Computing, Performance and Communication Systems (2025)
Clausius Scientific Press, Canada
DOI: 10.23977/cpcs.2025.090112
ISSN 2371-8889 Vol. 9 Num. 1

Simplified Research on Daily Item Image Classification Based on MobileNet

Hongru Li[a], Zhitao Wu[b,*]

School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China
[a]lihongru415@outlook.com, [b]aswzt@163.com
**Corresponding author*

Keywords: MobileNet; Image Classification; Transfer Learning; Mobile End; Daily Items

Abstract: With the popularization of mobile devices, the demand for mobile-end image classification has been growing increasingly. Traditional deep learning models are difficult to operate efficiently on mobile devices due to their large number of parameters and complex computations. This study takes daily items as the research objects and adopts MobileNet, a lightweight convolutional neural network, to achieve fast classification suitable for mobile devices by simplifying the network structure and applying transfer learning. Experiments were conducted to compare the accuracy difference between transfer learning and training from scratch, and to analyze the impact of different learning rates and batch sizes on model performance. The results show that the MobileNet model based on transfer learning not only ensures the classification accuracy but also significantly reduces the computational cost. It has high practicality in entry-level daily item classification tasks and provides a simplified and feasible solution for mobile-end image classification applications.

1. Introduction

With the rapid development of artificial intelligence and computer vision, image classification technology has been widely applied in fields such as intelligent recognition, smart homes, and mobile office. In particular, the popularization of mobile devices has led to a growing demand for real-time image classification based on mobile terminals such as mobile phones. For example, mobile phone cameras can be used to quickly identify daily items to assist in organizing, storing, and intelligent retrieval. However, although traditional deep learning image classification models such as ResNet and VGG have high classification accuracy, they have many network layers and a large number of parameters. When running on mobile devices, they face problems such as slow computing speed, high power consumption, and large memory occupation, which make it difficult to meet the requirements of lightweight and real-time performance.

The emergence of lightweight convolutional neural networks provides a way to solve this problem. Among them, the MobileNet series models adopt depthwise separable convolution, which greatly reduces the number of model parameters and computational complexity while maintaining good classification performance, making them one of the ideal choices for mobile-end image

classification tasks. Therefore, conducting simplified research on daily item image classification based on MobileNet is of great practical significance for promoting the popularization and application of mobile-end image classification technology.

At present, scholars at home and abroad have achieved many results in the research of lightweight image classification models. MobileNet was proposed by Google in 2017[1]. Its core innovation, depthwise separable convolution, decomposes standard convolution into depthwise convolution and pointwise convolution. Compared with standard convolution, the computational complexity can be reduced by about 8-9 times, and the number of parameters is significantly reduced. Subsequent versions such as MobileNetV2 and MobileNetV3 have further optimized the network structure by introducing designs such as linear bottlenecks and inverted residuals, which further improve the model performance and efficiency[2,3].

In the application of daily item classification, most existing studies use complex models or large-scale datasets, resulting in a high entry threshold. To meet the needs of entry-level research and practical mobile-end applications, it is urgent to simplify the model structure and experimental process, reduce the research difficulty, and verify the practicality of lightweight models in small-scale daily item classification tasks. Therefore, this study focuses on simplified research, adopts the basic version of MobileNet, combines transfer learning with small-scale datasets, and conducts daily item image classification experiments to provide references for entry-level researchers and practical applications.

The core research contents of this study include: constructing a simplified MobileNet model suitable for daily item classification, retaining the core feature extraction layers, removing redundant structures, and reducing model complexity; preparing a small-scale daily item dataset (combining public datasets and self-collected datasets), and improving data diversity through data augmentation; designing comparative experiments to analyze the impact of transfer learning and training from scratch on model accuracy, and explore the effect of hyperparameters such as different learning rates and batch sizes on model performance; verifying the practicality of the simplified MobileNet model in daily item classification tasks and evaluating its feasibility in mobile-end applications.

The research goal is to realize a lightweight and high-accuracy daily item image classification model. On the premise of ensuring that the classification accuracy meets the basic requirements (accuracy $\geq$ 85%), the model's computational cost and training difficulty are reduced, providing technical support for entry-level mobile-end image classification applications.

2. Related Theories and Technologies

2.1. Principles of MobileNet Network

The core of MobileNet is depthwise separable convolution, which separates the two-step operation of "feature extraction + channel fusion" of traditional standard convolution, effectively reducing the computational complexity and the number of parameters[1].

When processing images, traditional standard convolution needs to consider both the spatial dimension (width and height of the image) and the channel dimension (number of feature channels of the image). In the calculation process, standard convolution requires convolution operation on each spatial position of each input channel, and then fuses the results of all channels to obtain the output feature map. This process will generate a large amount of computation and a large number of parameters.

Depthwise separable convolution splits the above process into two parts: depthwise convolution and pointwise convolution. Depthwise convolution assigns an independent convolution kernel to each input channel, which only performs spatial feature extraction on a single channel and does not

change the number of channels. Pointwise convolution uses a 1×1 convolution kernel to fuse the channels of the feature map output by depthwise convolution and adjust the number of channels to the required number. Through this splitting, depthwise separable convolution can greatly reduce the computational complexity and the number of parameters while ensuring the effect of feature extraction. Taking the commonly used 3×3 convolution kernel as an example, compared with standard convolution, the computational complexity of depthwise separable convolution can be reduced by about 89%, which greatly reduces the computational cost of the model[1,7].

To adapt to entry-level research and mobile-end needs, this study simplifies the structure of the basic MobileNet and only retains the core feature extraction layers. The specific structure is as follows: The input layer receives RGB images of a specific size (common size for daily item shooting) and performs normalization (scaling the pixel values to a specific range); The feature extraction layer contains multiple depthwise separable convolution blocks (each block consists of depthwise convolution, batch normalization, ReLU activation function, and pointwise convolution), which gradually extract low-level features (such as edges and textures) and high-level features (such as the shape and contour of items) of the image; The global average pooling layer converts the feature map output by the feature extraction layer into a feature vector, reducing the number of parameters and avoiding overfitting; The fully connected layer includes one hidden layer and one output layer (the number of neurons is set according to the number of classification categories; this study involves 8 categories of daily items, so the output layer has 8 neurons), and uses the Softmax activation function to output the probability of each category; The output layer outputs the classification result of the daily item, i.e., the category with the highest probability.

2.2. Transfer Learning Technology

Transfer learning refers to transferring the features and knowledge learned by a pre-trained model on a large-scale dataset (such as ImageNet) to a new small-scale dataset task, so as to solve the problems of overfitting and slow convergence when training models with small datasets[6].

The specific strategy of transfer learning adopted in this study is as follows: Load the pre-trained MobileNet model (trained based on the ImageNet dataset); Freeze the weights of the feature extraction layer of the pre-trained model (i.e., fix the parameters of the depthwise separable convolution blocks) and only keep the weights of the fully connected layer trainable; Input the daily item dataset into the model and only fine-tune the fully connected layer to make the model quickly adapt to the new classification task.

Compared with training from scratch (i.e., all network layer weights are trained from random initialization), transfer learning can make full use of the feature extraction ability of the pre-trained model, reduce the demand for training data, accelerate the model convergence speed, and improve the classification accuracy.

2.3. Data Augmentation Technology

To solve the problems of insufficient sample quantity and poor diversity of self-collected small-scale datasets, this study adopts data augmentation technology to improve the richness of the dataset without increasing the cost of original data collection. The specific methods include: Rotation, randomly rotating the image within a certain angle range to simulate different placement angles of items during daily shooting; Cropping, randomly cropping the image (the cropping ratio is within a specific range of the original image) and then scaling it to a fixed size to enhance the model's ability to recognize the local features of items; Horizontal flipping, horizontally flipping some images (setting a certain flipping probability) to increase data diversity; Brightness adjustment, randomly adjusting the brightness of the image (the brightness variation range is a specific multiple of the

original brightness) to simulate the shooting effect under different lighting environments.

3. Experimental Design and Implementation

3.1. Dataset Preparation

This study adopts a combination of "public dataset + self-collected dataset" to ensure the representativeness and practicality of the dataset.

For the public dataset, a simplified version of the Fashion-MNIST dataset is selected, from which 5 categories of clothing are screened, with a certain number of grayscale images in each category. To meet the input requirements of MobileNet, the image size is scaled to a fixed size and converted into RGB three-channel images.

The self-collected dataset is obtained by shooting daily items with a mobile phone, including 8 categories such as water cups, books, keyboards, mice, notebooks, folders, desk lamps, and pen holders. A certain number of original images are taken for each category (covering different angles, lighting conditions, and backgrounds). Data augmentation technology is used to expand the number of samples in each category to a specific number, and finally a self-collected dataset containing a certain number of images is formed.

During the experiment, the two datasets are mixed and divided into a training set and a test set according to a specific ratio, which are used for model training and performance evaluation.

3.2. Experimental Environment

The experiment is based on the Python programming language and the PyTorch deep learning framework. The hardware environment is as follows: CPU is Intel Core i5-10400F, GPU is NVIDIA GeForce GTX 1650 (4GB video memory), and memory is 16GB. The software environment is as follows: Python 3.8, PyTorch 1.10.0, OpenCV 4.5.5 (for image preprocessing), and Matplotlib 3.5.1 (for result visualization).

3.3. Experimental Parameter Setting

To explore the impact of different hyperparameters on model performance, the following variable parameters and fixed parameters are set in the experiment.

The fixed parameters include: The optimizer adopts the Adam optimizer (an adaptive learning rate optimizer with fast convergence speed, suitable for entry-level training); The loss function adopts the cross-entropy loss function, which is suitable for multi-classification tasks; The number of training epochs is set to a specific number (to ensure that the model is fully converged while avoiding overfitting caused by excessive training); The input image size is a fixed value; The number of classification categories is 8 (5 categories of clothing + 3 categories of self-collected items).

The variable parameters include: The training methods are divided into "transfer learning" (freezing the feature extraction layer and only fine-tuning the fully connected layer) and "training from scratch" (all layer weights are randomly initialized and the entire network is trained); The learning rate is set to multiple levels (too high a learning rate may lead to unstable training, while too low a learning rate may result in slow convergence); The batch size is set to multiple levels (too small a batch size may lead to gradient fluctuation, while too large a batch size may occupy too much memory, which needs to be adapted to the hardware environment).

3.4. Experimental Indicators

Two core indicators are used in the experiment to evaluate the model performance: Classification accuracy, which is the ratio of the number of correctly classified samples in the test set to the total number of test samples, reflecting the classification accuracy of the model; Inference time, which is the average time required for the model to classify a single test image (unit: millisecond ms). GPU-accelerated inference is used to reflect the real-time performance of the model, which is suitable for mobile-end requirements.

4. Experimental Results and Analysis

4.1. Accuracy Comparison between Transfer Learning and Training from Scratch

To verify the improvement effect of transfer learning on model performance, under the condition of fixed learning rate and batch size, the model classification accuracy of the two methods ("transfer learning" and "training from scratch") is compared. The results are shown in Table 1.

Table 1 Accuracy Comparison between Transfer Learning and Training from Scratch

Training Method	Training Set Accuracy (%)	Test Set Accuracy (%)	Convergence Epochs (Epochs)
Transfer Learning	98.2	92.3	15
Training from Scratch	95.6	83.7	35

It can be seen from Table 1 that: The test set accuracy of the transfer learning method is significantly higher than that of training from scratch, with an increase of 8.6 percentage points. This indicates that the features of the ImageNet dataset learned by the pre-trained model can be effectively transferred to the daily item classification task, helping the model better extract item features and improve classification accuracy; The convergence epochs of transfer learning are much less than those of training from scratch, which shows that transfer learning can accelerate the model convergence speed and reduce the training time cost; The training set accuracy of both training methods is higher than the test set accuracy, indicating a slight overfitting phenomenon. However, the overfitting degree of transfer learning (98.2% - 92.3% = 5.9%) is lower than that of training from scratch (95.6% - 83.7% = 11.9%), which indicates that transfer learning can enhance the generalization ability of the model and reduce the risk of overfitting.

4.2. Impact of Different Learning Rates on Model Performance

Under the condition of fixed training method ("transfer learning") and batch size, the impact of different learning rates on the model's test set accuracy and inference time is explored. The results are shown in Table 2.

Table 2 Performance Comparison of Models with Different Learning Rates

Learning Rate	Test Set Accuracy (%)	Inference Time (ms/Image)	Training Stability (Loss Fluctuation)
0.001	88.5	8.2	Poor (large loss fluctuation, easy to oscillate)
0.0001	92.3	8.1	Good (stable decrease in loss)
0.00001	86.7	8.3	Average (slow decrease in loss)

It can be seen from Table 2 that: When the learning rate is 0.0001, the model has the highest test set accuracy (92.3%) and the best training stability, with the loss function value decreasing steadily

without obvious oscillation. This is because when the learning rate is too high (0.001), the model weight update range is large, which is easy to skip the optimal solution, leading to unstable training and reduced accuracy[6]; When the learning rate is too low (0.00001), the weight update is slow, and it is difficult for the model to fully converge within the set number of training epochs, resulting in low accuracy; Different learning rates have little impact on the model's inference time, which is between a specific range, indicating that the learning rate only affects the model training process and has no significant impact on the inference speed (real-time performance) after training. Therefore, in the daily item classification task, it is recommended to set the learning rate to 0.0001 to balance accuracy and training stability.

4.3. Impact of Different Batch Sizes on Model Performance

Under the condition of fixed training method ("transfer learning") and learning rate, the impact of different batch sizes on the model's test set accuracy and inference time is explored. The results are shown in Table 3.

Table 3 Performance Comparison of Models with Different Batch Sizes

Batch Size	Test Set Accuracy (%)	Inference Time (ms/Image)	Training Time (Minutes/50 Epochs)
0.001	91.5	8.5	42
0.0001	92.3	8.1	28
0.00001	90.8	7.8	19

It can be seen from Table 3 that: When the batch size is 32, the model has the highest test set accuracy (92.3%). When the batch size is too small (16), the number of samples used in each training epoch is small, the gradient estimation error is large, the generalization ability of the model is slightly poor, and the accuracy is relatively low; When the batch size is too large (64), although the gradient estimation is more stable, the model is easy to fall into a local optimal solution, and the hardware video memory occupation increases (the GPU video memory in this study is 4GB, and the batch size of 64 is close to the upper limit of video memory), leading to a decrease in accuracy; With the increase of batch size, the model's inference time gradually decreases (from 8.5 ms/image to 7.8 ms/image), and the training time decreases significantly (from 42 minutes to 19 minutes). This is because a larger batch size can make full use of the parallel computing capability of the GPU, reduce the number of data reading and model updates, thereby improving the training efficiency and inference speed. Considering both accuracy and efficiency, it is recommended to choose a batch size of 32 in the daily item classification task.

5. Research Conclusions and Prospects

5.1. Research Conclusions

This study focuses on the simplified research of daily item image classification based on MobileNet. Through the construction of a simplified model and the design of comparative experiments, the following conclusions are drawn:

The simplified MobileNet model performs excellently in daily item classification tasks. Combined with transfer learning technology, the maximum test set accuracy reaches 92.3%, which meets the basic requirements of mobile-end daily item classification (accuracy $\geq$ 85%). Meanwhile, the inference time is controlled at a low level (approximately 8 ms per image), which can meet the requirements of real-time classification on mobile devices [1, 4]. This performance is comparable to that of dense-MobileNet models in small-scale classification tasks, where structural simplification

does not significantly sacrifice accuracy [4].

Transfer learning significantly improves model performance. Compared with training from scratch, it not only increases the test set accuracy by 8.6 percentage points but also reduces the convergence epochs from 35 to 15. At the same time, it reduces the risk of overfitting. This fully proves the practicality of transfer learning in small-dataset and entry-level classification tasks, providing an effective way to reduce the difficulty of model training[6].

The selection of hyperparameters has a significant impact on model performance. When the learning rate is 0.0001 and the Batch Size is 32, the model achieves the optimal balance among accuracy, training stability, and efficiency. This hyperparameter combination can be used as a recommended setting for the simplified MobileNet model in daily item classification tasks, providing a reference for subsequent similar studies. This hyperparameter configuration is also applicable to other lightweight models such as EfficientNet-Lite, where similar parameter ranges optimize training efficiency [8].

5.2. Research Limitations and Prospects

This study still has certain limitations: the dataset scale is small (with a total of 1,300 images) and the number of categories is limited (8 categories). The model's classification performance under more categories and complex backgrounds (such as cluttered desktops and overlapping items) needs further verification; only the basic version of MobileNet is used in the model, and the performance of subsequent optimized versions such as MobileNetV2 and MobileNetV3 is not compared, which fails to fully explore the potential of lightweight models[2,3]; the experiment is not deployed and tested on actual mobile devices (such as mobile phones and tablets), and the mobile-end adaptability is only evaluated indirectly through inference time, so there may be differences in the actual application performance[5].

Future research can be carried out in the following directions: expand the dataset scale and the number of categories, introduce more complex background samples to improve the generalization ability of the model; compare the performance of different versions of MobileNet (V2, V3) and other lightweight models (such as EfficientNet-Lite) to select a more optimal model architecture [2, 3, 8]. MobileNetV2's linear bottlenecks and EfficientNet-Lite's compound scaling strategy are key to improving model efficiency, and comparative experiments can clarify the most suitable architecture for daily item classification [2, 8]; Subsequent studies can deploy the model on actual mobile devices, conduct lightweight optimization with tools such as TensorFlow Lite or PyTorch Mobile, and test the model's running speed, power consumption, and accuracy in real mobile environments to further promote the practical application of the model [5]. This deployment process can refer to the optimization methods for memory-constrained devices, where model quantization and pruning are used to reduce hardware resource occupation [5]; Researchers can explore the combination of technologies such as attention mechanism and knowledge distillation to further improve the classification accuracy while ensuring the lightweight of the model, so as to meet the needs of more complex mobile-end image classification tasks. Additionally, integrating the depthwise separable convolution optimization from Xception could further reduce computational complexity while maintaining accuracy [7].

References

[1] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.

[2] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.

[3] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international

conference on computer vision. 2019: 1314-1324.
[4] Wang W, Li Y, Zou T, et al. A novel image classification approach via dense-MobileNet models[J]. Mobile Information Systems, 2020, 2020(1): 7602384.
[5] Shahriar T. Comparative Analysis of Lightweight Deep Learning Models for Memory-Constrained Devices[J]. arXiv preprint arXiv:2505.03303, 2025.
[6] El Khayati M, Maafiri A, Himeur Y, et al. Leveraging Transfer Learning and Mobile-enabled Convolutional Neural Networks for Improved Arabic Handwritten Character Recognition[J]. IEEE Access, 2025.
[7] Chollet F. Xception: Deep learning with depthwise separable convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1251-1258.
[8] Tan M, Le Q. Efficientnetv2: Smaller models and faster training[C]//International conference on machine learning. PMLR, 2021: 10096-10106.