Comparison and Evaluation of Plant Pest and Disease Image Recognition Models Based on Tensorflow and CNN

Jiner Li
Intelligent Technology and Services, School of Data Science, City University of Macau, Macau, China

Abstract: This study addresses the potential for misjudgment in human visual identification of plant diseases, which is highly susceptible to subjective factors, and the variations in accuracy and F1 score among different image recognition models in the same environment. Utilizing comparative experimental methods and literature analysis, this paper explores the performance of two pre-trained models, MobileNetV3 and EfficientNetV2, in image recognition tasks, ultimately selecting the optimal model for agricultural pest and disease identification. Initially, a dataset was collected from the internet, encompassing 17 plant species. Tomatoes, cassava, corn, and apples each had no fewer than four types of pests and diseases, along with their healthy samples. The remaining plants each had at least one type of pest and disease, along with their healthy samples. After cleaning, removing ambiguous and incorrectly labeled samples, the dataset was divided into a training set, validation set, and test set in an 8:1:1 ratio. To address the scarcity of samples, additional samples were generated through random image augmentation techniques (flipping, rotating, brightness adjustment), ensuring each sample reached a total of 200. Subsequently, model construction was carried out. Based on the make_model function, MobileNetV3 Large and EfficientNetV2 B0 pre-trained models were flexibly constructed, configured with regularization, Dropout, and other overfitting strategies, and specified with the Adamax optimizer, cross-entropy loss, and F1 score evaluation metrics. Model training was then initiated, with settings for learning rate decay, early stopping callback, and interactive specification of training epochs, while monitoring convergence in real-time. Finally, the model convergence trend was analyzed through visualization of the training curve. Comprehensive evaluation of model performance, including accuracy and F1 score, was conducted using the test set, confusion matrix, and classification report, comparing the adaptability of the two pre-trained models. Experimental results indicated that the EfficientNetV2 B0 model achieved an accuracy of 93.41%, approaching an F1 score of 92.40%. The MobileNetV3 Large model reached a precision of 93.15%, nearing an F1 score of 92.16%. Evidently, the EfficientNetV2 B0 model outperforms the MobileNetV3 Large model in terms of accuracy and F1 score, particularly in identifying "confusable pests and diseases," making it suitable for scenarios demanding higher precision. Conversely, the MobileNetV3 Large model boasts a smaller parameter count and faster training/inference speed by approximately 15% to 20%, making it ideal for edge devices with limited computational power.

Keywords: Plant Pest and Disease Classification; Pre-Trained Model (EfficientNetV2 B0/MobileNetV3 Large); ACCURACY; F1 SCORE

1. Introduction

1.1 Reasons for Topic Selection

Plant diseases, as a major challenge in agricultural production, seriously affect global food security and the stable development of the agricultural economy. Traditional disease identification relies on visual observation by agricultural experts, which is time-consuming, laborious, and has inconsistent accuracy. With the rise of artificial intelligence technology, especially the breakthroughs in deep learning in the field of image recognition, new avenues have been provided for the precise identification and timely prevention and control of plant diseases. TensorFlow, as a mainstream deep learning framework, has become an ideal choice for building plant disease recognition systems due to its powerful computing capabilities and flexible

architecture. This experiment focuses on two typical lightweight pre-trained models: the MobileNetV3 model [1], which is represented by its "lightweight and efficient" characteristics, and the EfficientNetV2 model [2], which is represented by its "balance between accuracy and efficiency". Under a unified experimental environment, a comparative analysis is conducted from four dimensions: training convergence characteristics, accuracy on the test set, weighted F1 score, confusion matrix, parameter quantity, and inference speed. The performance differences and applicable scenarios of the two types of models in plant pest and disease tasks are clarified, providing a scientific basis for agricultural AI diagnosis.

1.2 Research Status at Home and Abroad

Current research status in China: In recent years, domestic research on plant disease identification based on deep learning has flourished, with numerous scholars and institutions conducting studies in this field. Jiaobao Jiao's team from Zhejiang Sci-Tech University [3] utilized AR glasses to capture images of rice pests and diseases, and manually cleaned, filtered, integrated, and labeled the image data, ultimately obtaining images of 10 diseases and 10 pests. These images were divided into a training set and a test set in an 8:2 ratio. Then, using their optimized EfficientNet-V2 model, they identified rice images. After testing, the accuracy of their optimized model increased from 76.14% to 84.92%. Regarding crop pest and disease image recognition technology, Jingjing Wang's team from Anhui Agricultural University [4] provided a comprehensive review of the research conducted by experts and scholars at home and abroad in the field of crop pest and disease image recognition over the past decade, focusing on three aspects: image segmentation, feature extraction, and classification recognition. For example, in image segmentation, methods such as threshold segmentation and edge detection were discussed. In feature extraction, methods such as morphological feature extraction and color feature extraction were discussed. In classification recognition, methods such as neural network methods, fuzzy clustering methods, and support vector machine methods were discussed.

Current research status abroad: Foreign research has also achieved remarkable accomplishments. The team led by Chowdhury Rafeed Rahman [5] collected 1426 rice images from real-world scenarios, compared large CNN architectures (VGG16, InceptionV3) with lightweight architectures (MobileNetv2, NasNet Mobile), and proposed their self-developed "Simple CNN". Their main research finding is that among all the models, the fine-tuning method yields the most significant results, with VGG16 achieving an accuracy of 97.12% after fine-tuning, InceptionV3 reaching 96.37%, and the lightweight models MobileNetv2 and NasNet Mobile achieving accuracies of 96.12% and 96.95%, respectively. Their self-created "Simple CNN" also achieved an accuracy of 94.33%, approaching the performance of large models. Meanwhile, the team led by M Shoaib [6] comprehensively analyzed the latest research, systematically elaborated on the current application status, challenges, and future directions of deep learning in plant disease detection. Finally, they concluded that deep learning exhibits an accuracy rate of 95% and a precision rate of 90% in plant pest and disease detection.

2. Experimental Methods

2.1 Experimental Flowchart

Figure 1 shows the flowchart of the entire experimental procedure.

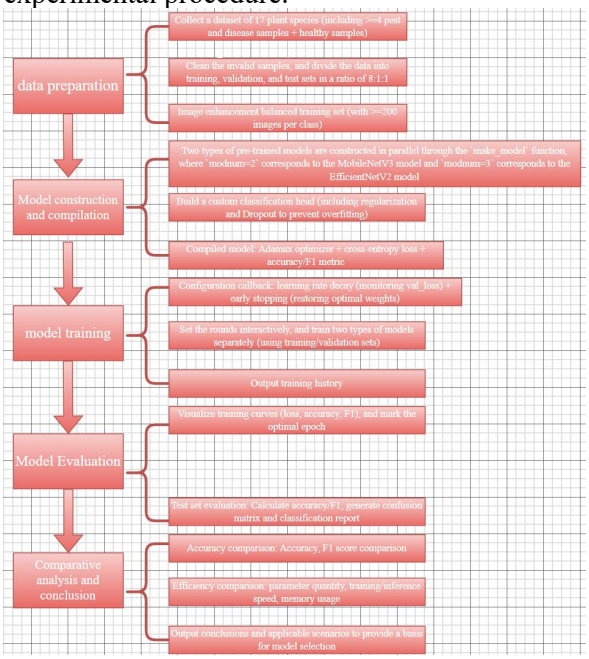

Figure 1. The Entire Experimental Process

2.2 Data Preparation

The data is sourced from Kaggle, a globally

renowned data science community. The dataset contains 17 different plant leaf diseases and pests, as well as their healthy samples, such as tomato late blight, tomato leaf roll disease, grape black rot, and so on. [7] In image recognition, imbalanced categories in the database are a common and challenging situation. If the number of samples in some common categories in the database is much larger than that in rare categories, it can lead to good performance of the model in recognizing common categories but errors in recognizing rare categories, thereby significantly reducing the accuracy of the model. Therefore, image data preprocessing becomes particularly important.

To address the aforementioned issues, this paper employs various methods. For instance, a value is input as the upper limit for the maximum number of images in each category. Here, 200 is input as the maximum number of images in each category. If a category has a smaller number of images, it will be automatically padded. If the number of images in a certain category exceeds the set upper limit, a portion of the images in that category will be randomly selected. Additionally, a function named "balance" is defined to achieve dataset balancing. The input parameters of the function include the original dataset DataFrame, the target number of samples per category n, the category column name column, the working directory working_dir, and the image size img_size, which is (224×224) in this case. The function contains two auxiliary functions: 1. `get_augmented_image(image)` performs data augmentation on the given image and returns the augmented image. This function utilizes the Albumentations library to implement data augmentation, including operations such as horizontal flipping, rotation, random brightness contrast adjustment, random Gamma value adjustment, and random cropping. 2. `dummy(image)` is a simple auxiliary function that returns the input image itself.

The main functions and steps are as follows:

1. Copy the input DataFrame and output the length of the initial dataset.
2. Create a directory named "aug_dir" to store the augmented images, and create subdirectories for each corresponding category within it.
3. For each category, generate additional samples through data augmentation to achieve the target sample size n, and save the augmented images to the specified directory.
4. Create a new DataFrame named "aug_df" that contains the paths and labels of the augmented image files.
5. Merge the original dataset and the enhanced dataset into a new dataset, and return the updated DataFrame.

Finally, the length of the enhanced dataset will be output at the end of function execution. The function's role is to achieve dataset balance through data augmentation, ensuring that the number of samples for each category is consistent, in order to improve the training effect of the model.

For dataset partitioning, the experiment divided it into a training set: validation set: test set ratio of 8:1:1.

Through these methods, this paper significantly increases the sample size of rare categories, such as durian and bitter gourd, enhancing the robustness and generalization ability of the model.

2.3 Model Construction and Model Training

In the model construction phase, the core foundation lies in transfer learning. Initially, pre-trained weights are reused. In the experiment, a function named `make_model` is defined to create a specific type of deep learning model. This function accepts four parameters: image size (224, 224), number of categories (61), learning rate fixed at 0.001, and model number (2 or 3). Depending on the value of the model number ‘modnum’, a specific pre-trained model is selected as the base model. If `modnum` is 2, the MobileNetV3 model is chosen; if it is 3, the EfficientNetV2 model is selected. Additionally, when constructing the model, the settings `weights='imagenet'` are used to load weights pre-trained on ImageNet, and `include_top=False` is set to retain only the backbone network for feature extraction, excluding the top-level classification head of the original model. Subsequently, by adding layers such as BatchNormalization and Dense, and replacing the custom classification head, the "replacement of only the top-level classification head" is achieved.

The second is the freezing and fine-tuning strategy, where base_model.trainable is set to False: the weights of the backbone network are frozen, and only the newly added custom classification head is trained.

The key optimization step is the design to prevent overfitting. For regularization, L1 and L2 regularization are added to the fully

connected layer to limit excessive weights and avoid overfitting the model to the training set. For the Dropout layer, the dropout rate is set to 0.4, randomly inactivating some neurons to reduce the dependency between neurons. The optimizer is selected as Adamax, which adapts to the convergence characteristics of lightweight models and provides a more stable learning rate adjustment. For the loss function, cross-entropy loss is adopted to fit the probability distribution optimization goal of multi-classification tasks. For evaluation metrics, in addition to accuracy, weighted F1 score is introduced to adapt to the possible class imbalance issue in the pest and disease dataset. The calculation method of F1 score is as follows.

1. Calculate true positives: Multiply the true label y_true by the predicted label y_pred, take the rounded value, and then accumulate the results.
2. Calculate possible positives: Accumulate the rounded true labels y_true.
3. Calculate the predicted positives: Accumulate the rounded predicted labels y_pred.
4. Calculate precision: It refers to the proportion of true positives among the samples classified as positive cases. It can be obtained by calculating the ratio of true_positives to (predicted_positives + K.epsilon()).
5. Calculate recall: It refers to the proportion of true positives accurately identified by the model among all true instances. It can be obtained by calculating the ratio of true_positives to (possible_positives + K.epsilon()).
6. Calculate F1 score: Apply the formula for F1 score, which is 2 * (precision * recall) / (precision + recall + K.epsilon()), where K.epsilon() is a very small value to avoid the denominator being zero.
7. Return the calculated F1 score.

In summary, the function's role is to select a specific pre-trained model based on the input parameters, construct a deep learning model with a specific structure, and apply it to image classification tasks. During the model construction process, customized neural network layers are added to accommodate specific problem requirements, and the model is compiled for training and evaluation.

In the model training stage, one of the core technical points is dynamic learning rate adjustment, which defines a ReduceLROnPlateau callback function: the metric monitored through monitor="val_loss" is the loss value on the validation set, and the callback is triggered when the loss value no longer decreases. After the loss value stops decreasing, the learning rate will be reduced by a factor of 0.4. The direction in which to decrease the learning rate will be automatically selected.

The other is the early stopping mechanism, which defines the EarlyStopping callback function: similarly, the monitored metric is the loss value on the validation set, and the callback is triggered when the loss value no longer decreases. min_delta=0 represents the minimum change in the loss value, and if the decrease in the loss value is less than this threshold, it will be considered as no improvement. [8]

In general, the role of these two callback functions is to assist in monitoring specific metrics (loss values) during the training process, and automatically adjust the learning rate or terminate training early based on predefined conditions, in order to enhance training efficiency and prevent overfitting. The defined callback functions will be passed into the model.fit function during the training process of the model to achieve the corresponding functionality.

3. Experimental Results and Analysis

3.1 Experimental Results

To objectively evaluate the pest and disease recognition performance of two models under the same environment, accuracy, F1 score, and confusion matrix are selected as evaluation metrics, as shown in Figure 2. These metrics are commonly used performance measures in classification tasks and can reflect the recognition ability of the model from different perspectives. Accuracy measures the proportion of samples correctly identified by the model as belonging to a specific pest or disease category. The F1 score is the harmonic mean of accuracy and recall, where recall measures the proportion of samples in a specific pest or disease category that can be identified by the model. Therefore, the F1 score comprehensively considers both accuracy and recall of the model and is a commonly used metric to measure the overall performance of the model. A higher F1 score indicates that the model performs well in both accuracy and recall, indicating a more balanced classification performance.

The experiment utilizes the Matplotlib library to create charts containing this information,

illustrating the trends during the training and validation processes, and marking the optimal data points, as shown in Figure 3. From the chart, it can be observed that the loss of the EfficientNetB0 model during training and validation is continuously decreasing, reaching its lowest point at the 14th epoch, which is 0.5396. In the second chart, it can be seen that both the training and validation accuracy of the model are continuously increasing, reaching their optimal values at the 12th epoch, which are 93.41%. The third chart shows the F1 score, which is also continuously increasing and reaches its optimal value at the 12th epoch, which is 92.40%.

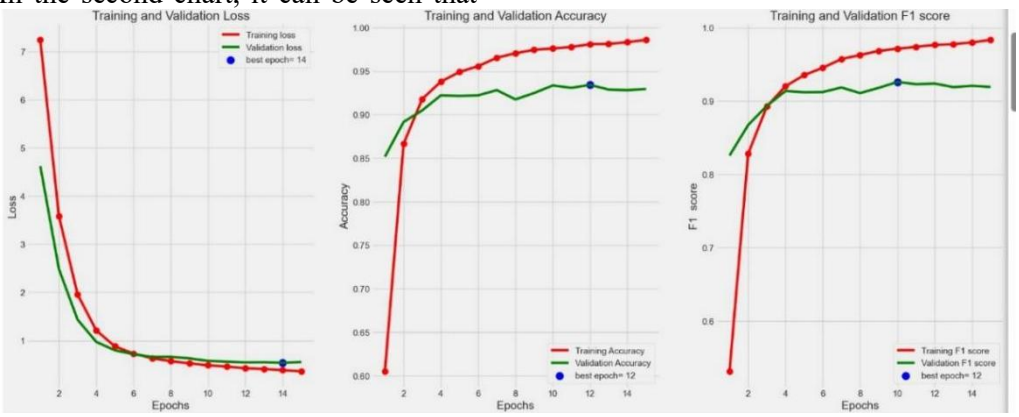

Figure 2. Loss, Accuracy, and F1 Score of the EfficientNetB0 Model on the Training and Validation Sets

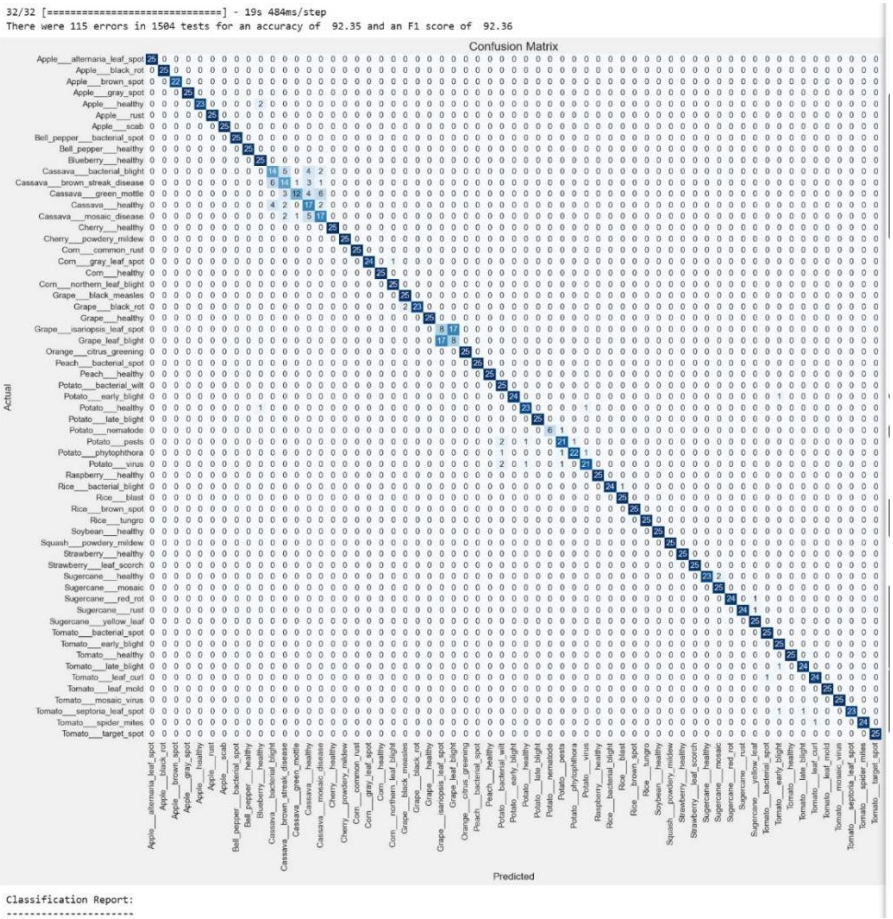

Figure 3. Confusion Matrix of EfficientNetB0 Model

According to the confusion matrix, the accuracy of the model was verified using a test dataset consisting of 1504 test data points. Among them, 115 were incorrect, while the rest were all correct. The accuracy rate reached 92.35%, and the weighted F1 score was 92.36%.

Then, train and validate the model with model_num=2, namely the MobileNetV3 model,

using the same approach. Show the changing trends during the training and validation process, as well as the confusion matrix, and highlight the optimal data points.

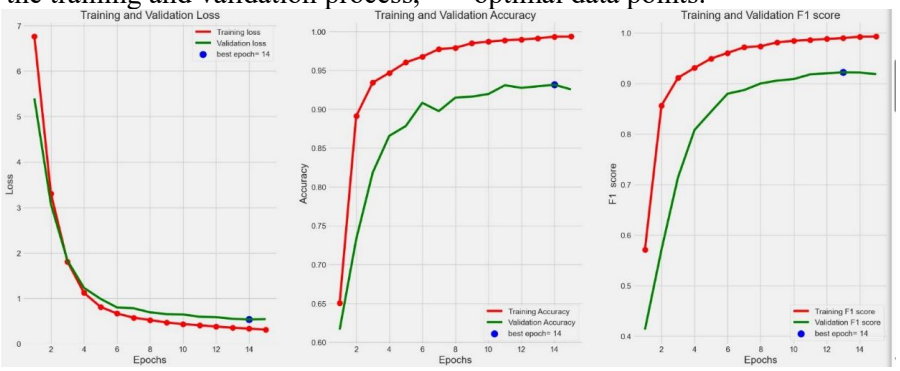

Figure 4. Loss, Accuracy, and F1 Score of the Training and Validation Sets for the MobileNetV3 Model

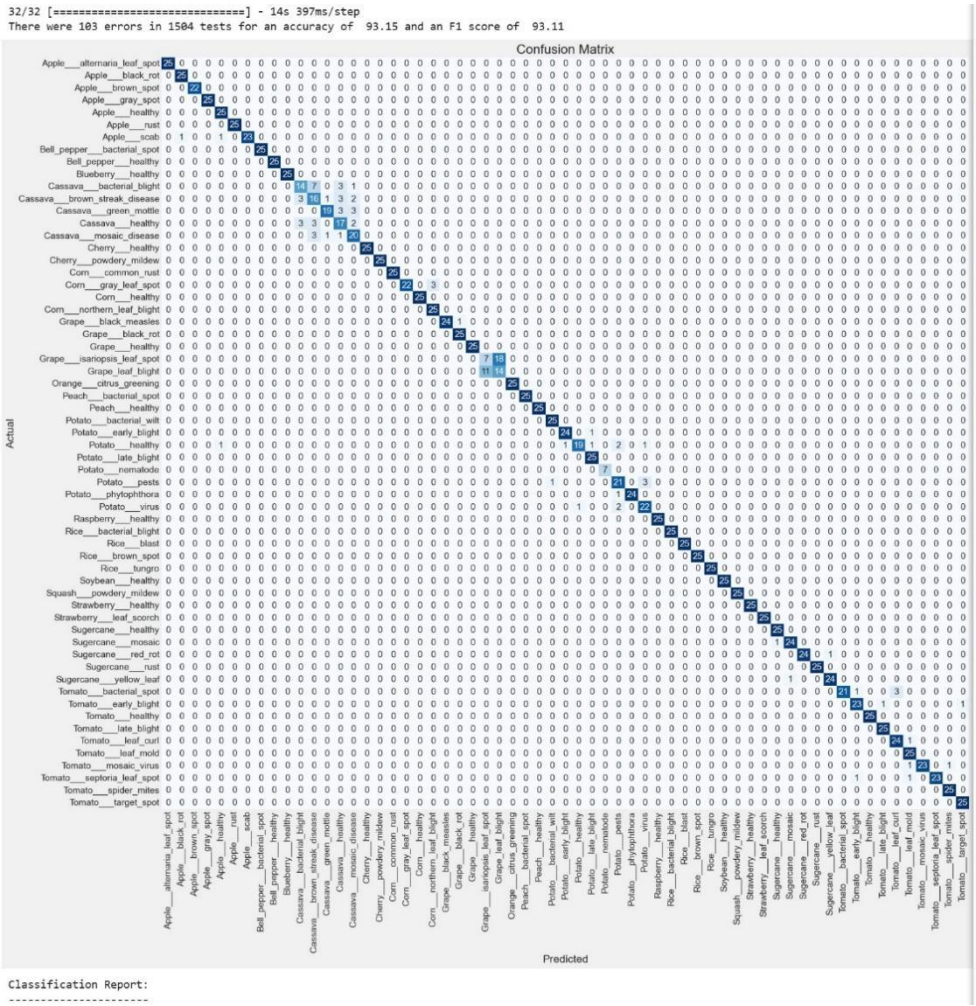

Figure 5. Confusion Matrix of MobileNetV3 Model

Similarly, Figure 4 shows that we can see that the loss of the MobileNetV3 model in training and validation in the first figure is continuously decreasing, reaching its lowest at the 14th epoch, which is 0.5353, with a slightly lower loss of 0.0043. In the second figure, we can observe that the training and validation accuracy of the model are continuously increasing, also reaching its optimal at the 14th epoch, which is 93.15%. However, it converges slightly later than the MobileNetV3 model and has an optimal accuracy that is 0.26% lower than that of the EfficientNetB0 model. The third figure shows the F1 score, which is also continuously increasing and reaches its optimal at the 13th epoch, which is 92.19%. This is 0.21% lower than the optimal F1 score of the EfficientNetB0 model.

Based on the confusion matrix (Figure 5), it can be observed that the accuracy of the model was validated using a test dataset consisting of 1504 test samples. Among them, 103 samples were incorrect, while the rest were all correct. Compared to the EfficientNetB0 model, there were 12 fewer errors, with an accuracy rate of 93.15%, which is 0.8% higher. The weighted F1 score was 93.11%, which is 0.75% higher than that of the EfficientNetB0 model.

3.2 Analysis of Experimental Results

During the training phase, EfficientNetV2 B0 converges faster, while MobileNetV3 has lower loss. For EfficientNetV2 B0, its validation set accuracy/F1 reaches its optimal at the 12th round, converging 1-2 rounds earlier than MobileNetV3 (13th/14th round), indicating its stronger feature extraction ability and faster learning of core features of pests and diseases. However, the final training/validation loss is slightly higher (0.5396 vs 0.5353), reflecting that its fit on the training set is slightly weaker than MobileNetV3. For MobileNetV3 Large, its loss decreases to a lower level, indicating a more adequate fit on the training set, but the convergence speed is slightly slower, due to its lightweight architecture (depthwise separable convolution) having slightly lower feature learning efficiency than EfficientNetV2 B0's compound scaling architecture.

During the testing phase, MobileNetV3 exhibited superior generalization capabilities, whereas EfficientNetV2 B0 suffered from a "validation-test" bias. The core contradiction lies in the fact that EfficientNetV2 B0 achieved higher accuracy/F1 scores on the validation set (93.41%/92.40%) than MobileNetV3, but lagged behind on the test set (92.35% vs 93.15%). This is a typical characteristic of "validation set overfitting" - EfficientNetV2 B0 has better adaptability to the data distribution of the validation set, but its generalization ability is inferior to MobileNetV3 when faced with an independent test set. Regarding the difference in error counts, the MobileNetV3 model had 12 fewer errors on the test set, and its F1 score advantage was more pronounced, indicating that it is more stable in recognizing "confusable pest and disease categories" (such as tomato leaf spot and cassava mosaic disease), whereas EfficientNetV2 B0 made more misclassifications on these fine-grained categories.

In practical application scenarios, if the dataset distribution of the deployment scenario is highly similar to the training/validation set (such as pest and disease images from fixed origins and under fixed shooting conditions), EfficientNetV2 B0 can be selected, as its high accuracy on the validation set can quickly demonstrate its advantages. If the deployment scenario is in a real agricultural environment (with complex image distribution and diverse shooting conditions), MobileNetV3 Large is preferred, as its high accuracy/F1 score and low error rate on the test set can ensure recognition stability on unknown data, and its lightweight architecture is more suitable for edge devices (such as agricultural inspection terminals).

4. Summary

This paper focuses on the research of precise identification of plant diseases and pests. By collecting pest and disease samples as well as healthy samples of 17 plant species through the network, and after cleaning, balancing, and an 8:1:1 split, we constructed two types of pre-trained models, MobileNetV3 Large and EfficientNetV2 B0, based on TensorFlow. We configured anti-overfitting strategies and adapted optimizers and evaluation metrics to complete the training. The experimental results show that EfficientNetV2 B0 converges faster and has slightly better accuracy (93.41%) and F1 score (92.40%) on the validation set, while MobileNetV3 Large performs better on the test set (accuracy 93.15%, F1 score 93.11%), with stronger generalization ability and being more lightweight and efficient. The two types of models have their respective suitable scenarios. EfficientNetV2 B0 is suitable for high-precision demand scenarios with stable data distribution, while MobileNetV3 Large is more suitable for real-world agricultural edge deployment scenarios with limited computing power, providing a scientific basis for model selection in agricultural AI disease diagnosis.

References

[1] Bi, C., Xu, S., Hu, N., Zhang, S., Zhu, Z., & Yu, H. 2023. Identification method of corn leaf disease based on improved Mobilenetv3 model. https://scholar.google.com/scholar?hl=zh-CN&as_sdt=0%2C5&q=mobilenetv3+plant+disease&oq=MobileNetV3++plant

[2] Devi, R. S., Kumar, V. R., & Sivakumar, P. 2023. EfficientNetV2 Model for Plant

Disease Classification and Pest Recognition. https://scholar.google.com/scholar?hl=zh-CN&as_sdt=0%2C5&q=EfficientNetV2+plant+disease&oq=

[3] Jiao Jiabao, Li Lingyi, Liu Yongjian, 2025. Research on Rice Pest and Disease Recognition System Based on Improved EfficientNet-V2.https://scholar.google.com/scholar?hl=zh-CN&as_sdt=0%2C5&q=%E7%84%A6%E4%BD%B3%E5%AE%9D&btnG=

[4] Wang Jingjing, Zhang Wu, Liu Lianzhong, Huang Shuai, 2014. A Review of Crop Pest and Disease Image Recognition Technology. https://scholar.google.com/scholar?hl=zh-CN&as_sdt=0%2C5&q=%E6%B1%AA%E4%BA%AC%E4%BA%AC&btnG=

[5] Rahman, C. R., Arko, P. S., Ali, M. E., Khan, M. A. I., Apon, S. H., Nowrin, F., & Wasif, A. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosystems Engineering*, *194*, 112-120.

[6] Shoaib M,Sadeghi-Niaraki A,Ali F, 2025. Leveraging deep learning for plant disease and pest detection: a comprehensive review and future directions. https://scholar.google.com/scholar?hl=zh-CN&as_sdt=0%2C5&q=.+Shoaib+M%E3%80%81Sadeghi-Niaraki+&btnG=

[7] Llorca, C., Yares, M. E., & Maderazo, C. 2018,. Image-based pest and disease recognition of tomato plants using a convolutional neural network. https://scholar.google.com/scholar?hl=zh-CN&as_sdt=0%2C5&q=Tomato+pest+and+disease+image+recognition&btnG=

[8] Prechelt, L. 1998. Automatic early stopping using cross validation: quantifying the criteria.https://scholar.google.com/scholar?hl=zh-CN&as_sdt=0%2C5&q=early+stopping+neural+network&btnG=&oq=early+stopping

