© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717386

Comparative Analysis of Deep Transfer Learning Models for Tomato Disease Classification

ARYAN BAKSHI[1], ABHISHEK KUMAR GUPTA[2], AYUSH VERMA[3], SUNITA SHARMA[4]
[1,2,3,4]*Dept. of Data Science Galgotia College of Engineering and Technology Knowledge Park II, India*

Abstract- ***One of the most challenging problems are tomato diseases which have been the cause of economic problems that have an effect on the production of agricultural products and food security worldwide. To stop the enormous loss of production and maintain the environmental sustainability, it is very important to locate the diseases at an early stage and be accurate. Six deep transfer learning architectures—VGG16, ResNet50, InceptionV3, DenseNet121, EfficientNetB0, and MobileNetV3- Large—have been studied very closely in this paper to classify tomato leaf diseases using the recently created Tomato-Village dataset. The dataset is very suitable for deployment at the field level as it contains the real photos of both healthy and diseased tomato leaves which were taken in a variety of lighting condition and background. To extend the models, they were updated with a substantial data augmentation and transfer learning to overcome the limitation of the dataset. Experiment results indicate that MobileNetV3-Large achieved the highest classification accuracy of 91.32 percent, thus it outperformed other heavier models with exceptional processing efficiency. In precision agriculture, a lightweight CNN architecture such as MobileNetV3-Large could be a practical and efficient way to automatically diagnose tomato disease in real time.***

Index Terms—Tomato disease detection, Deep learning, Trans- fer learning, MobileNetV3, Smart agriculture, Image classification, Convolutional Neural Networks (CNNs)

I. INTRODUCTION

Agriculture is the mainstay of global food security, but plant diseases continue to be a source of problems for both yield and quality. Compared to other crops, tomatoes are more susceptible to a wide range of diseases that can not only hamper economic stability but also drastically reduce productivity. The foremost measure to prevent such a drastic output loss and ensure agricultural sustainability is disease detection at the earliest stage and diagnosis that is accurate. The first line methods of disease detection are reliant on the skill of a human performing a manual check, which is at the same time, a labor-intensive, subjective process, and has several disadvantages in large farms.

Automated plant disease detection has been the focus of attention nowadays due to the application of computer vision and artificial intelligence in this area. Deep learning, in particular, has outperformed other methods in image classification-based Convolutional Neural Network (CNN) tasks. This research seeks to develop and test various deep learning models that use transfer learning to classify tomato leaf diseases.

This paper compares the tomato leaf disease detection performance of six state-of-the-art pre-trained CNN architectures, i.e., VGG16, ResNet50, InceptionV3, DenseNet121, EfficientNetB0, and MobileNetV3. The models have been fine- tuned in this study with transfer learning techniques on the Tomato Village dataset, which contains a wide variety of real images of tomato leaves with different backgrounds and lighting conditions.

The performance was measured through elaborate experiments using accuracy and loss metrics. MobileNetV3, among all the models tested, was the one that reached the highest classification accuracy of 91.32 percent while deeper networks had lower efficiency and more computational complexity. The paper proposes enhanced CNN architectures, in particular, lightweight models such as MobileNetV3, which can be a great resource for scalable and real-time disease detection in precision agriculture, and thus, have enormous potential.

II. METHODOLOGY

The proposed research involves the use of deep learn-ing and convolutional neural networks to

automatically cat- egorize diseases on tomato leaves. To compare the perfor- mance of the models, six pre-trained architectures—VGG16, ResNet50, InceptionV3, DenseNet121, EfficientNetB0, and MobileNetV3—are fine-tuned on the Tomato Village dataset through transfer learning.

III. DATASET DESCRIPTION

The dataset for this research is the Tomato-Village dataset, which is a publicly available set of data created for the detection of tomato diseases in the real world. Conventional datasets like PlantVillage are made up of pictures taken in a controlled lab environment, which means that models trained on these images have limited generalization when they are tested on natural field images. The Tomato-Village dataset has removed this obstacle by offering the images that were taken from open-field tomato farms in the Jodhpur and Jaipur districts of Rajasthan, India.

The dataset features the farm-to-table variations of the agrarian sector such as leaf occlusions, soil and background clutter, shadows, uneven illumination, and naturally occurring disease progression. It covers eight major categories of tomato leaf conditions, which are the most common ones in real farming environments:

- Early Blight
- Late Blight
- Leaf Miner
- Spotted Wilt Virus
- Magnesium Deficiency
- Nitrogen Deficiency
- Potassium Deficiency
- Healthy

To support different computer vision tasks, the Tomato- Village dataset is provided in three variants:

1) Variant-A: Multiclass Classification — Each picture has been given one label that reflects one of the eight categories of diseases or deficiencies. Such a version is appropriate for regular classification models that result in one predicted class for each input image.
2) Variant-B: Multilabel Classification — Images may show several symptoms at the same time due to over- lapping infections or nutrient deficiencies. This version allows the creation of multi-disease recognition systems that can deal with complicated situations in the field where tomato leaves have more than one condition.
3) Variant-C: Object Detection — Bounding box annota- tions are available for the diseased areas, which makes it possible to train object detection models like YOLO or Faster R-CNN. That helps to locate the disease in detail, which is a must for applications of precision agriculture, e.g., automatic disease mapping and treatment by the local region.

Different variants in total make the Tomato-Village dataset a complete standard that can be used to measure the per- formance of both classification and localization models. The dataset, thus, through these variants, enables the development of strong models and provides a less idealized evaluation of deep learning performance in real life by embodying real-world variation, complicated symptom overlapping, and exact diseased areas.

IV. MODEL ARCHITECTURE

This work compared six advanced convolutional neural network (CNN) architectures. Each network was first initial- ized with ImageNet pre-trained weights, and then transfer learning was employed to fine-tune them to the Tomato-Village dataset. In order to make a fair comparison, all models were trained with the same classification head. The architectures are outlined as follows:

- VGG16: A typical deep CNN with 16 weight layers is composed of a simple and uniform architectural design of convolutional layers stacked one after the other and followed by fully connected layers. For transfer learning, the last convolutional block was changed, and instead of the top classifier, a global average pooling layer, dropout for regularization, and a final dense layer with the number of classes were used.
- ResNet50: The ResNet50 is a 50-layer deep residual network which employs identity (skip) connections to mitigate the problem of vanishing gradients and enable the training of deeper models. We transformed the top- level modules of

ResNet50 to work as a feature extractor, replacing the original top with a custom pooling + dropout + dense classification head.

- InceptionV3: An innovative architectural design was developed which measures multi-scale features in a very efficient manner through the use of factorized convolu- tions and inception modules. The pretrained InceptionV3 backbone was changed by the addition of a shared classification head and the fine-tuning of the top inception modules for domain adaptation.
- DenseNet121: A densely connected network in which each layer obtains feature-maps from all preceding layers is aimed at feature reuse and efficient parameter usage. For tuning, we employed DenseNet121 with a fresh classification head and only higher layers were unfrozen selectively.
- EfficientNetB0: A modern compound-scaled network that manages to obtain very high accuracy with a small number of parameters by effectively balancing depth, width, and resolution. Transfer learning was employed to train EfficientNetB0 with the common top layers, and then its performance was evaluated using field images of tomatoes.
- MobileNetV3-Large: This small architecture, which is a good mobile/edge deployment candidate, makes use of inverted residuals, squeeze-and-excitation mod- ules, and platform-aware design. We have optimized MobileNetV3-Large in our experiments to achieve the best balance between accuracy and efficiency.

A. Common Classification Head

To offer consistency and equitable comparability across all pretrained topologies, an identical custom classification head was appended to each backbone network. This modular architecture, which maintains architectural uniformity among models, facilitates domain adaptation. The head consists of the components:

1) Input Layer: 224 × 224 × 3 RGB photos that have been standardized to the range [0, 1] are accepted.
2) Data Augmentation Block: To make the model ro- bust to real-world changes in background and light, it is trained with changes in brightness and contrast, translations by a small amount, magnification (up to 10%), rotations (up to ±20◦), and random flips in both horizontal and vertical directions. These modifications help the model to generalize better.
3) Backbone Preprocessing: To guarantee compatibility with pretrained Image Netweights, model-specific preprocessing routines (such as tf.keras.applications.*. preprocess_input) are used.
4) Feature Extraction Backbone: The convolutional base of the pretrained network (for example, VGG16, ResNet50, etc.) is where ImageNet weights are placed. The upper layers are modified to match the features of tomato diseases, the lower layers are still frozen, as the general features are retained.
5) Global Average Pooling (GAP): Minimizes overfitting and efficiently transforms the generated feature maps into compact feature vectors by reducing their spatial dimensions.
6) Batch Normalization: Enhances stability and speeds up convergence during training by normalizing feature distributions prior to the dense layers.
7) Dropout Layer: By randomly deactivating neurons during training, a dropout rate of 0.3–0.4 is used to lessen overfitting.
8) Fully Connected Layer: Non-linear feature combinations pertinent to disease categorization are learned via a dense layer comprising 256 ReLU-activated neurons.
9) Output Layer: Class probabilities matching to the number of tomato disease groups are provided by a final dense layer with softmax activation.

This shared head design ensures that differences in the classification layers are not the major reason for the performance of models to vary, but the representational power of the individual CNN backbones.

B. Training Configuration

To guarantee fair comparison and reproducibility, every model was trained using an identical experimental setup. The configuration used, unless otherwise indicated, was as follows:

- Input Size: Before being fed into the model, 224 × 224RGB photos were shrunk and normalized.
- Batch Size: For every experiment, a batch of

- Loss Function: sparse categorical crossentropy, appro- priate for integer-encoded labels in multi-class classifica- tion.
- Optimizer: Adam optimizer with an initial learning rate of 1 × 10−4 was used during the feature extraction stage,the learning rate was then gradually decreased to 1×10−5 (or less) during fine-tuning to stabilize convergence.
- Techniques for Regularization: In order to steer clear of overfitting and keep the best model weights, the follow-ing were employed: ReduceLROnPlateau (factor = 0.5, patience = 3, minimum LR = 1 × 10−6), EarlyStopping (monitoring validation accuracy with a patience of 5epochs), and ModelCheckpoint.
- Augmenting Data: Keras Sequential layers—random horizontal and vertical flips, rotations (±25◦), zoom (20%), brightness and contrast variation (±30%), and translation (20%)—were combined to produce a strong augmentation pipeline. By generating artificial field situ- ations, this gave the model more power to generalize.
- Hardware: Google Colab, which has an NVIDIA T4 GPU and 16 GB RAM, was used for all training trials.

This consistent setup guarantees that reported performance dis- parities between architectures are mainly due to their intrinsic representational capabilities rather than variances in training regimens or hyperparameter choices.

C. Training and Evaluation

Each model was implemented using TensorFlow and Keras deep learning frameworks. The training was per- formed for a maximum of 40 epochs with a batch size of 32 using the Adam optimizer and a learning rate of 1 × 10−4. The loss function utilized was sparse_categorical_crossentropy which is in line with integer-encoded class labels.

Disease classes were evenly distributed by first splitting the Tomato-Village dataset into three parts: training, validation, and testing. Approximately 70Model performance was judged on the basis of key metrics what came under the evaluation layer like accuracy, precision, recall, and F1-score. Along with that, for every model, there was a confounding matrix, which was used to investigate the misclassification sources and the performance of the classes. All the experiments were performed in a Google Colab environment with 16 GB of RAM and an NVIDIA T4 GPU. Such a configuration made the training process efficient and allowed for a fair comparison of the lightweight and heavy-weight architectures in the same conditions.

V. RESULTS AND DISCUSSION

The six transfer learning models used for tomato leaf disease classification. Their results are presented and discussed in this section. To make sure the comparison is fair, the same experimental setup was used to train and test all the models. Performance is primarily reported in terms of test accuracy and test loss; detailed per-class metrics (precision, recall, and F1 score) can be found in the Appendix.

A. Model Performance Visualization

Figures 1–12 provide the visual representations of the training and validation accuracy and loss that were performed for each of the models: VGG16, ResNet50, InceptionV3, DenseNet121, EfficientNetB0, and MobileNetV3-Large. These diagrams demonstrate changes in convergence and the impact of augmentation and regularization on different architectures.

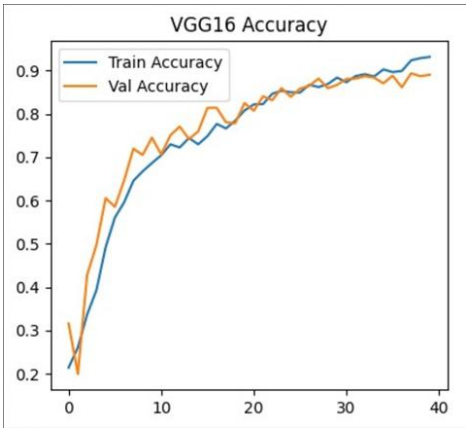

Fig. 1. Training and validation accuracy for VGG16.

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717386

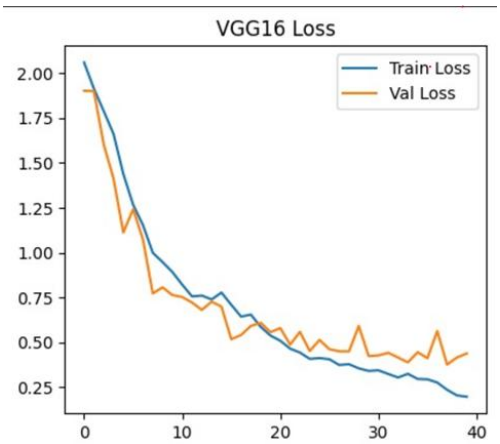

Fig. 2. Training and validation loss for VGG16.

B. Quantitative Performance Comparison

Table I summarizes the final test accuracy and test loss for all evaluated models.

C. Discussion

The experimental results illustrate a good trade-off between the representational power of the model and the computational efficiency, as evidenced by the fact that the MobileNetV3- Large model achieved the highest test accuracy (91.32%) ac- companied by the lowest test loss that was observed among all

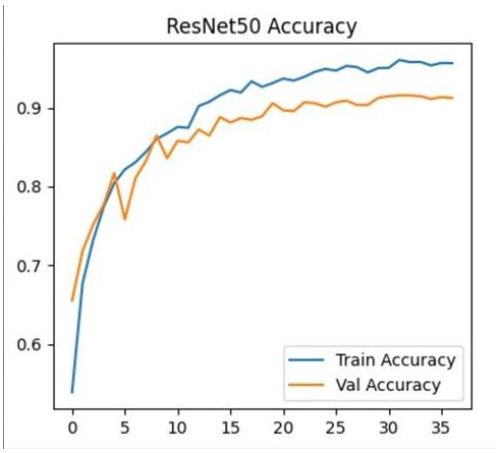

Fig. 3. Training and validation accuracy for ResNet50.

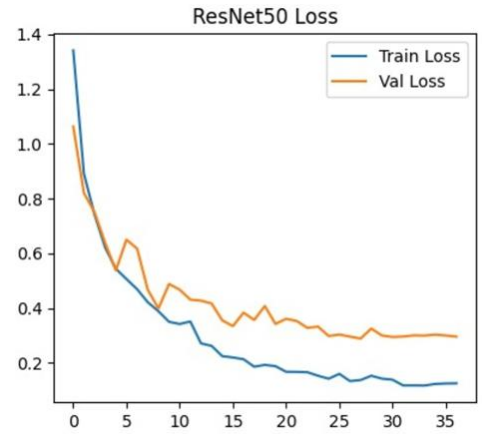

Fig. 4. Training and validation loss for ResNet50.

TABLE I TEST PERFORMANCE COMPARISON OF EVALUATED MODELS

Model	Test Accuracy (%)	Test Loss
VGG16	89.59	0.3330
ResNet50	89.80	0.2854
InceptionV3	91.11	0.2908
DenseNet121	91.11	0.2883
EfficientNetB0	85.90	0.3576
MobileNetV3-Large	91.32	0.2423

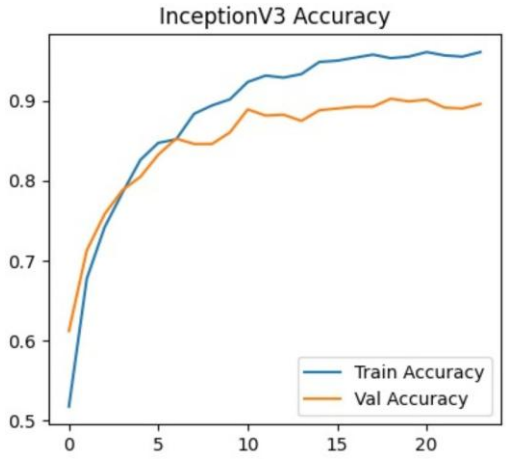

Fig. 5. Training and validation accuracy for InceptionV3.

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717386

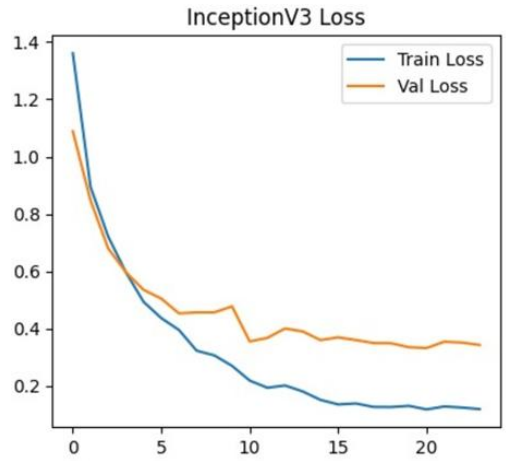

Fig. 6. Training and validation loss for InceptionV3.

models tested. In a nutshell, InceptionV3 and DenseNet121 have achieved very close accuracies (91.11%) that demonstrate the capacity of both architectures to capture the subtle texture and color changes in tomato leaf images taken in the field.

The performances of VGG16 and ResNet50 were both at par levels (89.6–89.8%) with VGG16 however requiring more training time and parameters for its accuracy. EfficientNetB0 on this dataset has done less than what was expected (85.90%); the explanation could be features that are specific to the dataset or simply that the model needs more fine-tuning (e.g., longer training, different unfreezing, or changed augmentation).

When it comes to deployment scenarios (mobile or edge devices) with scarce computing resources, light and efficient architectures such as MobileNetV3 can provide a decent compromise. The similar performance of several models suggests that further improvements might be achieved by using class-aware loss functions, targeted augmentation for underrepresented classes, or ensembling. A detailed per-class analysis (accuracy, recall, F1-score) and confusion matrices are pro- vided in the Appendix to help locate certain misclassification patterns (e.g., vitamin deficiency classes that share visual similarities).

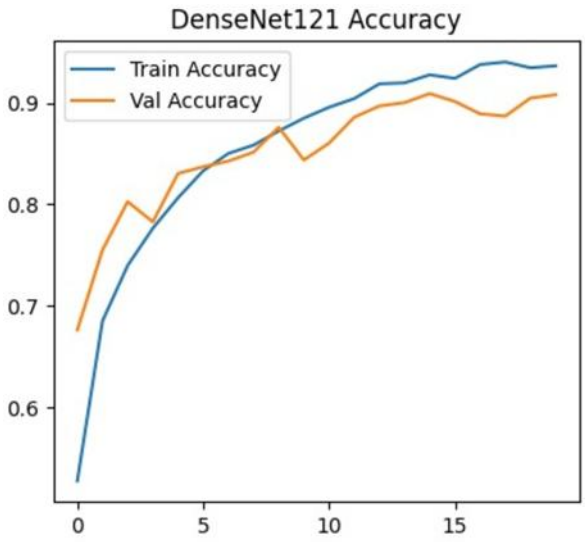

Fig. 7. Training and validation accuracy for DenseNet121.

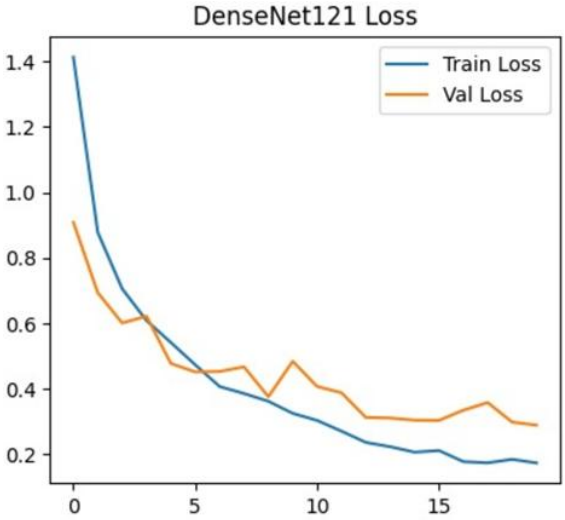

Fig. 8. Training and validation loss for DenseNet121.

ACKNOWLEDGMENT

The authors are indebted to their mentor, Ms. Sunita Sharma, for her direction and indefatigable support throughout this research. They also extend their gratitude to the Data Science Department of Galgotias College of Engineering and Technology for facilitating the resources and express their sincere thanks to the creators of Tomato-Village dataset for sharing their real-world images with the public.

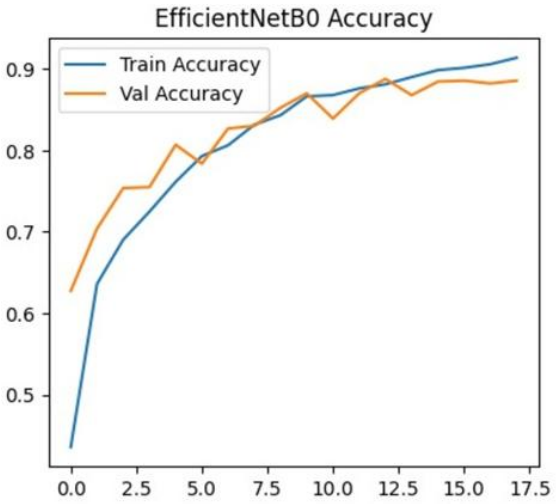

Fig. 9. Training and validation accuracy for EfficientNetB0.

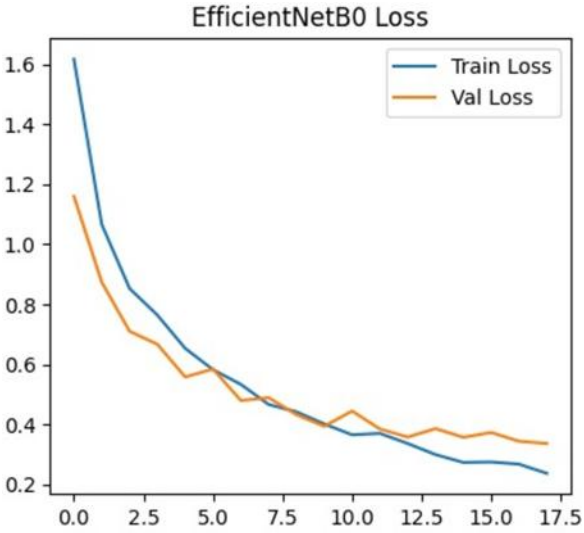

Fig. 10. Training and validation loss for EfficientNetB0.

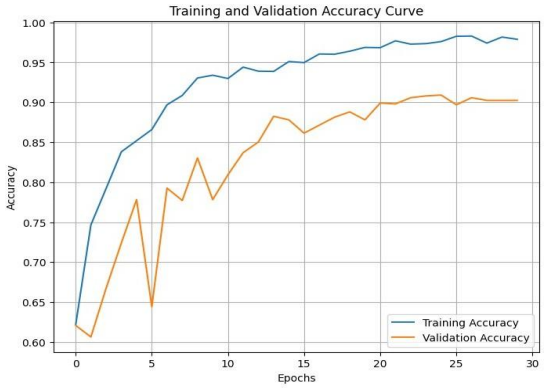

Fig. 11. Training and validation accuracy for MobileNetV3-Large.

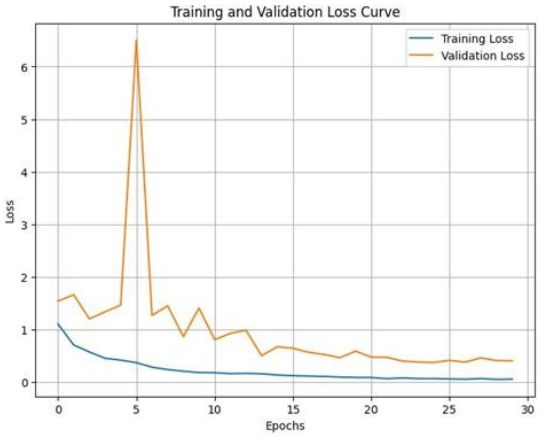

Fig. 12. Training and validation loss for MobileNetV3-Large.

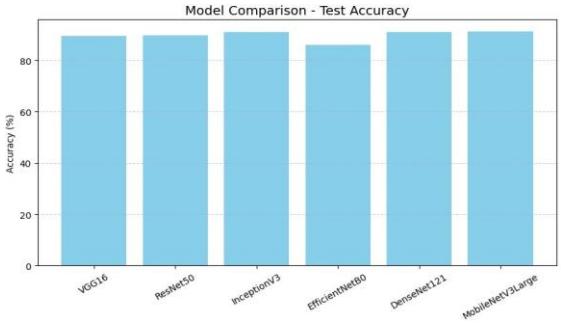

Fig. 13. Comparison of final test accuracies across all models.

REFERENCES

[1] M. Gehlot, R. K. Saxena, and G. C. Gandhi, "Tomato-Village: A dataset for end-to-end tomato disease detection in a real-world environment," Multimedia Systems, 2023. doi: 10.1007/s00530-023-01158-y.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[4] C. Szegedy et al., "Rethinking the inception architecture for computer vision," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826.

[5] G. Huang, Z. Liu, L. van der Maaten, and K. Weinberger, "Densely connected convolutional

networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700–4708.

[6] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolu- tional neural networks," in Proc. Int. Conf. Machine Learning (ICML), 2019.

[7] A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2019, pp. 1314–1324.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans- actions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.

[9] D. P. Hughes and M. Salathe´, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv:1511.08060, 2015.