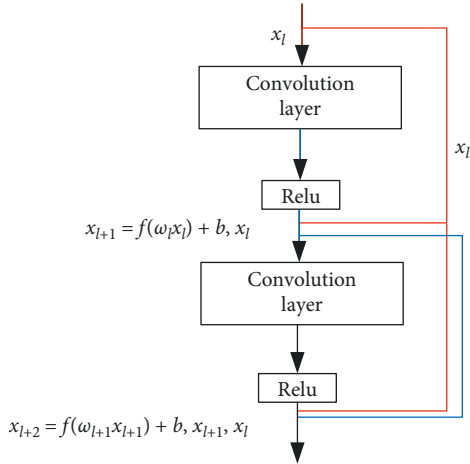


FIGURE 4: Two layers of dense connected modules.

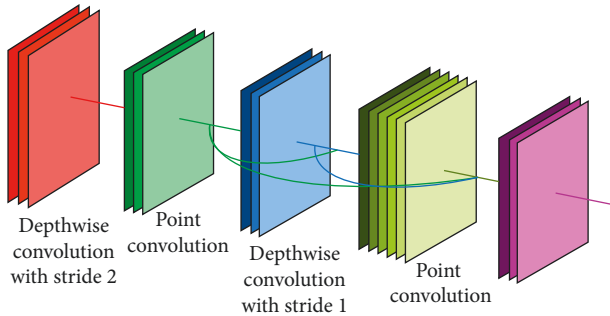


FIGURE 5: Schematic diagram of the Dense1-MobileNet model.

The input feature maps of depthwise separable convolution layer is the accumulation of output feature maps generated by point convolutions in all previous depthwise separable convolution layers, while the input feature map in point convolution layer is only the output feature map generated by the depthwise convolution in the dense block, not the superposition of the output feature maps of all the previous layers. So, the dense block structure in this model only has one dense connection, as shown in Figure 6.

In Dense2-MobileNet model, only one input feature map needs to overlay the output feature map of point convolution in the upper depthwise separable convolution layer. Because of the fewer cumulative times of structural feature maps, the number of output feature maps of all layers in a dense block is also less cumulative; so, it is not necessary to reduce the channel of feature maps by a $1 * 1$ convolution. After superimposing the output feature maps generated by the previous separable convolutions, the size of the feature map can be reduced by the depthwise convolution with stride 2; so, the Dense2-MobileNet model does not add other transition layers too. The MobileNet model is finally pooled globally and connected directly to the output layer. Experiments show that the classification accuracy of the global average prepooling depthwise separable convolution with dense connection before the global average pooling is higher than that of two-layer depthwise separable convolution without dense connection. Therefore, the depthwise separable

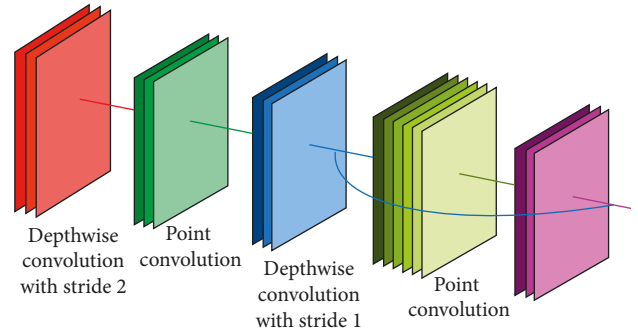


FIGURE 6: Diagram of the Dense2-MobileNet model.

convolution layer before global average pooling is also densely connected.

3.3. Dense-MobileNet Performance Analysis.

Dense-MobileNet model is constructed by adding dense connections in MobileNet. By setting a small hyperparameter growth rate, it achieves less parameters and computational complexity than that in the MobileNet model. In the MobileNet model, every 2 depthwise separable convolution layers need to reduce the dimension of the feature map by depth convolution with stride of 2. Since the sizes of the input feature maps in same dense blocks need to be the same, there are only 2 depthwise separable convolution layers included in a dense block. The growth rate in Dense-MobileNet is set by using the least difference between the number of input feature maps of each layer in MobileNets and that in Dense-MobileNet. In fact, other optimal growth rates can be selected based on the balance between the compression rate and the accuracy rate of the model.

In this paper, the Dense1-MobileNet model decomposes depthwise separable convolution into 2 separate layers, and uses 4 convolutions as a dense block. The growth rate of dense blocks in Dense1-MobileNet is $\{32, 64, 64, 128, 128, 128, 256\}$. When the parameters of the Dense1-MobileNet model decrease to 1/2 of MobileNet, its calculation decreases to 5/11 of MobileNet.

The Dense2-MobileNet model takes depthwise separable convolution as a whole and 4 convolution layers as a dense block, but only one dense connection is used. The Dense2-MobileNet model has a growth rate of $\{32, 64, 128, 256, 256, 512\}$ for dense blocks. When its model parameters drop to 1/3 of MobileNet, its calculation decreases to 5/13 of MobileNet. The parameters and calculation of each model are shown in Table 1.

The DenseNet121 model in Table 1 contains 121 convolutional layers. With 16 as growth rate, the compression ratio of transition layer is set to 0.5. That is, all output feature maps in the previous dense block are used as input feature maps in transition layer, and the number of output feature maps in this layer is half of the number of input feature maps. As can be seen from Table 1, DenseNet121 model is affected by dense connections, which has fewer parameters but a large amount of computation. At the same time, the parameters and computations of the two improved

TABLE 1: The parameters and calculation of each model.

| Network model | Calculations (millions) | Parameters number (millions) |
|------------------|-------------------------|------------------------------|
| DenseNet121 | 1364.7 | 1.78 |
| MobileNet | 568 | 3.21 |
| Dense1-MobileNet | 258 | 1.51 |
| Dense2-MobileNet | 217 | 1.12 |

Dense-MobileNets models are less than those of the MobileNet model.

4. Experiments and Result Analysis

In order to prove the validity of D-MobileNet models, we carry out classification experiments on Caltech-101 [18] and Uebingen Animals with Attributes, and compare the experimental results with those of the MobileNet model and the DenseNet121 model.

The Caltech-101 data set contains 9145 images in 102 classes, including 101 object classes and one background class. The number of images in each class ranges from 40 to 800. Figure 7 shows some samples in the Caltech-101 data set. In the experiments, the images in the data set are firstly labeled, and then fully scrambled. 1500 pictures are randomly selected as testing images, and the remaining pictures are used as training images.

The Uebingen Animals with Attributes database has 30475 pictures in 50 animal classes. Because the picture number is not the same in different classes, 21 largest animal classes with little difference in sample numbers are selected as our data set. There are 22742 pictures in the data set. The picture numbers in each class range from 850 to 1600. Figure 8 shows the samples in Uebingen Animals data set. Before training network, pictures in the data set are labeled and 2,000 of them are randomly selected as the test set. The rest of the pictures are used as the training data set.

The experiment uses Python language under TensorFlow framework. The model is implemented on a server equipped with NVIDIA TITAN GPU. RMSprop optimization algorithm with an initial learning rate of 0.1 is used to optimize the experiment. Depending on the number of training samples, we set different epoch numbers to reduce the learning rate. The weight initialization adopts the Xavier initialization method, which can determine the random initialization distribution range of parameters according to the number of inputs and outputs at each level. It is a uniform distribution with zero initial deviation. A total of 50,000 batches are trained, with 64 samples in each batch. ReLU is used as the activation function.

Table 2 shows the classification accuracy of four classification methods on the Caltech-101 data set. From Table 2, we can see that after 30,000 iterations, the accuracy of the 4 classification models has reached a balance, and the accuracy of our 2 improved structures is higher than that of DenseNet121. Compared with the accuracy of the standard MobileNet model, the accuracy of the Dense1-MobileNets

model is lower than that of the standard MobileNet model, while the accuracy of the Dense2-MobileNets model is higher than that of the standard MobileNet model. When the number of iterations is 50000, the accuracy of the Dense1-MobileNet model decreases by 0.13%, and the structure reduces less parameters and computation. When the number of iterations is 50000, the accuracy of the Dense2-MobileNet model increases by 1.2%, and its parameters and computation are reduced relatively.

Table 3 shows the classification accuracy of 4 classification methods on the Uebingen Animals data set. From Table 3, we can see that after 30,000 iterations, the accuracy of the 4 classification models also has reached a balance, and the accuracy of our 2 improved structures is higher than that of DenseNet121. Compared with the accuracy of the standard MobileNet model, the accuracy of the Dense1-MobileNets model is lower than that of the standard MobileNet model, while the accuracy of the Dense2-MobileNets model is higher than that of the standard MobileNet model. When the number of iterations is 5000, the accuracy of the Dense1-MobileNet model decreases by 0.1%, while the accuracy of the Dense2-MobileNet model increases by 1.2%.

The above two experiments were conducted under the same hyperparameter conditions. When the number of iterations is 5000, the classification accuracy of dense network on the Uebingen Animals data set is 0.4% higher than that of the MobileNet model, but it is 4.7% lower than that of the MobileNet model on the Caltech-101 data set. From the above two experiments, it can be seen that the classification accuracies of dense connection in the Dense1-MobileNet model are lost about 1% in both data sets, while they are improved in the Dense2-MobileNet mode. The main reason is that depthwise convolution and point convolution in depthwise separable convolution realize spatial correlation and channel correlation in standard convolution, respectively. However, Dense1-MobileNet using depthwise convolution and point convolution as the separate convolution layers will destroy channel correlation and reduce classification accuracy. The input feature map of the average pooling layer in Dense2-MobileNet is the superposition of the output feature maps of the previous 2 deep separable convolutions. It makes full use of the previous feature maps, reduces the parameters and computation, and improves the classification accuracy.

In order to further illustrate the performance of our method, we tested different methods in real data and other experimental environment. In the experimental comparison, we added the comparison with DenseNet161 and MobileNetV2 [19], and the experimental settings are shown in Table 4. The data set is our own children's colonoscopy polyp data set. There are two types of samples. One includes the samples with polyps, and the other includes the samples without polyp. As shown in Figure 9, the upper row is the samples with polyps, and the lower row is the samples without polyp.

The expanded training set contains 31450 samples, including 4005 polyp samples. The test set contains 4005 samples, including 1005 polyp samples. The size of each sample is $260 * 260$. The batch size of test set is set to 10, and



FIGURE 7: Samples in the Caltech-101 data set.

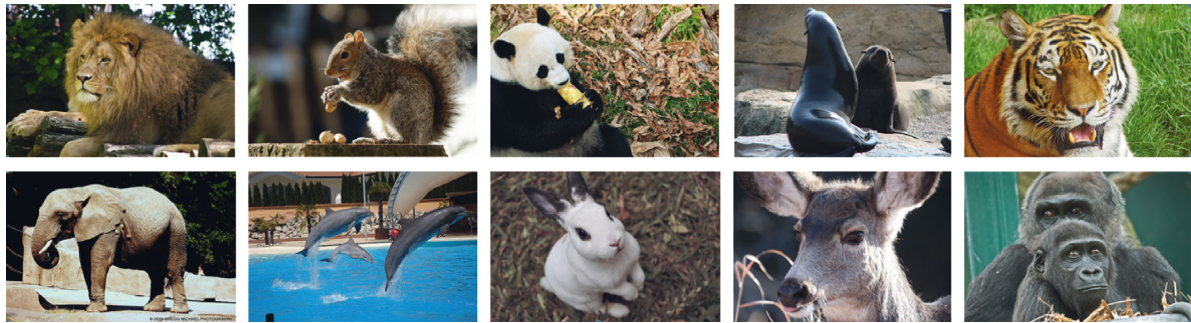


FIGURE 8: Samples in the Uebingen Animals (21) data set.

TABLE 2: Classification accuracy (%) on the Caltech-101 data set.

| Number of iterations | 30000 | 35000 | 40000 | 45000 | 50000 |
|----------------------|-------------|--------------|--------------|-------------|-------------|
| DenseNet | 72.07 | 72.27 | 72.07 | 72 | 71.9 |
| MobileNets | 76.73 | 76.6 | 76.6 | 76.8 | 76.6 |
| Dense1_MobileNet | 76.6 | 76.53 | 76.47 | 76.4 | 76.47 |
| Dense2_MobileNet | 77.6 | 77.67 | 77.87 | 77.8 | 77.8 |

TABLE 3: Classification accuracy (%) on the Uebingen Animals data set (21classes).

| Number of iterations | 30000 | 35000 | 40000 | 45000 | 50000 |
|----------------------|-------------|--------------|-------------|--------------|--------------|
| DenseNet | 91.85 | 92.15 | 91.95 | 92 | 92 |
| MobileNets | 91.6 | 91.6 | 91.6 | 91.55 | 91.6 |
| Dense1_MobileNet | 90.65 | 90.6 | 90.6 | 90.6 | 90.65 |
| Dense2_MobileNet | 92.1 | 92.05 | 92.1 | 92.05 | 92.05 |

TABLE 4: Experimental settings on children’s colonoscopy polyp data set.

| Attribute | Configuration information |
|-----------|--|
| OS | Ubuntu 14.04.5 LTS |
| CPU | Intel® Xeon® CPU E5-2670 v3 @ 2.30 GHz |
| GPU | Nvidia GeForce GTX TITAN X |
| CuDNN | CuDNN 6.0.21 |
| CUDA | CUDA 18.0.61 |
| Framework | PyTorch |

the initial learning rate is 0.1. Every network trains 200 epochs in total, and the learning rate decreases to half of the previous in the 50th epoch and then decays by half every 20 epoch. The average recognition accuracy of the last 100 epochs is taken as the final recognition result, as shown in Table 5.

Because there are only two types of test data sets, the classification accuracy of all methods is relatively high, all of which are over 96%. As can be seen from Table 5, the accuracy of Dense2_MobileNet (using full connection layer) is a little better than those of DenseNet121, MobileNet, and MobileNetV2, and slightly lower than that of DenseNet161. However, DenseNet161 is a deeper network with a large amount of parameters and calculation. In our experiments, the parameters and calculation of DenseNet161 are about 26.48 M and 10360.23 M, respectively, and the parameters of MobileNetV2 are about 2.23 M and 479.28 M, respectively. Although MobileNetV2 makes the network more lightweight, its parameter amount and calculation amount are still more than twice of our Dense_MobileNets. Therefore, the Dense_MobileNets still has certain advantages in the comprehensive evaluation of the

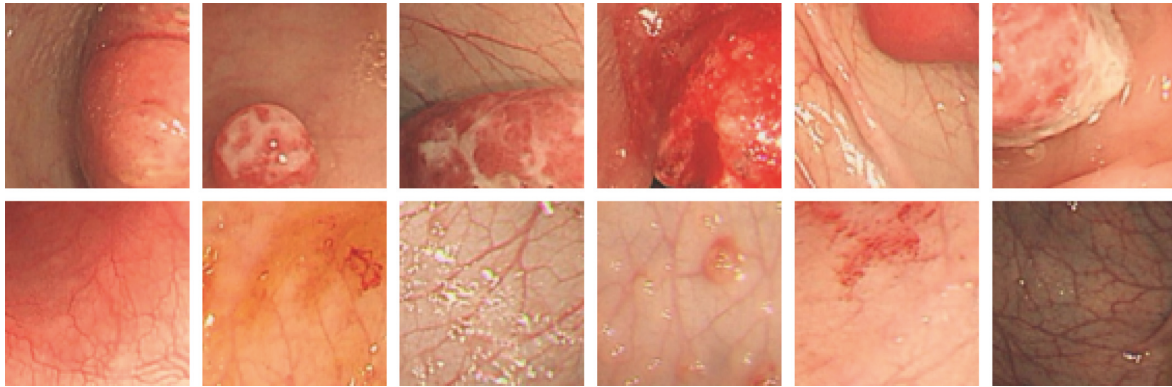


FIGURE 9: Samples in children's colonoscopy polyp data set.

TABLE 5: Classification accuracy (%) on children's colonoscopy polyp data set.

| Network | Accuracy |
|------------------|----------|
| DenseNet121 | 96.35 |
| DenseNet161 | 96.57 |
| MobileNet | 96.45 |
| MobileNetV2 | 96.46 |
| Dense1_MobileNet | 96.42 |
| Dense2_MobileNet | 96.48 |

accuracy of classification, the number of parameters, and the amount of calculation.

5. Conclusions

The memory intensive and highly computational intensive features of in deep learning restrict its application in portable devices. Compression and acceleration of network models will reduce the classification accuracy.

This paper introduces the Dense-MobileNet model with dense blocks for image classification. The dense blocks are used as the basic structure to improve the structure of MobileNet, and two improved models are proposed. These two models can reduce the parameters and calculation by setting the hyperparameter growth rate. At the same time, experiments show that Dense2-MobileNet can also increase the accuracy of classification. Compared with the MobileNet model, although the classification accuracy of Dense1-MobileNet is reduced, it reduces the number of parameters by at least half and the amount of calculation by nearly half. Generally speaking, the models proposed in this paper can be better applied to mobile devices.

Data Availability

All data sets are public data sets that can be downloaded online.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

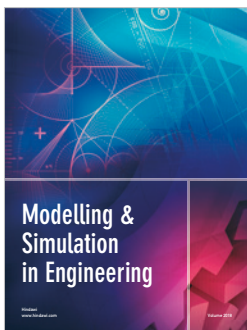
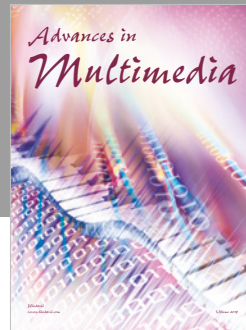
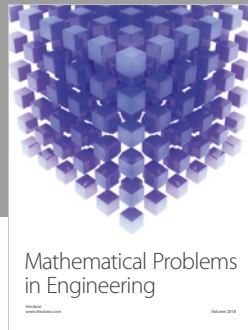
Acknowledgments

This work was supported by the National Defense Pre-Research Foundation of China (7301506), National Natural Science Foundation of China (61070040), Education Department of Hunan Province (17C0043), and Hunan Provincial Natural Science Fund (2019JJ80105).

References

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, MIT Press, Cambridge, MA, USA, 2012.
- [2] N. Wang and D. Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, pp. 809–817, MIT Press, Cambridge, MA, USA, 2013.
- [3] W. Wang, C. Tang, X. Wang, Y. Luo, Y. Hu, and J. Li, "Image object recognition via deep feature-based adaptive joint sparse representation," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 8258275, 9 pages, 2019.
- [4] W. Wang, Y. Yang, X. Wang, W. Wang, and J. Li, "The development of convolution neural network and its application in image classification: a survey," *Optical Engineering*, vol. 58, no. 4, Article ID 040901, 2019.
- [5] F. Li, C. Wang, X. Liu, Y. Peng, and S. Jin, "A composite model of wound segmentation based on traditional methods and deep neural networks," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 4967290, 1 pages, 2018.
- [6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, May 2015.
- [7] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [9] M. Denil, B. Shakibi, L. Dinh, M. A. Ranzato, and N. De Freitas, "Predicting parameters in deep learning," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2148–2156, Lake Tahoe, NV, USA, December 2013.

- [10] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," in *Advances in Neural Information Processing Systems*, pp. 1269–1277, MIT Press, Cambridge, MA, USA, 2014.
- [11] S. Han, H. Mao, and W. J. Dally, "Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding," 2015, <https://arxiv.org/abs/1510.00149>.
- [12] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *Proceedings of the International Conference on Machine Learning*, pp. 2285–2294, Lille, France, July 2015.
- [13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, <https://arxiv.org/abs/1503.02531>.
- [14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, <https://arxiv.org/abs/1602.07360>.
- [15] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, <https://arxiv.org/abs/1704.04861>.
- [16] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, Salt Lake City, UT, USA, June 2018.
- [17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu, HI, USA, July 2017.
- [18] F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," in *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*, p. 178, Washington, DC, USA, June 2004.
- [19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Salt Lake City, UT, USA, June 2018.



Hindawi

Submit your manuscripts at
www.hindawi.com

