

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/375429246>

# Cotton Leaf Disease Identification Using Transfer Learning

Conference Paper · September 2023

DOI: 10.1109/CICT4SD59951.2023.10303323

---

CITATION

1

---

READS

42

2 authors:



[Asaduzzaman Herok](#)

Islamic University of Technology

1 PUBLICATION 1 CITATION

[SEE PROFILE](#)



[Sabbir Ahmed](#)

Islamic University of Technology

25 PUBLICATIONS 189 CITATIONS

[SEE PROFILE](#)

# Cotton Leaf Disease Identification using Transfer Learning

Asaduzzaman Herok

Department of Computer Science and Engineering  
Islamic University of Technology  
Gazipur, Bangladesh  
asaduzzaman34@iut-dhaka.edu

Sabbir Ahmed

Department of Computer Science and Engineering  
Islamic University of Technology  
Gazipur, Bangladesh  
sabbirahmed@iut-dhaka.edu

**Abstract**—Accurate identification and classification of plant diseases play a vital role in safeguarding global food production and ensuring the economic well-being of the stakeholders. Various solutions have been developed using Deep Learning (DL) based systems for different essential crops. However, the research landscape regarding intelligent solutions for automatically classifying cotton leaf diseases remains largely unexplored despite cotton being a significant commercial crop in many regions of the world. This article describes a method for detecting diseases in cotton leaves based on transfer learning. Unlike the previous works focusing on a limited number of classes and/or samples, we combined samples from different resources and curated a dataset of seven cotton leaf diseases and one healthy class. The dataset reflects the challenges of heterogeneous sources, varying backgrounds, illumination, inter-class similarity, etc., making it suitable for estimating performance in real-life scenarios. After that, we assessed the effectiveness of various pretrained Deep Convolution Neural Networks, including InceptionV3, ResNet152V2, VGG16, InceptionResNetV2, Xception, MobileNetV2, and DenseNet121, where the VGG16 model achieved the highest accuracy of 95.02%. Finally, we provided a thorough class-wise performance and error analysis to show the capability of the model under different scenarios.

**Keywords**—Leaf disease classification, Pretrained Models, VGG16, Cotton disease

## I. INTRODUCTION

In developing countries, infectious diseases pose a significant threat to crop production, resulting in substantial reductions in potential yield. Studies have shown that these reductions can range from an average of 40% to as high as 100% [1]. However, traditional monitoring systems face numerous limitations and rely heavily on expert support. One major drawback is that reliance on manual observation methods is prone to errors and consumes considerable time. Moreover, the labor-intensive nature of the process makes it costly for real-time monitoring across large agricultural areas. This limitation, in turn, hampers effective decision-making for farmers and other stakeholders in the agriculture sector.

The issue is further exacerbated by the limited agricultural knowledge of crop growers [2]. To address these issues and ensure global food security, there is a growing need for intelligent agricultural systems that can rapidly and accurately predict the occurrence of diseases. One potential strategy is applying computer vision methods to examine leaf images for the early diagnosis of plant diseases [3]. By leveraging

these technologies, farmers can promptly identify and respond to disease outbreaks, thereby minimizing crop losses and improving overall productivity.

The classification of plant diseases has been approached in various ways using traditional Machine Learning (ML) based methods. These techniques, however, frequently rely on manually choosing features, which may not transfer well to larger datasets [4]. In contrast, the outstanding generalization capabilities of Deep Learning (DL) algorithms have transformed agriculture by removing the necessity for manual feature engineering [5]. These DL-based approaches have achieved outstanding accuracy in automatically learning features from large datasets of real-world leaf images containing a wide range of varieties [6]. Particularly, Convolutional Neural Networks (CNNs) have become a highly efficient technique for various classification tasks, including identifying leaf diseases [7]. Additionally, contemporary CNN designs like ResNets, DenseNets, InceptionNets, and others have shown that they can understand complicated patterns, outperforming humans in various detection and categorization tasks [8], [9].

Transfer learning has remarkably reduced the need for intensive computational resources by using a model that has worked well to solve one problem as the basis for another. Through this procedure, models trained on extensive datasets can comprehend nearly all critical aspects of an image [10]. These pre-trained architectures can improve their performance noticeably in just a few training epochs by fine-tuning with domain-specific datasets [11]. As a result, this has been widely used to classify leaf diseases by using pre-trained architectures based on publicly available datasets, leading to a wide range of solutions [12]. However, the use of these smart techniques for classifying cotton leaf diseases is still largely unexplored despite the extensive production of cotton, a substantial economic crop in many parts of the world [13], [14]. More investigation and research are required to fully utilize the potential of transfer learning in this regard.

## II. METHODOLOGY

### A. Dataset

We have curated a dataset by combining the samples from different online resources licensed under an open database



Fig. 1: Samples Images from the Dataset

license(ODbL) and creative commons (cc) public domain license. This resulted in a dataset of 7 disease classes and 1 class containing samples of healthy leaves. The motivation behind combining the samples was to produce a dataset containing diversified samples of each disease. The total sample count was 6158, and the detailed distribution is presented in Table I. The Healthy class contains the highest number of samples, whereas the Bacterial Blight is the second highest, containing 1048 images. The remaining classes have around 600 samples.

Figure 1 shows one sample from each class of the dataset, showing that the samples are extremely challenging, unlike the laboratory environment samples in the PlantVillage dataset [15]. The varying backgrounds, illumination, orientations, degree of infections, etc., make the classification task extremely challenging. Moreover, as the samples are collected from heterogeneous open-source data sources, they contain high intra-class variability. This kind of variety was not present in the earlier works of cotton leaf disease classification.

### B. Transfer Learning with Pretrained models

Using a pre-trained model that has often been trained on a large-scale dataset as a starting point to solve a separate but related job is known as transfer learning. This technique has the capabilities of performing well with little training data, strong feature extraction, improved generalization, and faster convergence, making it beneficial in image classification.

Training Deep CNNs from scratch requires a massive amount of labeled training data. Transfer learning allows us to leverage pre-existing large-scale datasets, such as ImageNet,

TABLE I: Detailed Distribution of the Samples of the Dataset

Class Name	Sample Count	Train Split	Validation Split	Test Split
Healthy	1874	1311	374	189
Aphids	600	420	120	60
Army Worm	600	420	120	60
Bacterial Blight	1048	733	209	106
Curl Virus	418	292	83	43
Fussarium Wilt	418	292	83	43
Powdery Mildew	600	420	120	60
Target Spot	600	420	120	60
Total	6158	4308	1229	621

which contain millions of labeled images. Moreover, Deep CNNs are excellent feature extractors, capable of learning intricate patterns and hierarchical representations from images. This saves computational resources and reduces the need for training from scratch. Since Pre-trained models have already learned rich and generic feature representations from large datasets, these representations capture useful information about image features that can be applied to different tasks.

For each of the pre-trained models, the last classification layer is discarded, and the parameters of the existing layers are frozen as non-trainable parameters. A new dense layer with eight nodes (the number of classes) is added as the classification layer with the softmax activation function. To find the best-suited model for this task, we have experimented with InceptionV3, ResNet152V2, InceptionResNetV2, Xception, MobileNetV2, DenseNet121, and VGG16 architectures.

1) *InceptionV3*: Google unveiled InceptionV3 architecture, introducing the idea of Inception modules, which are structural units made up of concurrent convolutional branches with various kernel sizes. The network can learn multiple and multi-level representations because of these branches, which capture characteristics at various sizes and resolutions. When it was first released, InceptionV3 showed cutting-edge performance on the ImageNet dataset. It uses  $1 \times 1$ ,  $3 \times 3$ , and  $5 \times 5$  convolutions within each Inception module to create a compromise between model complexity and computational effectiveness.

2) *ResNet152V2*: ResNet152V2 is an extension of the ResNet architecture, which is based on the idea of residual learning. It has 152 layers, making it a very deep neural network. The architecture introduces residual connections, where shortcut connections bypass one or more convolutional layers to help alleviate the vanishing gradient problem. ResNet152V2 enables training deep networks that can effectively learn complex and abstract features. It has been successfully applied to various computer vision tasks and has achieved remarkable performance on large-scale datasets.

3) *InceptionResNetV2*: InceptionResNetV2 combines the strengths of both the Inception and ResNet architectures. It incorporates residual connections of ResNets within the Inception modules. Residual connections allow faster optimization and better gradient flow during training. InceptionResNetV2 achieves improved accuracy and representation power by fusing Inception modules with residual connections. The architecture includes a combination of  $1 \times 1$ ,  $3 \times 3$ , and  $5 \times 5$  convolutions within the modules to capture features at different scales.

4) *Xception*: Xception (Extreme Inception) is an architecture that builds upon the concept of depthwise separable convolutions. It replaces the standard convolutional layers of traditional architectures with a combination of depthwise separable convolutions. Depthwise separable convolutions split the spatial and cross-channel filtering into separate operations, significantly reducing computational complexity. By reducing the number of parameters and computations, Xception achieves a good balance between accuracy and computational efficiency. It has been particularly effective for resource-constrained environments such as mobile and embedded devices.

5) *MobileNetV2*: The MobileNetV2 was created primarily for mobile and embedded devices with constrained computational capabilities. It reduces the number of parameters and the computing cost by using depthwise separable convolutions, like Xception. Inverted residual blocks are included together with linear bottlenecks to enhance information flow and promote feature propagation. It balances model accuracy and size, making it appropriate for real-time applications on hardware with constrained memory and computing capacity.

6) *DenseNet121*: DenseNet121 is a densely connected CNN architecture known for its densely connected blocks. It introduces skip connections, also known as dense connections, between all layers within a block. Dense connections allow direct information flow between layers, facilitating feature reuse and enhancing gradient flow during training. By con-

necting each layer to every other layer in a feed-forward fashion, DenseNet121 promotes extensive information exchange and feature propagation. This architecture has shown strong performance in image classification tasks, even with fewer parameters than other models.

7) *VGG16*: VGG16 is renowned for its ease of use and performance in image categorization. The design has 16 layers, including three fully connected levels and 13 convolutional layers. Small  $3 \times 3$  filters with a stride of 1 and padding of 1 are used in the convolutional layers. Max-pooling is used after each pair of convolutional layers to minimize the spatial dimensions. Due to the deep and consistent structure of VGG16, it can learn intricate and abstract information. To induce non-linearity, rectified linear units (ReLU) are utilized as activation functions. To reduce overfitting, dropout regularization is used. VGG16 has a size of about 138 million trainable parameters, which makes it a huge model. VGG16 is still a good option for transfer learning, serving as a benchmark for comparison in image classification tasks.

### III. RESULTS AND DISCUSSION

#### A. Experimental Setup

The proposed pipeline underwent execution using NVIDIA Tesla T4 GPUs having 15GB of virtual memory. This training process took place within the Google Colaboratory environment, which offers a platform for collaborative coding and execution of machine learning tasks. To prepare the data for training, the original input images were resized to a specific dimension of  $224 \times 224 \times 3$ . The allocation of data was done in a 70 : 20 : 10 split, with 70% of the data used for training the model, 20% for validating the model's performance, and the remaining 10% for evaluating the final model on unseen data. Considering the limitations of the GPU's memory capacity, a batch size of 32 was chosen for the training process. This batch size determines the number of training samples processed at once, and in this case, it was set to fit within the available GPU memory. Each model was trained for 20 epochs. It used the Adam optimizer. In deep learning, the Adam optimizer is a prominent option and helps to adjust the model's weights during training. It used a learning rate of 0.001, determining the step size in weight updates.

#### B. Performance of Pretrained Models

The performance of the different pre-trained CNN models is represented in Table II. We have listed the training, validation, and testing accuracies for each model. Although VGG16 has the highest training accuracy, the other models also have a similar kind of training performance. However, it is essential to note that training accuracy alone does not provide a complete picture of the performance. It is crucial to evaluate the model on validation and test datasets to assess its generalization ability. High training accuracy does not guarantee good performance on unseen data. Overfitting can occur if the model memorizes the training data without generalizing well.

DenseNet121 has the highest Validation accuracy; however, the performance on the test split was less than VGG16. The



TABLE II: Performance of Pretrained Architectures

Model	Train Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
InceptionV3	99.55	92.01	88.42
InceptionResNetV2	99.44	93.17	90.34
Resnet152V2	99.77	95.11	92.28
Xception	99.30	94.79	92.59
MobileNetV2	99.69	94.14	94.21
DenseNet121	99.76	96.26	94.86
VGG16	99.90	94.63	<b>95.02</b>

highest test accuracy was achieved by the VGG16 model, which means that it was the best in generalization and making accurate predictions on unseen data. It demonstrates that the model has learned the underlying patterns and features in the data and can make reliable predictions. It suggests that the model is well-suited for the given task and dataset and has the potential to perform well in practical applications.

### C. Class-wise Analysis

We have reported the precision, recall, and F1-score for the VGG16 model for each class in Table III. While accuracy is a useful metric, precision, recall, and F1-score provide a more comprehensive understanding of the performance, especially in imbalanced datasets and situations where false positives or false negatives have different implications. These metrics offer insights into the predictive behavior of the model, allowing for better assessment and comparison across different models or approaches. The model has achieved the highest precision, recall, and F1-score for the Powdery Mildew disease, whereas the Target Spot class has the minimum value.

### D. Error Analysis

Figure 2 represents the confusion matrix, a valuable tool for evaluating and interpreting the performance of a classification model. It helps in performance assessment, error analysis, and handling class imbalance and provides a visual representation of the model's predictions. By utilizing the information from the confusion matrix, we can make informed decisions to improve the model's performance and address specific challenges in the classification task. We observe that the 'Target spot' disease has the highest degree of similarity with 'Bacterial Blight'. The 'Aphids' and 'Army worm' diseases also have

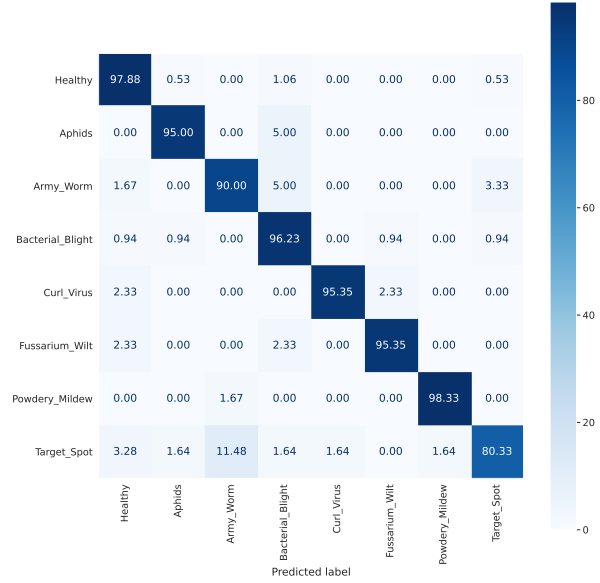


Fig. 2: Confusion Matrix

high similarity with the 'Bacterial Blight' class. The other classes have performance around 95%.

Finally, we have presented an analysis of the misclassified samples with the predicted and ground-truth labels in Figure 3. The first two images show a 'Bacterial Blight' sample to be classified as a 'Target Spot' and vice-versa. This happened due to the high inter-class similarity of the dataset. Figure 3 (c) shows a 'Target Spot' sample to be classified as 'Healthy'. However, the degree of disease was in the initial stage; hence, the leaf looked fresh, and the model classified it as 'Healthy'. Figure 3 (d)-(e) shows the samples to be classified as 'Powdery Mildew'. This may have happened due to the whitish spots in the samples, which is the prime distinguishing feature of this class. This would also be very difficult for humans to classify correctly. Finally, the last two samples show 'Healthy' samples to be classified as 'Diseased' ones. This has happened due to the high degree of background and diversity of the samples within the dataset.

## IV. CONCLUSION

Early identification and classification are imperative to fulfill the increasing food and industrial needs driven by the rapidly growing population. To address this, an approach based on transfer learning is presented in this study to detect and categorize cotton leaf diseases accurately. The proposed solution harnesses the power of deep transfer learning techniques to extract informative features and achieve highly accurate predictions. The system is extensively evaluated using the cotton leaf samples from different resources, with a promising accuracy of 95.02%. In contrast to prior studies that focused on a restricted range of classes, our approach involved merging samples from

TABLE III: Class-wise Performance Analysis using VGG16

Class Name	Precision	Recall	F1-Score	Support
Healthy	97.88	96.86	97.37	189
Aphids	95.00	95.00	95.00	60
Army Worm	90.00	98.18	93.91	60
Bacterial Blight	96.23	86.44	91.07	106
Curl Virus	95.35	97.62	96.47	43
Fussarium Wilt	95.35	95.35	95.35	43
Powdery Mildew	98.33	98.33	98.33	60
Target Spot	80.33	92.45	85.96	60

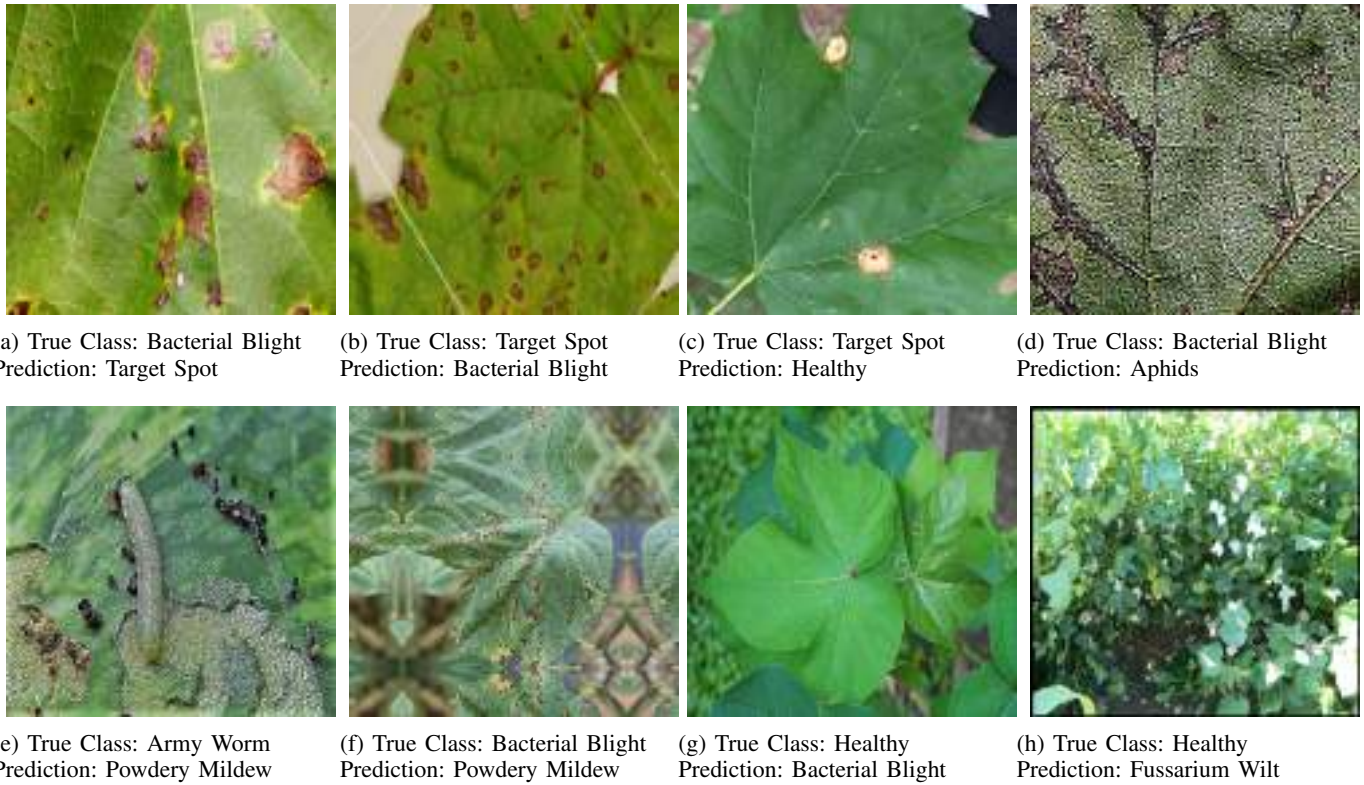


Fig. 3: Misclassified Samples from the Dataset

various sources and curating a dataset encompassing seven distinct cotton leaf diseases and a healthy class. This dataset includes a variety of difficulties brought on by heterogeneous sources, variations in backgrounds, lighting conditions, and other factors, ensuring its applicability in practical settings. In the future, the accuracy and resilience could be improved by including domain-specific information. This pipeline can be expanded to include the identification of more plant diseases and can be adapted to real-world applications to satisfy the needs of end users.

## REFERENCES

- [1] S. Panno, S. Davino, A. G. Caruso, S. Bertacca, A. Crnogorac, A. Mandić, E. Noris, and S. Matić, "A review of the most common and economically important diseases that undermine the cultivation of tomato crop in the mediterranean basin," *Agronomy*, vol. 11, no. 11, p. 2188, 2021.
- [2] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network (cnn) applied to plant leaf disease classification," *Agriculture*, vol. 11, no. 8, 2021.
- [3] M. H. Ashmafee, T. Ahmed, S. Ahmed, M. B. Hasan, M. N. Jahan, and A. Ashikur Rahman, "An efficient transfer learning-based approach for apple leaf disease classification," in *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2023, pp. 1–6.
- [4] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, 2018.
- [5] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
- [6] S. Ahmed, M. B. Hasan, T. Ahmed, M. R. K. Sony, and M. H. Kabir, "Less is more: lighter and faster deep neural architecture for tomato leaf disease classification," *IEEE Access*, vol. 10, pp. 68 868–68 884, 2022.
- [7] R. Mahbub, S. Shafiq Anuva, I. Towhid Khan, and Z. Islam, "A comparative analysis of efficient convolutional neural network based methods for plant disease classification," in *25th International Conference on Computer and Information Technology (ICCIT)*, 2022, pp. 870–875.
- [8] M. S. Morshed, S. Ahmed, T. Ahmed, M. U. Islam, and A. Ashikur Rahman, "Fruit quality assessment with densely connected convolutional neural network," in *12th International Conference on Electrical and Computer Engineering (ICECE)*, 2022, pp. 1–4.
- [9] M. A. Alanezi, M. S. Shahriar, M. B. Hasan, S. Ahmed, Y. A. Sha'aban, and H. R. E. H. Boucekara, "Livestock management with unmanned aerial vehicles: A review," *IEEE Access*, vol. 10, pp. 45 001–45 028, 2022.
- [10] A. M. Khan, A. Ashrafee, R. Sayera, S. Ivan, and S. Ahmed, "Rethinking cooking state recognition with vision transformers," in *2022 25th International Conference on Computer and Information Technology (ICCIT)*, 2022, pp. 170–175.
- [11] S. Ahmed, "Classification of plant disease from leaf images using few-shot learning," MSc Thesis, Department of Computer Science and Engineering (CSE), Islamic University of Technology, 2022.
- [12] 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.
- [13] A. Jenifa, R. Ramalakshmi, and V. Ramachandran, "Cotton leaf disease classification using deep convolution neural network for sustainable cotton production," in *2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES)*, 2019, pp. 1–3.
- [14] B. S. Prajapati, V. K. Dabhi, and H. B. Prajapati, "A survey on detection and classification of cotton leaf diseases," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 2499–2506.
- [15] D. Hughes, M. Salathé *et al.*, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv*, 2016.