ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0007

Research on Mobile Plant Leaf Recognition Based on Lightweight Convolutional Neural Networks

Jingyi Wang[1, a]

[1]Hebei University of Economics and Business, Shijiazhuang, China

[a]Email: 1329344714@qq.com

Abstract

For mobile devices, recognizing plant leaves in real time is an important task. But traditional convolutional neural networks are not good at this task. They struggle to work fast enough on these devices. To fix this problem, this paper looks into optimization methods for lightweight convolutional neural networks. The paper picks three baseline models: MobileNetV2, ShuffleNetV2, and EfficientNet-B0. It also uses several compression techniques, such as knowledge distillation, pruning, and quantization. Along with structural improvements, these methods help build lighter models. The paper tests these models on the Flavia dataset. The results are clear. The optimized model reaches a Top-1 accuracy of 97.51%. Its parameter count drops to 1.4M. The model size is reduced by 75%. And its inference speed becomes much faster, which meets the needs of mobile devices.

Keywords

Lightweight Convolutional Neural Network; Plant Leaf Recognition; Mobile Deployment; Model Compression; Deep Learning.

1. INTRODUCTION

Traditional recognition methods for plant leaves rely on manual feature extraction, such as shape and texture. These methods are inefficient, subjective, and have limited generalization ability. Plant leaf recognition is a key task for smart classification, pest diagnosis, and ecological surveys. To overcome the drawbacks of traditional approaches, deep learning has become a powerful alternative. In recent years, it has made great progress in image classification. Convolutional neural networks, in particular, can automatically learn high-level semantic features and thus improve the accuracy of leaf recognition [1].

Currently, models like MobileNet, ShuffleNet, and EfficientNet perform well on general image classification tasks. They reduce model size and computation cost a lot while keeping high accuracy. They do this by improving convolution structures, cutting down the number of channels, and simplifying network layers. So they offer a useful solution for visual tasks on mobile devices.

But for plant leaf recognition, these models still have some shortcomings. Most of them are general designs and are not made for fine details like leaf veins and serrated edges. Also, there is always a trade-off between making the model lightweight and keeping its recognition accuracy.

This paper works on plant leaf recognition for mobile devices. It has three main questions to answer. First, it checks how well the current lightweight CNNs perform on leaf recognition and for which use cases they are suitable. Second, it looks at ways to make these models better. That means improving accuracy and making them work well on mobile devices. This is done by

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0007

changing network structures and using compression techniques. Third, it tests the improved model to see if it can meet real mobile needs in terms of accuracy, file size, and processing speed.

To build a mobile plant leaf recognition model that balances accuracy and speed, this study uses three techniques: comparative experiments, structural improvements, and model compression. The experiments are run on the public Flavia leaf dataset.

2. CURRENT STATUS OF LIGHTWEIGHT CONVOLUTIONAL NEURAL NETWORKS IN PLANT LEAF RECOGNITION

2.1. Overview of Mainstream Lightweight Convolutional Neural Networks

Lightweight convolutional neural networks achieve a balance between accuracy and efficiency. They do this by innovating convolution methods and optimizing network structures. Their main goal is to reduce computational complexity and adapt to mobile device power.

MobileNetV2 adds inverted residual blocks and linear bottleneck layers to improve feature representation and work on low-power devices. The original MobileNet (V1) uses depthwise separable convolution as its key idea. This splits standard convolution into depthwise and pointwise convolution, so computation cost drops by 8–9 times and the number of parameters decreases a lot. Both versions come from Google's MobileNet series.

Pointwise group convolution is the main way ShuffleNet reduces computation. But group convolution can isolate channel information. To fix this, ShuffleNet uses channel shuffling. In this way, it lowers the number of parameters and keeps high accuracy. So it works well for mid-to-low-end mobile devices with very low computing power.

For high-precision mobile leaf recognition, EfficientNet offers a useful option. Its core method is a compound scaling strategy. This strategy does not adjust only one dimension. It changes network depth, width, and input resolution all together. It also optimizes the ratio among these three dimensions. As a result, it achieves higher recognition accuracy with fewer parameters[2,3,4].

2.2. Advantages and Disadvantages of Existing Methods

Lightweight CNNs have two main advantages over traditional manual feature methods. First, they automatically learn deep features like texture, shape, and edges from leaf images. This removes the need for hand-designed features. So they generalize better and work well under different lighting, angles, and growth conditions. Second, they have much fewer parameters and lower computation cost than traditional CNNs. This makes them able to run on mobile devices. They can also adjust their network size based on the device's computing power. So they can fit different types of devices.

Lightweight models for plant leaf recognition face three main problems in practice. First, most studies test only in labs. They ignore real mobile conditions like limited computing power, battery use, changing light, and complex backgrounds. So lab results often do not match real-world performance. Second, the models themselves have design flaws. Current lightweight networks are general image classifiers. They are not tuned for fine details like leaf veins, serrated edges, or shape outlines. This leads to poor accuracy for leaves of closely related species. Third, there is a basic trade-off between being lightweight and keeping accuracy. Very light models lose too much accuracy. High-accuracy models cannot run in real time on mobile devices. So it is hard to achieve both practicality and good recognition results[5].

3. OPTIMIZATION STRATEGIES FOR LIGHTWEIGHT CNN MODELS

To address the problems of insufficient fine-grained feature extraction, difficulty in balancing lightweighting and accuracy, and poor mobile adaptability of existing lightweight models in

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0007

plant leaf recognition, this paper constructs a dual optimization framework of structural improvement + model compression to enhance model performance from three dimensions: feature enhancement, network simplification, and model compression.

3.1. Network Structural Improvement

3.1.1 Fine-Grained Feature Enhancement Module

Targeting fine-grained features such as leaf vein texture and serrated margins, leaf vein texture enhancement modules and leaf margin feature capture modules are newly added and embedded into the middle-level feature extraction stage of the network to precisely enhance the representation of key features.

(1) Leaf Vein Texture Enhancement Module: It adopts 3×3 small-kernel depthwise separable convolution combined with 1×3 and 3×1 multi-scale convolution to capture leaf vein texture details from horizontal and vertical dimensions; paired with a lightweight SE channel attention module, it adaptively strengthens important feature channels and suppresses invalid channels without additional computational burden, significantly improving the distinguishability of leaf vein features.

(2) Leaf Margin Feature Capture Module: It uses dilated convolution with dilation rates of 2 and 4 to expand the receptive field without increasing kernel size and computational complexity, accurately capturing edge features such as serrated margins and contours, solving the problem that small kernels have insufficient receptive fields and struggle to capture global edge features[6].

3.1.2 Network Structural Simplification

On the basis of feature enhancement, redundant network structures are simplified to significantly reduce the number of parameters.

(1) Simplify redundant high-level convolutional and pooling layers, reducing unnecessary feature dimensionality reduction operations to avoid the loss of fine-grained features.

(2) Use global average pooling and 1×1 convolution instead of fully connected layers. This cuts the number of parameters by over 90% and keeps global feature information.

(3) Use the normal 224×224 size for leaf images. Also reduce the number of downsampling steps. This stops fine details like leaf edges and textures from being lost due to too much downsampling.

3.2. Model Compression Technology

To further reduce model size and speed up inference, we apply three compression techniques after improving the network structure. These are knowledge distillation, channel pruning, and INT8 weight quantization. They help us keep accuracy loss within a controlled range[7].

3.2.1 Knowledge Distillation

To improve the lightweight model's recognition accuracy, we use a teacher–student distillation strategy. The teacher is EfficientNet-B0, a high-accuracy model with a large number of parameters. The student is MobileNetV2, which has been improved in structure. The student learns from the teacher during training. As a result, the lightweight model achieves better accuracy.

To help the student model capture correlation information between classes, it should learn not only the true labels but also the class probability distribution from the teacher. So we choose a soft label distillation method. The loss function combines cross-entropy loss and KL divergence with weighted terms, and we set the temperature to 10. This lets the student learn from both sources. Experiments show this approach can boost the student's accuracy by 2%–3% without adding more parameters or computation.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0007

3.2.2 Channel Pruning

An L1 regularization iterative pruning strategy is adopted to address convolutional layer channel redundancy, with differentiated pruning rates set by layer: low-level convolutional layers extract basic texture features with a pruning rate of 5%–10%; middle and high-level convolutional layers extract advanced semantic features with higher redundancy and a pruning rate of 10%–20%.

A pruning–fine-tuning iterative mode is adopted: after pruning 5% of channels each time, the model is fine-tuned with a small learning rate to repair accuracy loss caused by pruning. Ultimately, the number of model parameters can be reduced by 30%–40%, computational complexity by 25%–35%, with accuracy loss controlled within 1%–2%.

3.2.3 INT8 Weight Quantization

A combined scheme of post-training quantization and quantization-aware training is adopted to convert the model's 32-bit floating-point weights into 8-bit integers, significantly reducing model size and improving inference speed.

(1) Post-training Quantization: Directly convert the weights of the trained 32-bit floating-point model to INT8 without retraining, reducing model size by 75% and inference speed is improved by 2-3 times, but causing certain accuracy loss.

(2) Quantization-Aware Training: Simulate quantization errors and adapt to accuracy deviations caused by quantization in advance during training, fine-tuning model parameters to control accuracy loss within 1%–2%, balancing model size, speed, and accuracy.

The three compression technologies are combined in the order of distillation → pruning → quantization, with superimposed compression effects and overall accuracy loss controlled within 5%, achieving the optimal balance between lightweighting and high accuracy.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1. Experimental Setup and Dataset

4.1.1 Dataset

To improve the model's generalization, we apply data augmentation: random rotation (0–360°), horizontal flipping, and random changes in brightness and contrast. These simulate leaf images taken under different angles and lighting. We then split the dataset into training, validation, and test sets at a ratio of 8:1:1. The Flavia leaf dataset used in the experiment has 4,248 images of 32 common plants, with simple backgrounds and clear leaf contours. This makes it well suited for fine-grained leaf feature recognition research.

4.1.2 Experimental Environment and Parameter Settings

The training uses a batch size of 32, an initial learning rate of 0.001, and the AdamW optimizer. The maximum number of epochs is 50, and early stopping is applied: training halts if the validation loss does not decrease for 5 consecutive epochs. The experiments run on an NVIDIA RTX 3060 GPU with 16GB memory, using PyTorch 2.0 as the deep learning framework. The model performance is evaluated by Top-1 accuracy, parameter count, model size, and inference speed.

4.2. Performance Comparison of Baseline Models

To compare their performance, we select three baseline models: MobileNetV2, ShuffleNetV2, and EfficientNet-B0. All of them are trained and tested under the same experimental conditions.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0007

Table 4-2. Performance Comparison of Baseline Models

Model	Top-1 Accuracy	Parameters (M)
MobileNetV2	88.51%	2.35
ShuffleNetV2	87.40%	1.36
EfficientNet-B0	81.88%	4.14

Among the three lightweight models, ShuffleNetV2 has the fewest parameters (1.36M) but the lowest accuracy (87.40%). EfficientNet-B0 has the most parameters (4.14M) but its accuracy is only 81.88%. MobileNetV2 achieves 88.51% accuracy with 2.35M parameters.

4.3. Performance of the Optimized Model

To significantly improve model performance, we apply four optimization methods to the base model MobileNetV2. These include structural improvement, knowledge distillation, channel pruning, and INT8 quantization. As a result, the optimized model performs much better than the original.

Table 4-3. Comparison of Key Indicators Before and After Optimization

Model	Top-1 Accuracy	Parameters (M)	Model Size (MB)
Original MobileNetV2	88.51%	2.35	9.4
Optimized Mode	97.51%	1.40	2.35

The optimized model meets real-time mobile recognition requirements. On a Snapdragon 870 Android phone, INT8 quantization makes inference about 2.5 times faster, and the inference time for one leaf image is about 15 ms. The model is also very lightweight. Its size drops from 9.4MB to 2.35MB, a 75% reduction, and its parameter count falls from 2.35M to 1.40M, a cut of about 40%. This fits mobile storage limits. In addition, recognition accuracy improves a lot. Top-1 accuracy goes up from 88.51% to 97.51%, which is about a 9% gain. The training and validation accuracy curves rise steadily, and validation loss keeps decreasing. There is no overfitting, so the model has strong generalization.

4.4. Result Discussion

For mobile plant leaf recognition, the optimized model reaches a Top-1 accuracy of 97% or higher. Its parameter count is no more than 1.5M, and its model size stays below 3MB. These numbers show that the model strikes a good balance.

The optimized model has high accuracy, a lightweight design, and fast inference speed. These three features make it suitable for various mobile devices, from high-end to low-end. It also outperforms the baseline models in all these aspects.

The recognition accuracy for very similar leaf species can still be improved. In addition, the validation is done only on the Flavia dataset, which has simple backgrounds, so real-world conditions like complex backgrounds, extreme lighting, and blurred leaves are not covered. Also, we have not measured inference time or power consumption on mobile devices. These are the main limitations of this research.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0007

5. CONCLUSION

To balance lightweight and high accuracy for mobile plant leaf recognition, we construct a dual optimization framework that includes structural improvement and model compression. We evaluate three baseline models – MobileNetV2, ShuffleNetV2, and EfficientNet-B0 – and select MobileNetV2 as the base model because it offers the best trade-off between accuracy and efficiency. In the structural improvement part, we embed special modules to extract vein and margin features and also simplify redundant network layers. In the compression part, we use knowledge distillation, channel pruning, and INT8 quantization to systematically reduce model size and computation.

The optimized model has its parameter count reduced to 1.40M, and its model size cut by 75%. Its inference speed becomes 2-3 times faster. Its Top-1 accuracy reaches 97.51% on the Flavia dataset, which is about 9% higher than the original MobileNetV2.

To address the current shortcomings, future work will focus on three areas. First, we will introduce fine-grained feature fusion and multi-modal recognition to improve discrimination among closely related leaf species. Second, we will collect more diverse data, including complex backgrounds, extreme lighting, and blurred leaves, to enhance model robustness. Third, we will measure inference time and power usage on mobile devices to refine our deployment evaluation.

ACKNOWLEDGMENTS

The completion of the experiments in this research is inseparable from the support and help of teachers, classmates, and family members. I hereby express my sincere gratitude to my supervisors Professor Li Yu and Wu Zhujiao. Thank you for your careful guidance, patient answers, and selfless help in topic selection, research ideas, experimental design, and thesis writing. Your rigorous academic attitude, profound academic attainments, and pragmatic research spirit have benefited me greatly.

Thank you to my classmates for their exchanges and encouragement during my studies and research, and for their help and support in experiments, enabling me to smoothly solve problems encountered in the research.

I sincerely thank my parents and family for their silent companionship and selfless dedication over the years. Your support is the most solid backing for my studies, allowing me to focus on learning without distractions.

Finally, I would like to express my sincere thanks to all teachers, classmates, friends, and family who care for, support, and help me.

REFERENCES

[1] Wang, Z. R., & Yan, C. L. (2011). A review of image feature extraction methods. Journal of Jishou University (Natural Science Edition), 32, 43–47.

[2] Xu, H. T., Cai, W. Y., Zhang, M. Y., & others. (2025). A lightweight convolutional neural network model combining MobileNet and ShuffleNet. Journal of Hangzhou Dianzi University (Natural Sciences), 45, 50–61.

[3] Jin, Z. W., & Ge, D. Y. (2026). Image recognition method based on improved ShuffleNet network. Computer Applications and Software, 43, 182–188.

[4] Zhou, L., & Cao, C. W. (2026). Plant pest and disease recognition method based on knowledge distillation and EfficientNet_v2. Jiangsu Agricultural Sciences, 54, 269–276.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0007

[5] Xiao, Y. (2025). Research on image recognition of crop diseases and insect pests based on lightweight convolutional neural network [Master's thesis]. University of Science and Technology Liaoning.

[6] Ma, Y., Zhi, M., Yin, Y. J., & Ping, P. (2022). Review of applications of CNN and Transformer in fine-grained image recognition. Computer Engineering and Applications, 58, 53–63.

[7] Molchanov, P., Tyree, S., Karras, T., Aila, T., & Kautz, J. (2017). Pruning convolutional neural networks for resource efficient inference. In International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1608.08710