Ginger Disease Detection Using a Computer Vision Pre-trained Model

Chapter · January 2023

DOI: 10.1007/978-3-031-40688-1_19

Olga Kolesnikova
Instituto Politécnico Nacional

Mesay Gemeda Yigezu
Interino del Centro de Investigación en Computación Instituto Politécnico Nacional

Atnafu Lambebo Tonja
University of Colorado Colorado Springs

Ginger Disease Detection Using a Computer Vision Pre-trained Model

Olga Kolesnikova, Mesay Gemeda Yigezu, Atnafu Lambebo Tonja, Michael Meles Woldeyohannis, Grigori Sidorov, and Alexander Gelbukh

Abstract Ethiopia is one of the African countries with a high potential for the creation of several crop varieties utilized in traditional medicine and daily life. Ginger is one of the plants that is afflicted by illness. The detection of disease requires specific attention from professionals, which is not achievable for mass production. On the other hand, cutting-edge technology can be used to circumvent the issue by applying image processing to a crop of ginger that is grown on a large scale. To that end, early ginger disease detection from the leaf is presented in stages after collecting 7,014 ginger photos with the assistance of domain experts from various farms using a transfer learning approach. The obtained data were subjected to various picture pre-processing techniques in order to construct and develop a model capable of detecting and dealing with a variety of circumstances. The study conducted two experiments: one using a pre-trained model for feature extraction, which has an accuracy of 91%, and the other fine-tuning a pre-trained model, which has superior performance than a pre-trained model for feature extraction, which has an accuracy of 97%. The experimental results show that the suggested technique is effective for detecting ginger diseases, particularly bacterial wilt.

O. Kolesnikova (✉) · M. G. Yigezu · A. L. Tonja · G. Sidorov · A. Gelbukh
Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional, Mexico City, Mexico
e-mail: kolesolga@gmail.com

M. G. Yigezu
e-mail: mgemedak2022@cic.ipn.mx

A. L. Tonja
e-mail: alambedot2022@cic.ipn.mx

G. Sidorov
e-mail: sidorov@cic.ipn.mx

A. Gelbukh
e-mail: gelbukh@cic.ipn.mx

M. M. Woldeyohannis
Addis Ababa University, Addis Ababa, Ethiopia
e-mail: michael.melese@aau.edu.et

G. Rivera et al. (eds.), *Innovations in Machine and Deep Learning*,
Studies in Big Data 134, https://doi.org/10.1007/978-3-031-40688-1_19

Keywords Transfer learning · Pre-trained model · Image detection · MobileNet · Ginger disease · Computer vision

1 Introduction

Agriculture is one of the sectors that drive the country's economy, but it is still done using conventional methods, which causes a variety of structural issues, including quality and quantity [5]. Crop productivity is greatly influenced by disease, which also has an impact on product quality. Ethiopian farmers grow a variety of crops and cereal items for export. Among these, the ginger crop generates a significant quantity of foreign cash for the country. Ginger production in Ethiopia is hampered by a lack of high-yielding varieties, a lack of innovative technology, and a limited role for the private sector in spice manufacturing [12]. Most fruit tree diseases and pests are identified on-site by agricultural and forestry professionals or by farmers using their knowledge. Not only is this procedure subjective, but it is also time-consuming, difficult, and inefficient. Farmers in Ethiopia, for example, do not have adequate facilities for contacting specialists because of the high cost and time required to obtain a consultant, and farmers with less experience may make mistakes and misuse drugs during the identification procedure [4]. This also results in environmental pollution, producing in avoidable economic losses. In order to address these concerns, image-processing methods for plant disease recognition have become an active area of research. Deep learning is one of the computer vision techniques used to detect and categorize numerous different things related to visual symptoms, like plant diseases [3]. The introduction of transfer learning has accelerated the rapid growth of computer vision and hence image classification. Simply put, transfer learning allows us to employ a previously trained model on a large data set for our own tasks. As a result, the cost of training new deep learning models is reduced, and because the data sets have been validated, we may be confident in their quality. There are various pre-trained models for image classification, among those such as[1] AlexNet which have 60 million parameters,[2] Visual Graphics Group (VGG) 138 million parameters, and[3] GoogleNet has 4 million parameters. It is generally accepted that deep learning models contain a very large number of parameters; consequently, these models require a significant amount of computational power in order to be trained from scratch, and they need a very large amount of data, and those are used mostly to identify the object, to detect disease. Previous researchers have proven that AI-based detection and classification of crop diseases were effective in maximizing product production and that previously trained Convolutional Neural Networks (CNNs) for different tasks on big benchmarks can be successfully used for age range categorization [9, 13]. As a result, we need computerized disease detection in a short period of

[1] https://www.doc.ic.ac.uk/~bkainz/teaching/DL/notes/AlexNet.pdf.

[2] https://www.jcancer.org/v10p4876.pdf.

[3] https://arxiv.org/pdf/1409.4842.pdf.

time by looking at plant symptoms in an easier and less expensive way to enhance the agricultural fields and the country's economy by enhancing the productivity and quality of ginger crops: a plant that shows symptoms on its leaves. Therefore, there is a need for a more accurate and useful model right now [10].

In this study, we realized two main activities: we released the ginger leaf disease data set, which was collected by the previous research [15] and made it available (https://github.com/QoMH/Gingerdiseasedetection) to researchers for further study. The second contribution is that we analyzed how we could improve the performance by using the pre-trained model as well as by fine-tuning it.

Generally, this chapter has eight sections. In Sect. 1, we discussed the overview of the study. In Sect. 2, we have discussed the related work that helped us do this study. Section 3 tells us how to prepare the data set for this study. Section 4 includes the pre-trained model and its timeline. In Sect. 5, we discussed in detail the workflow of the proposed model. Section 7 tells us about the experiment and performance of the models. The last Sect. 8, summarizes the entire activities of the study as a conclusion.

2 Related Work

Gemeda et al. [15] utilized a CNN algorithm to classify and detect ginger diseases. Their experimental results show that the proposed technique is effective for identifying ginger diseases, particularly bacterial wilt, with a test accuracy of 95.2%.

Selvaraj et al. [11] developed six different models from 18 different classes and used images collected from various parts of the banana plant, a technique that is based on transfer learning, ResNet50 and InceptionV2 pre-trained models which performed better than MobileNetV1 and obtained an accuracy of more than 90% in most of the models on the test set.

A total of 2,756 images were gathered to look for cassava disease by Ramcharan et al. [9] to train and diagnose the disease type and employ a deep convolutional neural network technique using transfer learning. As a result, a 93% total accuracy was attained.

Ashebir et al. [1] used a CNN to identify bacterial wilt, while a 4-way Softmax was used to classify bacterial wilt, and about 1600 photos were utilized to evaluate the model's performance, with 70% used for training, 15% for validation, and 15% for testing and achieved accuracy of 90.53%.

Oo et al. [8] used K-means clustering for segmentation and Support Vector Machine (SVM) for classification to classify cotton diseases. To do this, 560 images from four classes were captured: Bacterial Blight, Cercospora Leaf Spot, Powdery Mildew, and Rust Leaf Disease. The results reveal that the suggested system can successfully detect and classify plant diseases with an accuracy of 98.22% using an SVM classifier.

Mude et al. [7] demonstrated the use of image processing and an ANN to solve the problem of cotton plant disease identification. The effectiveness of cotton plant disease detection was achieved utilizing a variety of strategies, including the use of

Artificial Neural Network (ANN) as a classifier for testing in MATLAB to detect the types of disease on cotton leaves. The planned work had an accuracy of roughly 84%.

Wiesner et al. [14] acquired a large amount of data (18,000) for one plant disease from many platforms to aid the system in real-time observation. All of the image data sets, however, were obtained from a single source utilizing drones outfitted with a CNN model. As a result of this, it has come to light that a great number of strategies for identifying and locating plant diseases through the utilization of digital image processing have been developed and put into practice. The majority of the researchers sourced their data sets from the internet, specifically from databases like Plant Village that are open to the public. There are a lot of laborious pre-processing stages, such as handcrafted feature extraction, color histogram, texture features, and shape features; most importantly, the methods used by previously conducted research works are not state of the art, i.e., most of the studies in the literature of computer vision are not up to date. Using publicly available data sets is recommended, but most of the previously conducted research images are captured under controlled environments like in laboratory setups. According to the findings of a diverse group of researchers, computer vision is increasingly utilized, in particular for activities such as the detection and identification of diseases, and it has demonstrated fascinating results, particularly in the field of agriculture. Moreover, deep learning algorithms like CNN and ANN are most commonly used, but in order to detect ginger disease with better performance, we proposed a pre-trained model which is trained on a vast number of benchmark data sets.

3 Data Preparation

To train the model, we used data that was collected from previous work [15]. It was assisted by domain experts such as farmers, extension agents, and agricultural researchers. As a result, more than 7,014 photos were obtained from the farm. Among the 7,014 data-set photos, 2,734 are labeled as healthy, while the remaining 4,280 are labeled as bacterial wilt. Figure 1 exhibits leaf photos acquired for healthy and infected by bacterial wilt ginger with a 224×224 pixel size.

4 Pre-trained Model Description

A pre-trained model is a network that has already been saved and trained on a large dataset, typically on a large-scale image classification task. This type of model is referred to as "pre-trained." You have the option of utilizing the pre-trained and trained model in its current state, or you can utilize transfer learning in order to customize it to a particular job [2].

These open-source models have considerably aided the Deep Learning community. Furthermore, pre-trained models are a crucial component in the rapid advance-

(a) Healthy ginger leaf (b) Bacterial wilt ginger leaf

Fig. 1 Healthy ginger leaf and bacterial wilt ginger leaf

ment of computer vision research.[4] Figure 2 shows the timeline of pre-trained models. We observed various types of pre-trained models, as mentioned below.

VGG-16—which was made by the Visual Graphics Group at the University of Oxford, is one of the most popular pre-trained image categorization models. The model used numerous filters and is sequential in nature. Small 3×3 filters are employed at each stage to reduce the total number of parameters. The relu activation function is used to activate the hidden layers. Even yet, there are 138 billion parameters, which makes this model slower and larger to train than others.

Inception—has been a key part of making pre-trained image classification algorithms that are very popular. Having only 7 million parameters, it was far less complex than the then-dominant models like VGG and AlexNet. Its lower mistake rate makes it clear why it was a breakthrough model and conducted Max pooling the input with various filter sizes of convolutions and concatenating the result for the next inception. When the 1 by 1 convolution process is used, the parameters are dramatically reduced.

ResNet50—is not the first model from the ResNet family, like Inception V3. The original model, known as the ResNet, was an important development in the computer vision (CV) field in 2015. This model was designed to avoid poor accuracy as the model becomes deeper. It intended to address this problem as well.

MobileNet—is a type of CNN that was recently pre-trained and was created for embedded and mobile vision applications. They are built using depth-wise separable convolutions, which are lightweight deep neural networks that can have minimal latency for embedded and mobile devices. We used models with varying degrees of classification accuracy; therefore, from the pre-trained models listed above, we've chosen the MobileNet model because it has previously been trained on a sizable amount of data.

With these pre-trained models, which have a variety of accuracy levels and scales, we can efficiently fulfill any computer vision task without incurring the cost of creating fresh deep-learning algorithms. There are 2.5 million frozen parameters in MobileNet, However, there are 1,250 trainable parameters in the dense layer. These

[4] https://pytorch.org/hub/pytorch_vision_resnet/.

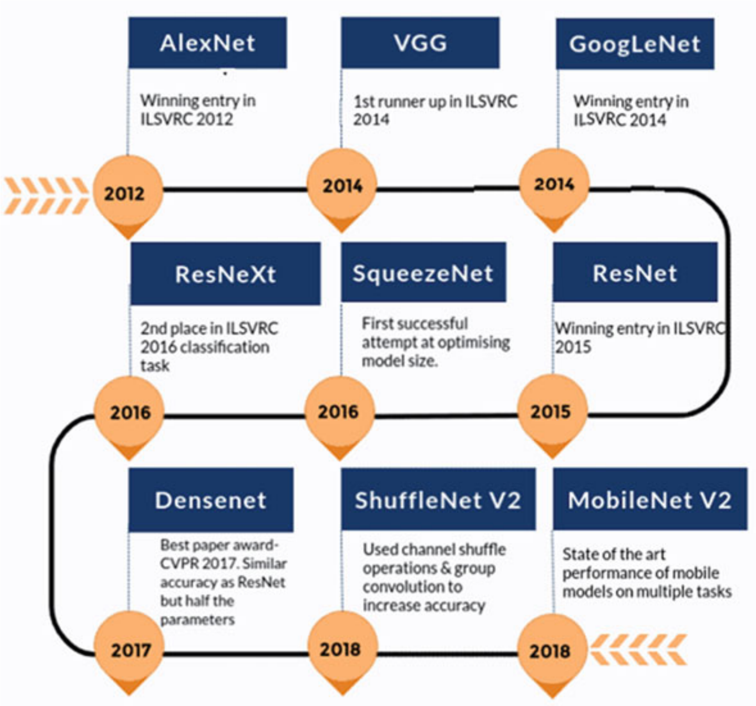

Fig. 2 The timeline of pre-trained models

are separated into two distinct categories: the weights and the biases. We used the ImageNet data set to train the pre-trained model. The majority of WordNet's[5] 80,000 synsets are filled with an average of 500–1,000 crisp, high-resolution images thanks to ImageNet. WordNet is an online lexical database intended for use with programs. Each set of synonyms for a lexicalized concept in English includes nouns, verbs, adjectives, and adverbs [6]. Therefore, the WordNet hierarchy (currently only the nouns) is used to organize the images in ImageNet, where each node of the hierarchy is represented by thousands of images. The outcome is tens of millions of annotated photographs arranged according to WordNet's semantic hierarchy. ImageNet's study of the current status of the database reveals 12 sub-trees with 5,247 synsets and 3.2 million images overall. We demonstrate that compared to the present image databases, ImageNet is significantly more accurate, diverse, and big in scale. We don't need to build complex models from scratch or collect a large amount of data to accomplish these studies. In addition to that, the key feature of MobileNet is the reason to select it, which is its use of depth-wise separable convolutions, which greatly reduces the number of parameters and computational requirements of the network while maintaining its performance. This makes it well-suited for use on mobile

[5] https://www.image-net.org/.

devices, where resources are limited. Additionally, MobileNet has been optimized for speed and can achieve real-time performance on modern smartphones and other devices. It has been used in a wide range of applications, including object detection, image segmentation, and facial recognition. Overall, MobileNet is a powerful and efficient architecture that is well-suited for a variety of computer vision tasks on mobile and embedded devices. Here we illustrate the timeline of the pre-trained model.[6]

5 Methodology

During the experiment, different detection scenarios were carried out to test the classification performance by applying the transfer learning method. The idea underlying transfer learning for ginger disease identification is that a model can successfully function as a generic model of the visual environment provided, trained on a sizable and general enough data set. By training a big model on a big data set, we used these learned feature maps instead of starting from scratch. Therefore, to accomplish the above objective, we used the following methods.

Figure 3 depicts the six activities in order to detect ginger disease by applying the transfer learning approach. Here we discuss them.

Prepare and understand the data: One of the most important elements for deep learning research is data. In this study, ginger leaf image data [15] is used as the main input to build a model by using a transfer learning approach.

Train the model: In order to detect a ginger disease, we train our model based on the given label, and we split our dataset into three classes which are training, testing, and validation with 70%, 20%, 10% of data, respectively.

Build an input pipeline: We employed Keras ImageDataGenerator in this instance. We need an alternative to arrays since, at some point, especially when working with photos, the data becomes too huge to fit in memory. By doing so, a Python generator is produced that gradually feeds data to the neural network without storing anything in memory.

Data augmentation: Deep learning takes a large amount of data unless there is sufficient data to apply data augmentation techniques, which may be used to increase the number of training data points in a dataset by producing more data from the existing training sample. Extending confined data sets to make use of the potential of large data can improve the performance of models. It facilitates organization, memorization of more complicated features from the data, foresees the over-fitting issue, and also provides a data space solution to the issue of limited data. The detection of leaf disease, as well as the collecting and labeling of a large number of disease photos, requires a huge amount of people, material resources, and financial resources. Some plant diseases have a shorter onset period and are more difficult to collect.

[6] https://learnopencv.com/pytorch-for-beginners-image-classification-using-pre-trained-models/.

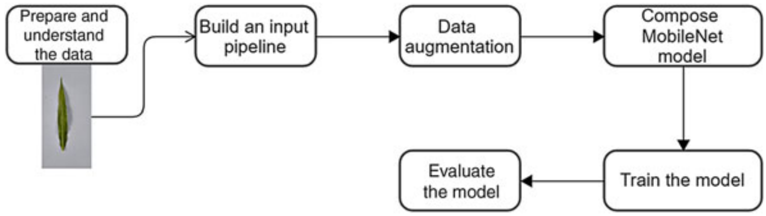

Fig. 3 Workflow to build our model

Table 1 Data augmentation parameters

Data augmentation parameters and it's range						
Rotation	width_shift	height_shift	Shear	Zoom	horizontal_flip	fill_mode
20	0.2	0.2	0.2	0.5	True	Nearest

Small sample sizes and dataset imbalances are the primary causes of poor recognition performance in deep learning.

As a result, expanding the amount of data for the deep learning model for leaf disease detection is required and was applied to the original image data in order to gain additional images, solve the problem of small data, and obtain a superior training and classifier model. As Table 1 depicts, after being augmented with various parameters, the training dataset contains 147,294 images with 30 epochs.

Compose the model: Because we are training a much larger model and intend to readjust the weights that have already been trained, it is essential that we make use of a slower learning rate at this stage. In that case, the model might quickly become overfit, and in this part of the process, two activities are carried out. After loading in the pre-trained base model (along with the pre-trained weights), the classification layers were stacked on top of the base model.

Evaluate the model: We need to know how the model generalizes for never-before-seen data after training it. This enables us to say if the model performs well when categorizing new data or whether it performs well just when classifying trained data and not when classifying new data. As a result, model evaluation is the process of determining the model's generalization correctness using unobserved data. It is not advised to utilize training data to evaluate a model since the model remembers all data samples fed during training, meaning that it accurately predicts all of the training data points but not for data that hasn't been seen during training. We used accuracy, precision, recall, and F1-score to evaluate the model.

6 Hyper-Parameter Setting

Hyper-parameter tuning is an important step in the process of building a deep-learning model for image detection. In Table 2, we mention some of the hyper-parameters that are used for tuning the proposed model.

Learning rate (LR): The learning rate determines how quickly the model learns from the data. If the learning rate is too high, the model may converge too quickly and not be able to generalize well. If the learning rate is too low, the model may take too long to converge or get stuck in a local minimum. In the proposed experiment, we used and considered it as best at 0.0001 LR.

Batch size (BS): The batch size determines how many images are processed in each iteration of training. Larger batch size can lead to faster training times, but it may also require more memory and lead to overfitting. Therefore, we conducted the experiment on 32 BS.

Number of epochs: The number of epochs determines how many times the model goes through the training data. Too few epochs may result in underfitting, while too many epochs may result in overfitting. In the proposed model, the optimum level is 30 epochs.

Dropout rate (DR): Dropout is a regularization technique that randomly drops out some units in the neural network during training. The dropout rate determines the probability that a unit will be dropped out. A higher dropout rate can help prevent overfitting, but too high a dropout rate can lead to underfitting, and the model achieved better results at 0.2 DR.

Activation function: The choice of activation function can affect the performance of the model. In this experiment, we used ReLU activation functions.

Optimizer: The choice of optimizer can affect the performance of the model, and in this scenario, the Adam optimizer performs better. In conclusion, we mention in tabular form what hyper-parameters we used and tuned to improve the performance of the model.

7 Experimental Result

In this study, we have conducted the experiment in two ways to customize a pre-trained model. At first, we used feature extraction: the representations learned by a previous network to extract valuable features from a new dataset. The pre-trained

Table 2 Hyper-parameter setting

Learning rate	Batch size	Epochs	Dropout rate	Activation function	Optimizer
0.0001	32	30	0.2	ReLu	Adam

model is simply added on top of a new classifier being trained from scratch to reuse feature maps already created for the dataset. The complete model was not (re)trained. Features that are often helpful for categorizing images are already present in the base convolutional network. However, the pre-trained model's final classification component is unique to the original classification task and subsequently specific to the set of classes on which the model was trained. We used the base model from the MobileNet V2 model developed at Google. This is pre-trained on the ImageNet dataset, a sizable dataset with 1.4M images and 1,000 classes. The study training dataset ImageNet has a wide range of categories, including jackfruit and syringe. This collection of data assisted us in categorizing healthy and infectious individuals from the particular dataset. In this experiment, the convolutional basis built in the previous stage is frozen and used as an extractor of features. In addition, we added a classifier on top of it, trained the top-level classifier, and (according to Fig. 4) achieved an accuracy of 91% with a macro avg score of 79% precision, 74% recall, 76% F1-scores. In the second experiment, which involves fine-tuning, some of the top layers of a frozen model base are defrosted, and the final layers of the base model and the newly-added classifier layers are jointly trained. This enabled us to "fine-tune" the basic model's higher-order feature representations to make them more pertinent to the task at hand. We merely trained a few layers on top of a MobileNetV2 base model for the feature extraction experiment. During training, there was no modification made to the weights of the pre-trained network. One strategy to improve performance even further in this scenario was to train (or "fine-tune") the weights of the top layers of the previously trained model in conjunction with the training of the classifier that we added. This is one technique that can be used. The weights had to be adjusted during the training process from generic feature maps to features particular to the dataset, and as Fig. 5 shows we achieved best performance than the first experiment accuracy with 97% and a micro score of 84% precision, 80% recall, 84% F1-scores performance.

In Table 3, we out the results of both the previous and the proposed models. Previously, a convolutional neural network was utilized, which achieved an accuracy of 95%. In the proposed model, a computer vision pre-trained model with the same data-set size was employed. The initial results were sub-optimal, but we were eventually able to enhance the performance by fine-tuning the pre-trained model, achieving an accuracy of 97%. In fine-tuning the model, we used the pre-trained weights as a starting point and then trained the model on our specific task, reducing the amount of time needed for training.

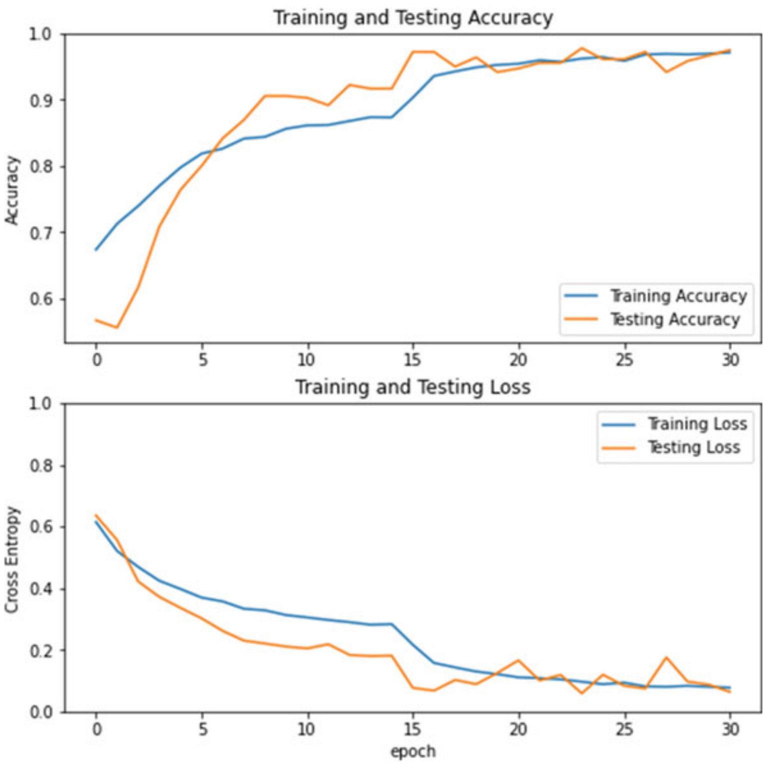

Fig. 4 Experiment result of the pre-trained model with feature extraction

Table 3 Result discussion on proposed and previous model

Method	Dataset	Data-augmentation	Metrics, %			
			Accuracy	Precision	Recall	F1-score
CNN-based	7014 images	✓	95	–	–	–
Pre-trained CV mode	7014 images	✓	91	79	74	76
Fine-tuning	7014 images	✓	97	84	80	84

This result shows us that fine-tuning a pre-trained model is a powerful technique that can lead to faster training times, improved performance, and better generalization, making it a valuable tool in computer vision applications.

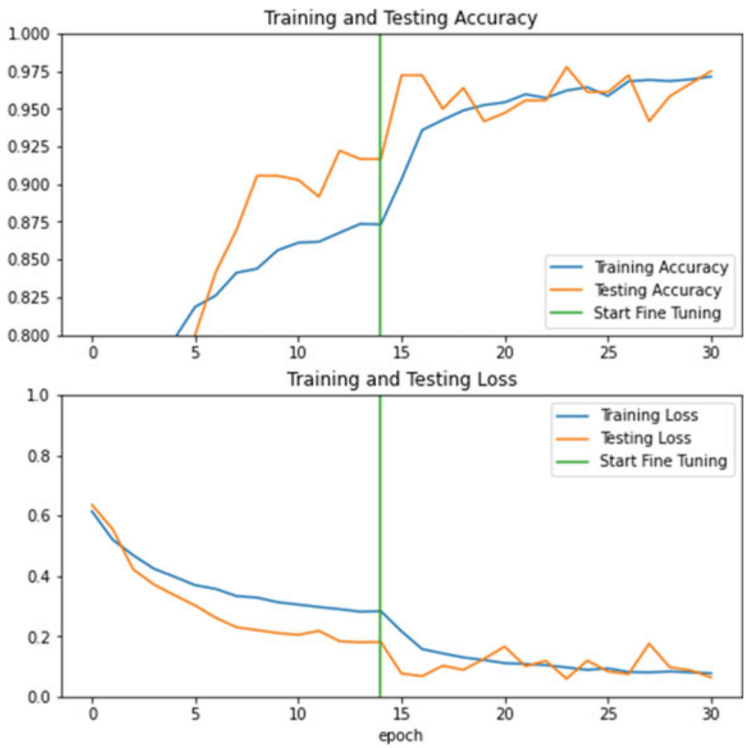

Fig. 5 Experiment result of fine-tuned model

8 Conclusion

In summary, the results of this experiment indicate that it is beneficial to use a pre-trained model for image identification tasks because it can help with a wide variety of issues.

The first reason is that utilizing a model that has already been pre-trained requires less time and effort to be invested in developing the architecture of the model. It is much more accurate to use a model that has already been pre-trained rather than to construct a CNN from the ground up. When attempting image detection, it makes perfect sense to get started with a model that has already been pre-trained because this is almost always the most effective way to proceed.

Using a model that has already been trained to extract features: when working with a limited data set, it is common practice to make use of the features that have been learned by a model that has been trained on a larger data set in the same domain. In order to achieve this goal, we first need to instantiate the pre-trained model and then layer it on top of a fully-connected classifier.

During training, the pre-trained model is “frozen,” and the only thing that gets changed is the weights of the classifiers. In this scenario, the convolutional base extracted all of the features that were associated with each image, and one can simply train a classifier that selects the image class based on the features that were extracted from the images. During the process of fine-tuning a pre-trained model: in order to raise the level of performance even further, fine-tune the top-level layers of the pre-trained models to the new dataset. In this particular scenario, we made adjustments to the weights so that our model could pick up high-level characteristics of the data set. This strategy is typically recommended whenever the training data set has a sizeable amount of data that is highly comparable to the initial data set on which the pre-trained model was trained.

Acknowledgements The work was done with partial support from the Mexican Government through grant A1S-47854 of CONACYT, Mexico, grants 20220852, 20220859, and 20221627 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico, and acknowledge the support of Microsoft through the Microsoft Latin America Ph.D. Award.

References

1. Ashebir, D., Tadesse, G.: Bwenet: detection and grading of bacterial wilt using deep convolutional neural network. Indian J. Sci. Technol. **15**(22), 1100–1111 (2022). https://doi.org/10.17485/IJST/v15i22.634
2. Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., et al.: Pre-trained models: past, present and future. AI Open **2**, 225–250 (2021). https://doi.org/10.1016/j.aiopen.2021.08.002
3. Kamilaris, A., Prenafeta-Boldú, F.X.: Deep learning in agriculture: a survey. Comput. Electron. Agric. **147**, 70–90 (2018). https://doi.org/10.1016/j.compag.2018.02.016
4. Li, L., Zhang, S., Wang, B.: Plant disease detection and classification by deep learning-a review. IEEE Access **9**, 56683–56698 (2021). https://doi.org/10.1109/ACCESS.2021.3069646
5. Mengistu, A.D., Alemayehu, D.M., Mengistu, S.G.: Ethiopian coffee plant diseases recognition based on imaging and machine learning techniques. Int. J. Database Theory Appl. **9**(4), 79–88 (2016). https://doi.org/10.14257/ijdta.2016.9.4.07
6. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995). https://doi.org/10.1145/219717.219748
7. Mude, S., Naik, D., Patil, A.: Leaf disease detection using image processing for pesticide spraying. Int. J. Adv. Eng. Res. Dev. **4**(4), 1–5 (2017)
8. Oo, Y.M., Htun, N.C.: Plant leaf disease detection and classification using image processing. Int. J. Res. Eng. **5**(9), 516–523 (2018). https://doi.org/10.21276/ijre.2018.5.9.4
9. Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., Hughes, D.P.: Deep learning for image-based cassava disease detection. Front. Plant Sci. **8**, 1852 (2017). https://doi.org/10.3389/fpls.2017.01852
10. Rivera, G., Porras, R., Florencia, R., Sánchez-Solís, J.P.: Lidar applications in precision agriculture for cultivating crops: a review of recent advances. Comput. Electron. Agric. **207**, 107737 (2023). https://doi.org/10.1016/j.compag.2023.107737

11. Selvaraj, M.G., Vergara, A., Ruiz, H., Safari, N., Elayabalan, S., Ocimati, W., Blomme, G.: Ai-powered banana diseases and pest detection. Plant Methods **15**, 1–11 (2019). https://doi.org/10.1186/s13007-019-0475-z
12. Shivakumar, N.: Biotechnology and crop improvement of ginger (zingiber officinale rosc.). In: Ginger Cultivation and its Antimicrobial and Pharmacological Potentials. IntechOpen (2019). https://doi.org/10.5772/intechopen.83688
13. Siricharoen, P., Scotney, B., Morrow, P., Parr, G.: A lightweight mobile system for crop disease diagnosis. In: Image Analysis and Recognition: 13th International Conference, ICIAR 2016, in Memory of Mohamed Kamel, Póvoa de Varzim, Portugal, July 13–15, 2016, Proceedings 13, pp. 783–791. Springer (2016). https://doi.org/10.1007/978-3-319-41501-7_87
14. Wiesner-Hanks, T., Stewart, E.L., Kaczmar, N., DeChant, C., Wu, H., Nelson, R.J., Lipson, H., Gore, M.A.: Image set for deep learning: field images of maize annotated with disease symptoms. BMC Res. Notes **11**(1), 1–3 (2018). https://doi.org/10.1186/s13104-018-3548-6
15. Yigezu, M.G., Woldeyohannis, M.M., Tonja, A.L.: Early ginger disease detection using deep learning approach. In: Advances of Science and Technology: 9th EAI International Conference, ICAST 2021, Hybrid Event, Bahir Dar, Ethiopia, August 27–29, 2021, Proceedings, Part I, pp. 480–488. Springer (2022). https://doi.org/10.1007/978-3-030-93709-6_32