

Research Article

Optimizing Pretrained Convolutional Neural Networks for Tomato Leaf Disease Detection

Iftikhar Ahmad,¹ Muhammad Hamid,¹ Suhail Yousaf,¹ Syed Tanveer Shah,²
and Muhammad Ovais Ahmad ³

¹Department of Computer Science and Information Technology, University of Engineering and Technology, Peshawar, 25000, Pakistan

²Department of Horticulture, The University of Agriculture, Peshawar, 25000, Pakistan

³Department of Mathematics and Computer Science, Karlstad University, Karlstad, 65188, Sweden

Correspondence should be addressed to Muhammad Ovais Ahmad; ovais.ahmad@kau.se

Received 30 August 2020; Revised 10 September 2020; Accepted 13 September 2020; Published 23 September 2020

Academic Editor: M. Irfan Uddin

Copyright © 2020 Iftikhar Ahmad et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Vegetable and fruit plants facilitate around 7.5 billion people around the globe, playing a crucial role in sustaining life on the planet. The rapid increase in the use of chemicals such as fungicides and bactericides to curtail plant diseases is causing negative effects on the agro-ecosystem. The high scale prevalence of diseases in crops affects the production quantity and quality. Solving the problem of early identification/diagnosis of diseases by exploiting a quick and consistent reliable method will benefit the farmers. In this context, our research work focuses on classification and identification of tomato leaf diseases using convolutional neural network (CNN) techniques. We consider four CNN architectures, namely, VGG-16, VGG-19, ResNet, and Inception V3, and use feature extraction and parameter-tuning to identify and classify tomato leaf diseases. We test the underlying models on two datasets, a laboratory-based dataset and self-collected data from the field. We observe that all architectures perform better on the laboratory-based dataset than on field-based data, with performance on various metrics showing variance in the range 10%–15%. Inception V3 is identified as the best performing algorithm on both datasets.

1. Introduction

No life is possible without plants; they provide food to all terrestrial living organisms and protect the ozone layer that is responsible for filtering harmful UV radiations of the sun. Although plants are essential for life, they face several challenges to grow as a variety of diseases hit them. The need for rapid recognition and diagnosis of diseases helps reduce the chances of damage to ecosystem. In the absence of systematic disease identification, the quality and quantity of products are affected. This further affects the economy of a country [1]. The United Nations Food and Agriculture Organization (FAO) proposes that agriculture production needs to increase by 70% by 2050 to overcome the world's food needs [2]. On the other hand, a rapid increase in the use of chemicals such as fungicides and bactericides to curtail

diseases has been negatively affecting the agro-ecosystem. Thus, we need rapid and effective early disease detection and classification techniques to identify the plant disease to sustain the agro-ecosystem. Among several fruit plants, tomato is a part of the daily diet. The need for early identification of tomato plant leaf diseases through technology-oriented approaches such as image processing and deep learning provides the opportunity for the development of such systems. Approximately 50% of the plant's production is damaged due to several diseases [1]. Farmers identify the disease by examining the plant and making judgements based on their past experiences [3]. This method does not provide accurate results as different farmers may have different experiences and the method lacks scientific rigour as well. There are chances that the farmers might miss classifying a disease and a wrong treatment may cause more

damage to the plant. Likewise, domain experts' visit to the field is costly. This necessitates the need of an automated image-based disease detection and classification mechanism that can replace the domain expert.

A number of researchers focused on the development of automated techniques for plant's disease identification using state-of-the-art techniques [4–9]. Durmuş et al. [4] used two different deep learning neural network architectures, namely, AlexNet and SqueezeNet for automated detection of disease in tomato leaves. The authors used images of the PlanetVillage dataset. The authors did not evaluate the performance of the neural network architectures based on standard performance metrics of F1, recall, precision, etc., instead they used only accuracy and inference time of the model. Tm et al. [5] proposed a variant of the convolutional neural net called LeNet for detection and identification of diseases in tomato leaves. The objective of the work was to identify a computationally robust technique for the underlying problem. The authors used images from PlanetVillage repository and reported an accuracy of 94%. Mohanty et al. [6] used AlexNet and GoogLeNet deep learning architectures to develop models for classification of tomato leaf diseases. The authors used a combination of learning algorithms and various splits of training and testing and reported an accuracy of 99.35% using the PlanetVillage dataset. For other contributions, the reader is referred to [7–9]. A common problem observed in the literature is the choice of dataset. The majority of the techniques proposed used controlled datasets that contain images obtained in perfect conditions in a controlled environment. However, in the real world, it is not possible to obtain high quality and high-resolution images of higher quality for possible detection and classification of tomato leaf diseases.

In this work, we evaluate convolutional neural network-based architectures, namely, VGG-16, VGG-19, ResNet, and Inception V3, for image-based detection and classification of tomato leaf diseases. Unlike previous studies, we used two types of datasets; firstly, we collected real field data from a tomato field in an uncontrolled environment and then used data augmentation technique to increase the number of instances; secondly, we also used laboratory data collected in a controlled environment. We report the results based on various performance metrics including accuracy, recall, precision, and F1-score. Thus, our evaluation methods are more robust and representative of a real-world scenario.

Rest of the paper is organized as follows. Section 2 presents the proposed approach including working of the CNN models, description of the datasets, and performance evaluation metrics. Section 3 presents the results based on the performance evaluation metrics for feature extraction as well as parameter-tuning. The results are discussed in the same section as well. Section 4 concludes the work and provides directions for future research.

2. Materials and Methods

2.1. Approach. We consider four well known convolutional neural networks (CNNs) architectures for identification and classification of diseases in the tomato plant leaf. These

architectures include VGG (VGG-16, VGG-19), residual neural network (ResNet), and Inception V3. Although the concept of deep neural network is not new, the availability of substantial amount of data and affordable computational power made it a reliable method in a variety of domains. CNNs are well known for image-based classification problems [10–12]. A distinguishing feature of CNN is the use of convolution layer, which omits the need of matrix multiplication. The various layers in a typical CNN include convolution, activation, pooling, and classification [13]. The purpose of the convolution layer is to reduce the dimension of input. The task of the activation layer is to apply nonlinear operators such as rectified linear unit (ReLU). The pooling layer is applied to further reduce the dimensions by the application of a statistical function such as MaxPool on neighboring values. After applications of these steps, a softmax function can be applied to classify the input into one of predetermined classes.

Although CNNs are shown to achieve excellent performance on image classification problems [14], two key problems are reported in the implementation and use of CNNs. Firstly, CNN involves a considerable number of parameters that are estimated in the training phase; secondly, the training phase by itself requires a large number of input images. Thus, designing and training CNNs from scratch is not considered an ideal solution. Instead, a rather unique and novel approach is used in which pretrained CNN models are considered and only the last few layers of the model are used in the training phase to estimate the parameters associated with those layers. Several such pretrained models are proposed in the literature, and we discuss the notable ones selected for our study.

2.1.1. VGG Net. The pretrained model was introduced by Visual Geometric Group (VGG) at the University of Oxford, and thus the name VGG [10]. The basic working principle of VGG Net is to use deeper layer with smaller filters. The input layer dimension of the VGG architecture is set for an image size of 244×244 . Preprocessing involves subtraction of the mean RGB value from each pixel of the input image. Preprocessing is followed by a stack of 5 convolutional layers, each of which is followed by a MaxPool layer, i.e., each set of convolutional layers is followed by a MaxPool layer. The final MaxPool layer precedes three fully connected (FC) layers. The first two FC layers have 64×64 (4096) channels, whereas the last FC layer has 1000 channels, which is followed by a softmax activation function. VGG network has multiple flavors, notably VGG-16 and VGG-19. VGG-16 and VGG-19 use the same architecture with different number of layers. VGG-16 uses 16 layers, whereas VGG-19 uses 19 layers. The differentiating factor is the number of convolution layers in the 3rd, 4th, and 5th layers of convolutional layers stacks.

2.1.2. ResNet. Residual network (ResNet) addressed the problem of training and overfitting in deep neural networks by introducing the concept of residual learning [15]. He et al. [15] highlighted that as the neural network architecture

becomes deeper, degradation occurs. Degradation is the phenomenon of increase in the training error as more layers are added to the architecture of a neural network. To solve the problem of degradation, the authors introduced residual block. Unlike VGG, which adds a stack of convolutional layers followed by a MaxPool layer, ResNet attempts to identify a residual mapping between the input to the convolutional layers and the output at the MaxPool layer, thus eliminating the computational cost of input being processed by the convolutional layer stack.

2.1.3. Inception Network. Szegedy et al. [16] extended the concept of network in network and proposed a modified CNN architecture to achieve improved performance by increasing the depth of the network and keeping the computational cost low. In contrast to VGG, Inception networks proved to be computationally efficient in terms of computing resource utilization as well as the number of parameters. However, the downside of the original inception network was its limited application adaptability in new use cases. Szegedy et al. [14] refined the original inception network model by introducing factorized convolutions with large filter size, factorization into smaller convolutions, and asymmetric convolutions. For details, the reader is referred to [14, 16].

As a preprocessing step, we used histogram equalization to increase the contrast. In addition, the input images are resized to match the requirements of the individual network (for instance, for VGG, the images are resized to 244×244).

2.2. Datasets. A categorized dataset is an essential part of a quantitative assessment. Although a standard categorized laboratory-based tomato leaf disease dataset has been developed for the assessment of the system [17], it is recorded in a controlled environment. There is no inclusive standard field-based database. In this research, we collected tomato leaf data from various fields in a natural uncontrolled environment. Afterward, the data are inspected by a domain expert to identify and classify the images into various categories.

The laboratory-based dataset contains 2364 images categorized into four types of different tomato leaf disease-infected high-resolution images. Each class contains 591 images. For system training and evaluation, we divided this dataset into three parts for training, validation, and testing, respectively. The specific ratio of each is 70%, 20%, and 10%, respectively. The detailed summary of the laboratory-based dataset is provided in Table 1.

It was a challenging task to collect datasets from different fields of tomato crop. The data were collected using a cell phone and in natural daylight conditions. The resultant datasets contain six types of infected tomato leaves. A total number of 317 images were collected with a cell phone camera. These were less in numbers for model training and evaluation. A higher number of images were needed to train a deep learning algorithm. Therefore, the data augmentation technique was used to increase the number of samples in the dataset. After the data augmentation, we obtained 15,216

TABLE 1: Distribution of the laboratory-based dataset for training, validation, and evaluation.

Type of tomato leaf disease	Training	Validation	Testing
Healthy	413	118	60
Late blight	413	118	60
Septoria leafspot	413	118	60
Yellow-curved	413	118	60

samples for the field-based dataset. The dataset was further divided into three parts for model training (70%), validation (20%), and testing (10%), respectively. Summary of distribution of the field-based dataset is provided in Table 2.

2.3. Performance Evaluation Metrics. We used accuracy, precision, recall, and F1-score as performance evaluation metrics. Note that the basic confusion matrix can be misleading; therefore, we used the aforementioned performance evaluation criteria.

2.3.1. Accuracy. Accuracy (A) represents the proportion of currently classified predictions and is calculated as follows:

$$A = \frac{TP + TN}{TP + TN + FP + FN}. \quad (1)$$

Note that TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively.

2.3.2. Precision. Precision (P) represents the proportion of positive outcomes that were actually correct and is calculated as follows:

$$P = \frac{TP}{TP + FP}. \quad (2)$$

2.3.3. Recall. Recall (R) measures the proportion of actual positives that were identified correctly and is calculated as follows:

$$R = \frac{TP}{TP + FN}. \quad (3)$$

2.3.4. F1-Score. F1-score is defined as the harmonic mean of precision and recall and calculated as follows:

$$F1 = 2 \frac{P \cdot R}{P + R}. \quad (4)$$

3. Results and Discussion

In this section, we report the results of the experiments performed in the study. The experiments performed in the study utilized several pretrained neural network architectures as feature extractors and fine-tuned the higher dimensional layers (last few layers) to learn features corresponding to the dataset. Pretrained neural network architectures utilized are VGG1-6, VGG-19, ResNet, and

TABLE 2: Distribution of the field-based dataset for training, validation, and evaluation.

Type of tomato leaf disease	Training	Validation	Testing
Healthy	2790	796	398
Late blight	3393	968	485
Septoria leafspot	403	115	58
Bacterial speck	336	96	48
Bacterial spot	739	211	106
Early blight	2991	854	427

Inception V3. A total of sixteen experiments were performed, eight on each dataset. In four experiments, pretrained neural network architectures are used as feature extractors, whereas in the remaining four experiments, hyperparameter tuning is used. For all experiments, we have used 10-fold cross-validation.

3.1. Results Using Feature Extraction. In this section, results are reported for both datasets using pretrained neural networks as feature extractors. The image classification process can be divided into two parts. Feature extraction is carried out by convolutional neural networks, and the classification is performed by fully connected layers with the ReLu activation function and softmax. Table 3 presents the results on both datasets using pretrained neural network architectures as feature extractors. Analysis of results concludes that Inception V3 outperformed all other pretrained models with the highest reported accuracy in both datasets. However, the laboratory dataset achieved better classification accuracy (93.40% using Inception V3). This may be due to the reason that the laboratory dataset is a standard balanced dataset curated by experts in the domain, whereas our field dataset was collected through a cell phone camera and is an imbalanced dataset.

3.2. Results Using Parameter-Tuning. In this section, results are reported for both datasets by fine-tuning the parameters of the pretrained neural network architectures. The high-dimensional layers of pretrained neural network architectures are trained to adjust the parameters according to our dataset. Low-dimensional layers’ features are kept the same for both datasets. The classification task is carried out using fully connected layers with the ReLu activation function and softmax is employed at the final layer. Table 4 presents the results on both datasets by fine-tuning the pretrained neural network architecture. Analysis of results concludes that the Inception V3 architecture outperformed all other pretrained models with the highest reported accuracy in both datasets. As expected, the accuracy is high on the laboratory-based dataset.

Table 5 summarizes recall, precision, and F1-score achieved by the four models on the two datasets.

In terms of recall score, as expected, all models performed better on a laboratory-based dataset than a field-

TABLE 3: Classification accuracy using feature extraction.

Model	Laboratory dataset	Field dataset
VGG-16	88.10	76.20
VGG-19	90.00	79.60
ResNet	91.70	81.70
Inception V3	93.40	85.00

TABLE 4: Classification accuracy using parameter-tuning.

Model	Laboratory dataset (%)	Field dataset (%)
VGG-16	98.50	84.10
VGG-19	98.30	86.30
ResNet	99.40	91.30
Inception V3	99.60	93.70

based dataset. The average performance difference in the recall score for the two datasets is 13.7%. Inception V3 is the best performing model achieving 0.996 and 0.906 recall score on the laboratory-based dataset and field-based dataset. The same performance trend is observed on precision and F1-score. In both instances, the score achieved on the laboratory-based dataset is superior to that achieved on the field-based dataset. In terms of precision, the difference between the score achieved on the laboratory-based data and field-based dataset is 17.9%. For F1-score, the difference is 15.8%. Figure 1 summarizes the average accuracy, recall, precision, and F1-score using the parameter-tuning technique on laboratory-based and field-based datasets.

Several interesting observations can be drawn from the reported performance metrics. For all models, fine-tuning the parameters of a pretrained neural network architecture achieved better classification accuracy as compared to using the neural network architecture with feature extraction only. This observation is typically common as using the feature extractor that is trained on a different dataset may not always capture the best set of discriminative features of the images under study (tomato leaf diseases in our case). As far as accuracies among models are concerned, Inception V3 outperformed all other pretrained models. This may be due to the reason that Inception V3 uses different kernel sizes for the effective recognition of variable-sized features. Instead of simply going deeper in terms of the number of layers, it goes wider. Multiple kernels of different sizes are implemented within the same layer. Expectedly, all models performed

TABLE 5: Recall, precision, and F1-score using parameter-tuning.

Model	Recall		Precision		F1-score	
	Laboratory	Field	Laboratory	Field	Laboratory	Field
VGG-16	0.991	0.792	0.99	0.758	0.99	0.776
VGG-19	0.991	0.847	0.989	0.811	0.989	0.828
ResNet	0.995	0.883	0.994	0.841	0.994	0.861
Inception V3	0.996	0.906	0.996	0.845	0.995	0.876

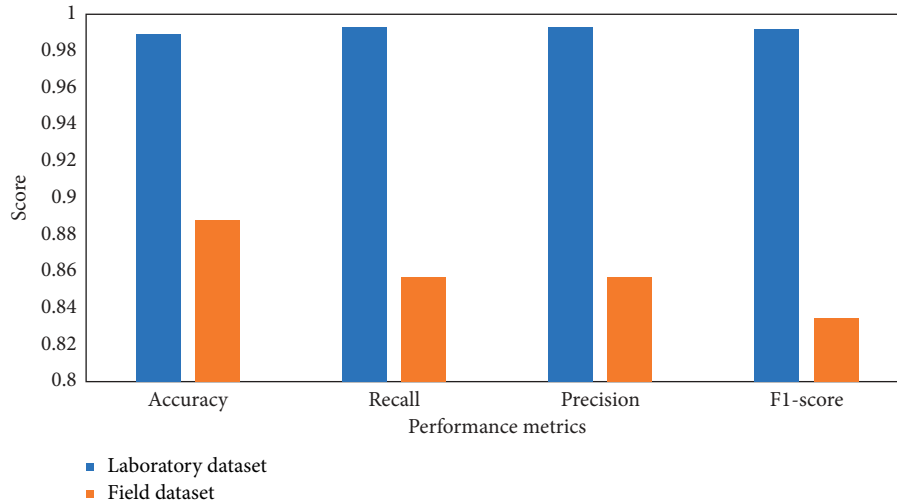


FIGURE 1: Summary of performance on laboratory- and field-based datasets.

better on the laboratory-based dataset than field-based dataset as the data are collected from the field in an uncontrolled manner.

4. Conclusions

In this work, we used different pretrained convolutional neural networks for automatic detection and classification of diseases in a tomato plant leaf. We considered four different models, namely, VGG-19, VGG-16, ResNet, and Inception V3, and evaluated their performance on two divergent datasets. The first dataset is a controlled dataset whose images are acquired in a laboratory; the second dataset is prepared by us by collecting data from the field in natural light with the help of a cell phone. Thus, the second data are representative of a real-world situation and were hence proved to be more challenging for various pretrained neural network models. We observed that parameter-tuning results in more accurate results than feature extraction. Likewise, the average performance on the laboratory-based dataset was 10%–15% superior in comparison to the field-based dataset. Inception V3 was the best performing model on both the datasets.

As these models do not perform well on the field-based dataset, therefore, a natural extension of our work will be to optimize these models for better performance on real-world field-based data.

Data Availability

Previously reported data (laboratory-based dataset) were used to support this study and are available at <https://github.com/PrajwalaTM/tomato-leaf-disease-detection>.

The field-based data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] C. A. Harvey, Z. I. Rakotobe, N. Rao et al., "Extreme vulnerability of smallholder farmers to agricultural risks and climate change in Madagascar," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, no. 1639, Article ID 20130089, 2014.
- [2] FAO, *2050: A Third More Mouths to Feed*, FAO, Rome, Italy, 2020, <http://www.fao.org/news/story/en/item/35571/icode/>.
- [3] F. Martinelli, R. Scalenghe, S. Davino et al., "Advanced methods of plant disease detection. a review," *Agronomy for Sustainable Development*, vol. 35, no. 1, pp. 1–25, 2015.
- [4] H. Durmuş, E. O. Güneş, and M. Kırç, "Disease detection on the leaves of the tomato plants by using deep learning," in *Proceedings of the 2017 6th International Conference on Agro-Geoinformatics*, pp. 1–5, Fairfax, VA, USA, August 2017.
- [5] P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato leaf disease detection using convolutional neural networks," in *Proceedings of the 2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 1–5, Noida, India, August 2018.
- [6] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.

- [7] C. Xie, Y. Shao, X. Li, and L. He, "Detection of early blight and late blight diseases on tomato leaves using hyperspectral imaging," *Scientific Reports*, vol. 5, Article ID 16564, 2015.
- [8] R. Karthik, M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, and R. Menaka, "Attention embedded residual CNN for disease detection in tomato leaves," *Applied Soft Computing*, vol. 86, Article ID 105933, 2020.
- [9] A. E. Hassaniien, T. Gaber, U. Mokhtar, and H. Hefny, "An improved moth flame optimization algorithm based on rough sets for tomato diseases detection," *Computers and Electronics in Agriculture*, vol. 136, pp. 86–96, 2017.
- [10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, May 2015.
- [11] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision*, Springer, Zurich, Switzerland, pp. 818–833, September 2014.
- [12] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: integrated recognition, localization and detection using convolutional networks," in *Proceedings of the ICLR*, Banff, Canada, April 2014.
- [13] A. E. Tio, "Face shape classification using inception v3," 2019, <https://arxiv.org/abs/1911.07916>.
- [14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and pattern Recognition*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [16] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.
- [17] T. M. Prajwala, Tomato-leaf-disease-detection, 2020, <https://github.com/PrajwalaTM/tomato-leaf-disease-detection>.