Dicle University
Journal of Engineering

https://dergipark.org.tr/tr/pub/**dumf**
duje.dicle.edu.tr

Research Article

A Novel Deep Feature Extraction Approach Based on DenseNet201 and ResNet50 for Cotton Leaf Disease Detection

Nursena BAYĞIN[1*]

[1] Erzurum Technical University, Computer Engineering Department, nursena.baygin@erzurum.edu.tr, Orcid No: 0000-0003-4457-5503

ARTICLE INFO

Article history:

Received 6 January 2025
Received in revised form 5 February 2025
Accepted 20 February 2025
Available online 26 March 2025

Keywords:

Deep feature extraction, Cotton leaf diseases, Feature selection, Machine learning, Lightweight classification model

Doi: 10.24012/dumf.1614458

* Corresponding author

ABSTRACT

In this study, a new deep feature extraction approach is proposed for automatic detection of diseases observed on cotton plant leaves. In the proposed approach, feature extraction is performed using DenseNet201 and ResNet50 deep learning architectures. and the obtained feature vectors are combined. Then, the most informative features are selected with the Iterative Chi2 algorithm, and disease detection is performed using the Support Vector Machine (SVM) classifier. The developed model is tested on an open access dataset consisting of 2,137 cotton leaf images and 7 different classes (1 healthy, 6 diseased). 10-fold cross-validation and 80:20 hold-out cross-validation strategies are applied in the testing phase. As a result of the tests performed without using any data augmentation technique, 97.29% and 96.96% classification accuracies are obtained, respectively. The proposed approach makes significant contributions to the literature in terms of showing high success on the imbalanced dataset and providing a computationally lightweight architecture.

Introduction

Sustainable agriculture is among the areas where researchers are working intensively today. The efficient execution of the life cycle of agricultural products is of critical importance in terms of meeting the needs of the increasing population. The economic gains of countries are directly proportional to the quality and quantity of the products. In this context, the cotton plant is one of the products used in natural fiber production worldwide and provides significant contribution to the economy. Cotton, which is produced in approximately 100 countries, covers about 2.5% of the world's arable land [1].

Diseases in cotton leaves can reduce product productivity and cause economic losses. Traditionally, lesions in cotton leaves are evaluated by experts with the naked eye [2]. However, this method is time-consuming, costly and sometimes misleading. In terms of product productivity, monitoring large agricultural areas and detecting diseases quickly and accurately at low cost are of great importance [3-5]. Various approaches have been developed in the literature for the detection of these diseases, and methods such as image processing, machine learning and deep learning are frequently used among these approaches. When the diseases seen in cotton leaves are examined, approximately 80-90% of these diseases consist of certain species. These are diseases such as Cercospora, Bacterial Blight, Red Spot, Ascochyta Blight, Target Spot, etc. [6-7]. Cotton plants are vulnerable to pathogens such as bacteria, viruses, and fungi. In a study conducted by Parashar et al., diseases occurring on cotton leaves were detected and labeled and divided into two classes as healthy and diseased. In this study, a Convolutional Neural Network (CNN) model was developed to improve cotton leaf diseases and the MobileNetV2 model was used to optimize it. With the 99.91% accuracy rate obtained, the proposed model has the potential to be compatible with mobile devices and is shown to be usable by farmers [8]. Artificial intelligence-based applications can help increase cotton production efficiency by detecting cotton leaf diseases early. In a study conducted by Islam et al., the performances of deep network architectures such as Xception, VGG-16, VGG-19, and Inception-V3 were examined using the transfer learning approach. As a result of the study, an accuracy rate of 98.70% was achieved, and this accuracy value was obtained with the Xception deep network architecture [9].

As a result, meeting the needs of sustainable agriculture and increasing population and maintaining environmental balance are of critical importance today. Studies carried out on plants such as cotton and developed technological solutions have an important role in achieving these goals. The widespread use of artificial intelligence techniques in the agricultural sector will contribute to the creation of efficient, sustainable systems in the future. In this study, a new approach was proposed for the automatic detection of diseases occurring on the cotton plant, which is of critical importance today. With the proposed approach, a classification performance of over 96% was achieved on a six-class, unbalanced data.

Literature Survey

The use of machine learning for detecting cotton leaf diseases is an important factor in improving agricultural productivity. As seen in Table 1, when the studies in the literature are examined, it is seen that different machine learning techniques, especially deep learning models, have been developed to increase the diagnosis and accuracy rates of the disease.

Table 1. Recent studies on cotton leaf disease detection methods in the literature

Author (s) and Year	Dataset (s)	Disease	Method (s)	Result (s) (%)	Key Points
Azath M. et al. 2021 [10]	2400 image	4 class -Bacterial blight -Healthy -Leaf miner -Spider mite	CNN	Acc. = 96.4	-Detection of cotton leaf disease and pests -Feature extraction process -K-fold cross-validation -Unbalanced dataset
Chitranjan K. et al. 2023 [11]	2293 image	4 class -Diseased cotton leaf -Diseased cotton plant -Fresh cotton leaf -Fresh cotton plant	DCNN	Acc. = 97.98	-Identify and predict cotton plant disease -Data augmentation
Kaur A. et al. 2024 [12]	1711 image	4 class -Bacterial blight -Curl virus -Fusarium wilt -Healthy	VGG16	Acc. = 95.5	-Disease management of cotton crops -Small sample size -Data augmentation
Kukadiya H. et al. 2024 [13]		4 class -Bacterial blight -Curl virus -Fusarium wilt -Healthy	VGG16 InceptionV3 DCNN	Acc. = 98	-Early detection of diseases affecting cotton leaves -Small size dataset
Kavinandhan B. et al. 2024 [14]	1786 image	4 class -Bacterial blight, -Curl virus, -Fusarium wilt -Healthy	DCGAN InceptionV3 VGG16 ResNet 50 MobileNet V2	Acc. = 99	-Automated computer vision detection for cotton leaf diseases -Unbalanced dataset -Data augmentation

			CNN		
Rai K. C. et al. 2024 [15]	1700 image	2 class -Diseased cotton plant -Fresh cotton plants 4 class -Blight -Curl -Healthy -Wilt	InceptionV3, InceptionResNetV2 VGG16 MobileNet Xception	Acc. = 99.48	-Detection and classification of diseased cotton leaves and plants -Data augmentation
Saleh A. et al. 2024 [16]	2293 image	4 class -Cotton leaf, diseased -Cotton plant, -Fresh cotton leaf, -Fresh cotton plant	SVM-VGG16 MobileNet	Acc. = 91 Acc. = 99	-Detection cotton diseases before they spread to crops -Data augmentation

When the studies given in Table 1 are examined, it is seen that these studies generally focus on 4 classes [10,13]. In addition, the majority of the studies use data augmentation methods [11,14-16]. The most important reason for this situation is that data sets are generally small in size [12], have an unbalanced distribution [10,14] and artificial intelligence methods require a large amount of data. The aim of this study is to eliminate all these deficiencies. In this context, it is aimed to classify an unbalanced data set without using any data augmentation technique. A multi-class, unbalanced data set was classified with the developed model and a classification result of over 96% was obtained.

Literature Gap

Some literature gaps observed as a result of the literature studies given in Table 1 are listed below:

- Methods performed in the literature on image classification generally use deep learning approaches. However, these studies generally prefer the end-to-end training solution.
- Deep learning approaches require a lot of data and a balanced data set due to their structure. For this reason, researchers generally need methods such as data augmentation and balance the data set with this approach.
- Since researchers generally use the end-to-end training approach, the computational complexity of these architectures is quite high.

This study aimed to fill the literature gaps listed above. In this context, an open access data set was used and the classification process was performed directly on the data set without any preprocessing.

Motivation and Proposed Model

In this study, a new machine learning approach has been developed for automatic detection of diseases occurring in cotton plants. The developed architecture was tested on an open access cotton leaf disease dataset and a classification accuracy of over 96% was achieved. The dataset used in the study has 7 classes and is basically unbalanced. The researchers who shared the dataset applied data augmentation techniques to balance the dataset. However, in this study, the classification process was performed using only raw leaf images.

Deep feature extraction was achieved in the study using the transfer learning approach. For this process, some pre-trained deep network architectures were tested and the two deep network architectures with the highest performance were preferred. Later, the feature vector was obtained using the fully connected and pooling layers of these networks. The feature selection algorithm was applied to reduce the size of the obtained feature vector and to reduce the computational complexity of the architecture. In this step where the Iterative Chi2 method was preferred, the optimum number of features was selected and thus the selected feature vector was obtained. In the last stage of the system, the SVM method [17], a well-known classification algorithm in the literature, was preferred. An accuracy value of over 96%

was achieved with the SVM algorithm, which is basically a lightweight classification approach.

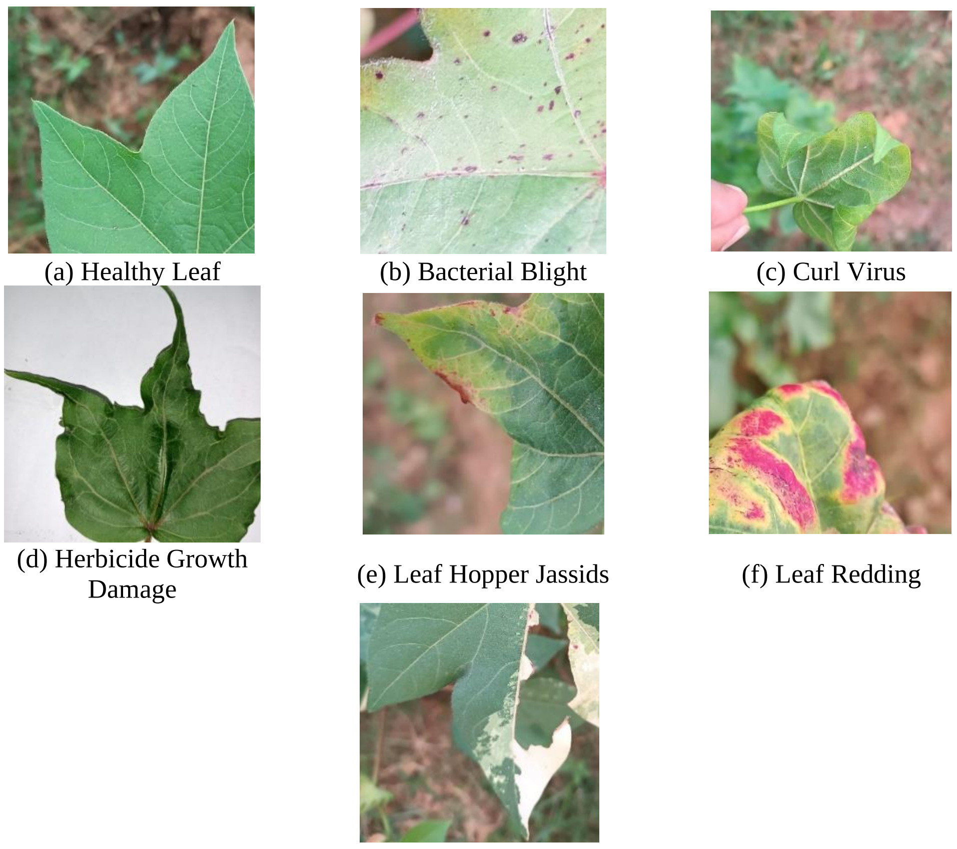
Although DenseNet201 and ResNet50 are deep learning architectures with high computational complexity, our approach does not perform end-to-end training. Instead, we use these networks in a transfer learning manner, extracting feature vectors from pre-trained models. This significantly reduces the computational burden compared to full model training. Additionally, by employing feature selection using the Iterative Chi2 algorithm, we minimize the number of features processed in the classification phase, making our model more efficient. The feature selection step reduces the final input size to the classifier, thereby improving both inference speed and memory efficiency.

Dataset

In this study, two open-access datasets for cotton leaf disease detection were utilized to enhance the robustness and generalizability of the proposed model. The first dataset was obtained from the National Cotton Research Institute field in Gazipur, Bangladesh, while the second dataset was collected from cotton fields in India. These datasets contain images of cotton leaves affected by various diseases, as well as healthy leaves, captured under different environmental conditions.

The first dataset consists of a total of 2137 original images, which are divided into 7 different classes: bacterial blight, curl virus, herbicide growth damage, leaf hopper jassids, leaf reddening, leaf variegation, and healthy leaves [18]. The images were systematically collected over the course of approximately one year using a Redmi Note 11s model smartphone under varying lighting conditions. Additionally, the dataset contains 7000 augmented images created through data augmentation techniques. However, in this study, only the original images were used, and the augmented images were excluded to evaluate the classification performance on raw data. The images corresponding to the first dataset are shown in Figure 1.

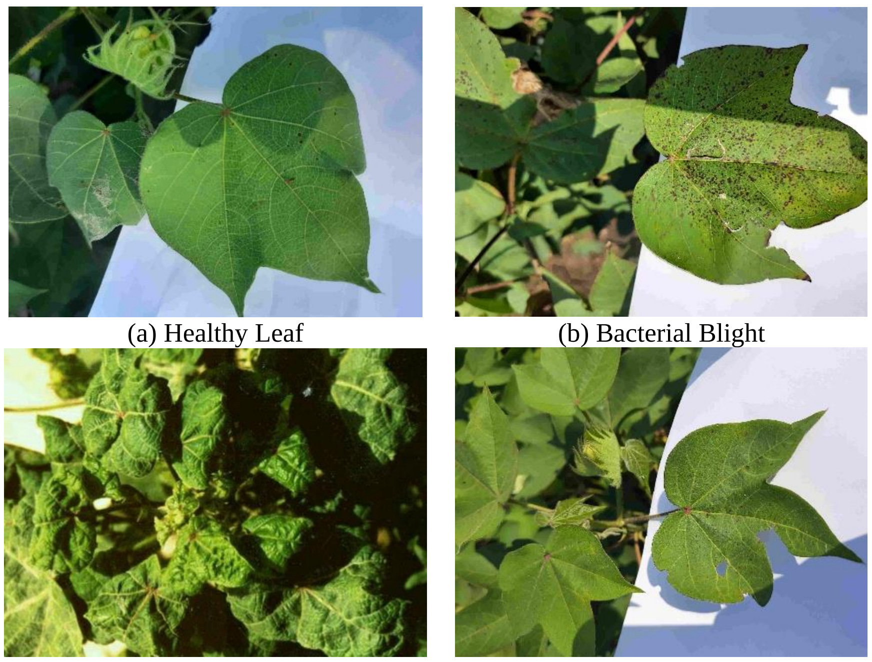
The second dataset comprises 980 images of cotton leaves affected by various diseases, as well as healthy leaves. The images are meticulously organized into different disease categories and collected under diverse environmental conditions to improve the robustness of machine learning models. Similar to the first dataset, this dataset also exhibits an imbalanced distribution, with certain disease categories having significantly fewer images than others. The dataset includes images of healthy cotton leaves, as well as leaves affected by bacterial blight, fusarium wilt, and curl virus. The images corresponding to the second dataset are shown in Figure 2. By incorporating two datasets from different geographical regions, this study aims to improve the generalization ability of the model and evaluate its robustness against varying environmental factors.


(a) Healthy Leaf (b) Bacterial Blight (c) Curl Virus

(d) Herbicide Growth Damage (e) Leaf Hopper Jassids (f) Leaf Redding

(g) Leaf Variegation

Figure 1. Cotton leaf diseases for Dataset-1


(a) Healthy Leaf (b) Bacterial Blight

(c) Herbicide Growth Damage (d) Leaf Hopper Jassids

Figure 2. Cotton leaf diseases for Dataset-2

Proposed Method

In this study, a new deep feature extraction approach is proposed for automatic identification and classification of cotton leaf disease. The deep feature extraction-based architecture developed for this unbalanced dataset, where no data augmentation method is applied, is given in Figure 3.

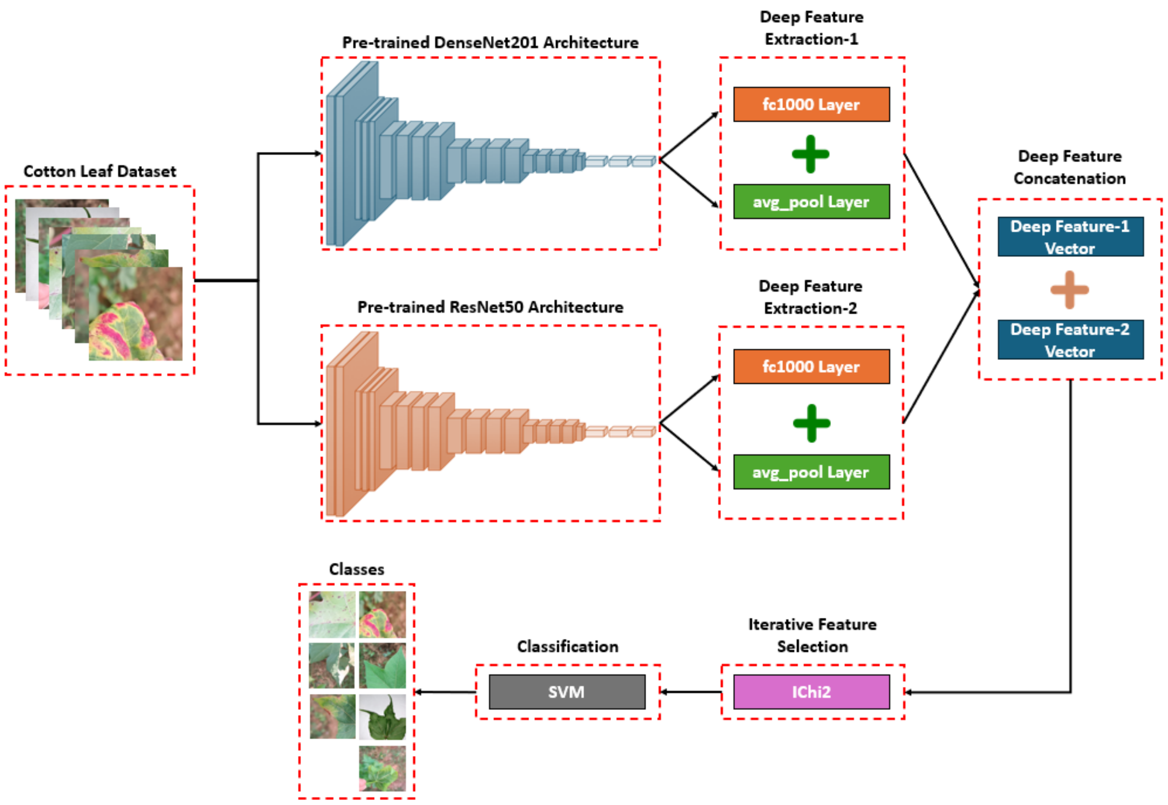

Cotton Leaf Dataset
Pre-trained DenseNet201 Architecture
Pre-trained ResNet50 Architecture
Deep Feature Extraction-1
fc1000 Layer
avg_pool Layer
Deep Feature Extraction-2
fc1000 Layer
avg_pool Layer
Deep Feature Concatenation
Deep Feature-1 Vector
Deep Feature-2 Vector
Iterative Feature Selection
IChi2
Classification
SVM
Classes

Figure 3. DenseNet201 [19] and ResNet50 [20] based deep feature extraction

As given in Figure 3, the system first takes the cotton leaf dataset as input. Then, it gives these images as input to both deep network architectures. In this phase, pre-trained networks are used and deep feature extraction is performed. In this way, instead of end-to-end training, previously determined weights of the networks are used and an approach with lower computational complexity is obtained. In this phase, features are extracted using the last fully connected layers and the last pooling layers of the networks. After this phase, the feature vectors obtained using both networks are combined and a feature vector representing the performance of both networks is obtained. In the next phase of the system, feature selection is performed using the iterative Chi2 algorithm. The most informative features are selected with this method. The reason for using the iterative approach is to determine the optimum number of feature vectors. The selected features in this phase are given as input to SVM [17], a shallow classifier. The selected feature vectors are classified with the SVM algorithm and the estimated values are obtained. These steps are explained in detail in the subsections.

Feature Extraction and Concatenation

The most important phase of the developed architecture is features extraction. In this phase, where the deep feature extraction approach is adopted, feature extraction is performed using the final pooling and fully connected layers of DenseNet201[19] and ResNet50 [20] architectures. In this context, the pseudo codes of DenseNet201 and ResNet50 architectures are given in Algorithms 1 and 2, respectively.

Algorithm 1. Pseudocode of DenseNet201 Architecure

```
No  DenseNet201 Architecture
def DenseNet201():
    x = Conv(64, 7x7, stride=2)  # Initial Conv
    x = MaxPool(3x3, stride=2)   # Initial Pool
    # Dense Blocks with Transition Layers
    for layers, growth_rate in [(6, 32), (12, 32), (48, 32), (32, 32)]:
        x = DenseBlock(x, layers, growth_rate)
        if layers != 32:  # No transition after the last DenseBlock
            x = TransitionLayer(x, 0.5)
    # Classification
    x = GlobalAvgPool(x)  # Global Average Pooling
    x = FullyConnected(x, num_classes, activation=softmax)  # Fully Connected Layer
    return x
def DenseBlock(x, layers, growth_rate):
    for _ in range(layers):
        x = Concatenate([x, Conv(growth_rate, 3x3, padding=same, activation=ReLU, batch_norm=True)])
    return x
def TransitionLayer(x, reduction):
    return AvgPool(Conv(int(x.filters * reduction), 1x1, activation=ReLU, batch_norm=True), 2x2, stride=2)
def FullyConnected(x, units, activation):
    return Dense(x, units, activation=activation)
```

Algorithm 2. Pseudocode of ResNet50 Architecure

```
No  ResNet50 Architecture
def ResNet50():
    x = Conv(64, 7x7, stride=2, activation=ReLU, batch_norm=True)  # Initial Conv
    x = MaxPool(3x3, stride=2)  # Initial Pool
    # Residual Blocks
    for filters, blocks, strides in [(64, 3, 1), (128, 4, 2), (256, 6, 2), (512, 3, 2)]:
        x = ResidualGroup(x, filters, blocks, strides)
    # Classification
    x = GlobalAvgPool(x)
    return FullyConnected(x, num_classes, activation=softmax)
def ResidualGroup(x, filters, blocks, strides):
    x = ResidualBlock(x, filters, strides)  # First block with downsampling
    for _ in range(1, blocks):
        x = ResidualBlock(x, filters)  # Remaining blocks
    return x
def ResidualBlock(x, filters, strides=1):
  shortcut = Conv(filters * 4, 1x1, stride=strides, batch_norm=True) if strides > 1 or x.filters != filters * 4 else x
```

```
    x = Conv(filters, 1x1, stride=strides, activation=ReLU, batch_norm=True)
    x = Conv(filters, 3x3, activation=ReLU, batch_norm=True)
    x = Conv(filters * 4, 1x1, batch_norm=True)
    return ReLU(Add(x, shortcut))
def FullyConnected(x, units, activation):
    return Dense(x, units, activation=activation)
```

The developed model performs deep feature extraction using two pre-trained networks. These networks are DenseNet201 and ResNet50 architectures, respectively. In the feature generation phase, features are extracted using the final pooling and fully connected layer. The system uses the “fc1000” and “avg_pool” layers of the DenseNet201 architecture. These layers represent the fully connected and pooling layers, respectively. Again, 1000 and 1920 features are produced from these layers, respectively. In this way, a feature vector of 2920 (=1000+1920) length is obtained per image in total using the DenseNet201 architecture. A similar situation is valid for the ResNet50 architecture, and features are produced from the “fc1000” and “avg_pool” layers in this architecture. Similar to the DenseNet201 architecture, 1000 and 1920 features are produced from these layers, respectively, and thus a feature vector of 2920 (=1000+1920) length is obtained. In the next stage of the architecture, the feature vectors obtained using both deep networks are combined and a new feature vector of 5840 (=2920 from DenseNet201 + 2920 from ResNet50) length is provided, which uses the power of the two networks. A block diagram summarizing these process steps is given in Figure 4.

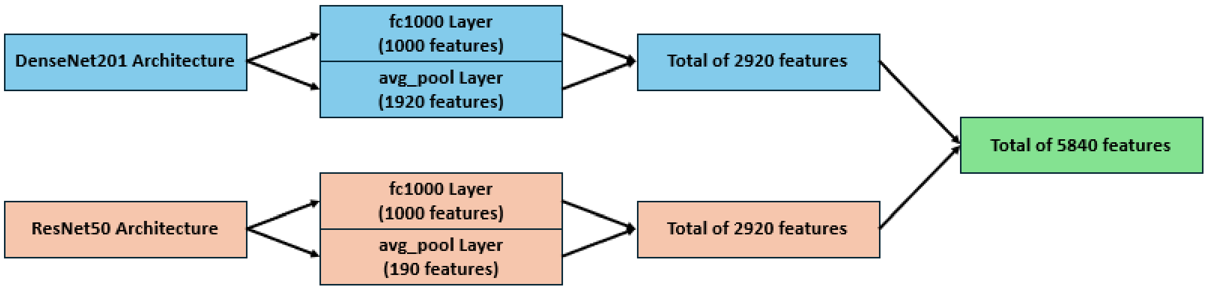

DenseNet201 Architecture
fc1000 Layer (1000 features)
avg_pool Layer (1920 features)
Total of 2920 features
ResNet50 Architecture
fc1000 Layer (1000 features)
avg_pool Layer (190 features)
Total of 2920 features
Total of 5840 features

Figure 4. Deep features extraction and features concatenation

As given in Figure 4, the system uses two separate pre-trained deep network architectures to extract features from final pooling (avg_pool) and fully connected layers (fc1000). Then, 2920 features are produced by each network and these produced features are combined to obtain a new feature vector with a total length of 5840.

Feature Selection

Iterative Chi2 (IChi2) [21] algorithm is an advanced version of Chi2 algorithm which is frequently used in literature. With this method, the most informative features are selected iteratively. The algorithm takes the generated deep features and prediction values as input. Then, the weights of all features are determined using Chi2 algorithm and the features are ranked according to these weight values. After this process, the features are selected and classified iteratively in order. The point where the highest classification accuracy is obtained is determined as the optimum number of features. kNN algorithm is used as classifier in IChi2 algorithm. The selected features are classified with this method in each iteration and the number of features for which the highest classification result is calculated is determined. The pseudo code of IChi2 algorithm used to select the most informative features in the developed architecture is given in Algorithm 3.

Algorithm 3. Iterative Chi2 approach

```
IChi2 procedures
def IterativeChi2(data, target, max_features):
    # Step 1: Initialize variables
    index = FeatureSelector(data, target)  # Qualified feature indices
    selected_features = []
    # Step 2: Iteratively select features
    for i in range(max_features):
        losses = []
        for j in index:
            temp_features = selected_features + [data[:, j]]
            loss = ComputeLoss(temp_features, target)
            losses.append(loss)
        # Step 3: Select feature with minimum loss
        min_loss_index = ArgMin(losses)
        selected_features.append(data[:, index[min_loss_index]])
        index.remove(index[min_loss_index])
    # Step 4: Return final selected features
    return selected_features
def FeatureSelector(data, target):
    # Implement feature selection logic based on Chi2 criteria
    return QualifiedIndices(data, target)
def ComputeLoss(features, target):
    # Calculate the loss value for the selected feature vector
    return LossFunction(features, target)
def ArgMin(values):
    # Find the index of the minimum value in the list
    return values.index(min(values))
```

Classification

The last phase of the developed model is classification. For this process, a shallow classifier, the SVM [17] algorithm, was used. In this algorithm, which works very well on high-dimensional data sets, the 3rd degree polynomial kernel (Cubic SVM) was preferred. Two different strategies were applied to verify the developed model. These are 10-fold CV and hold-out (80:20) CV techniques, respectively.

Experimental Results and Discussion

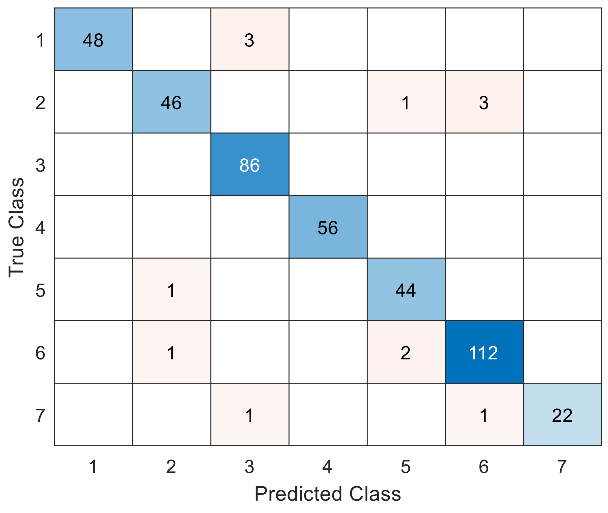
Experimental Setup

The model developed in this research consists of three main phases. These phases are deep feature extraction, iterative feature selection and classification, respectively. DenseNet201[19] and ResNet50 [20] architectures were used for deep feature extraction. Feature extraction was provided from these pre-trained networks through transfer learning. Feature vectors were obtained using the last fully connected layer and the last pooling layers. In the feature selection phase of the model, the IChi2 [21] algorithm, an advanced version of the Chi2 method, was used. In the last phase of the architecture, the SVM [17] algorithm, a well-known method in the literature, was preferred.

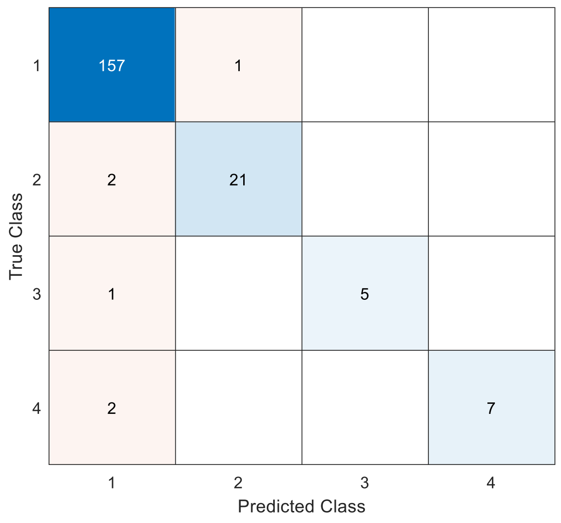
The developed model was coded on the MATLAB 2021b platform. In addition, the MATLAB Classification Learner Toolbox was used for the classification process. The model was developed on a basic server and no GPU card was used. All operations were carried out on the CPU. The computer used in the test process has an Intel Xeon 2.7 GHz processor, 256 GB RAM and 500 GB hard disk, respectively, and there is no GPU card among the hardware. In the testing phase of the model, 10-fold CV and 80:20 hold-out CV strategies were applied. To evaluate the classification performance of the model, accuracy, precision, recall and F1-Score values were calculated. In order to calculate these values, a confusion matrix was created for each validation strategy.

Results

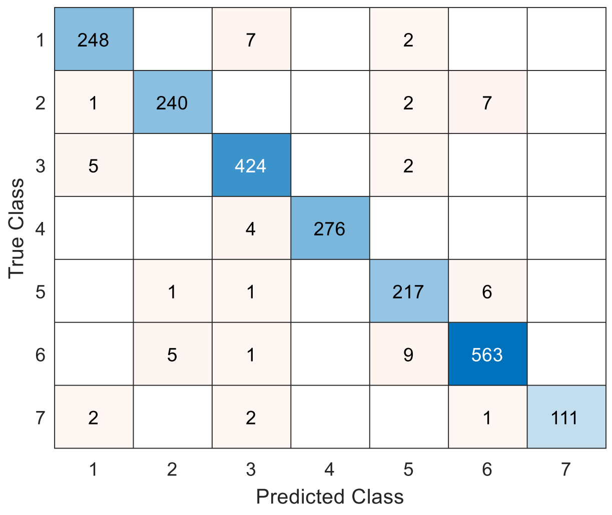
The developed model was tested on a dataset consisting of 7 classes and containing open access images of cotton plant diseases. In the test phase, 10-fold and 80:20 hold-out CV strategies were applied. Confusion matrix was calculated for both strategies and performance metric values were determined using this matrix. Confusion matrices calculated for 10-fold and 80:20 hold-out CV techniques as a result of the test processes are given in Figure 5.



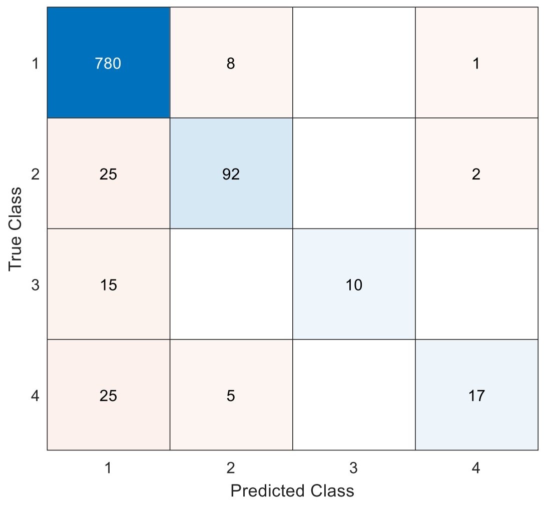
(a)



(b)



(c)



(d)

Figure 5. Confusion matrices obtained as a result of the test operations (a) 80:20 hold-out CV for Dataset-1 (b) 80:20 hold-out CV for Dataset-2 (c) 10-fold CV for Dataset-1 (d) 10-fold CV for Dataset-2

Confusion matrices obtained for both verification techniques are given in Figure 5. The classes in these matrices (Figure 4-(a) and (c)) are as follows: 1- Healthy Leaf, 2- Bacterial Blight, 3- Curl Virus, 4- Herbicide Growth Damage, 5- Leaf Hopper Jassids, 6- Leaf Redding and 7- Leaf Variegation (for Dataset-1). In addition, for Dataset-2, classes are as follows: 1- Healthy Leaf, 2- Bacterial Blight, 3- Curl Virus and 4- Fussarium Wilt, respectively. In this context, the performance metric values calculated using the matrices given in Figure 4 are given in Table 2.

Table 2. Performance metric values

Metric	80:20 Hold-out CV	10-fold CV	80:20 Hold-out CV	10-fold CV
Accuracy	96.96	97.29	96.94	91.73
Average Precision	97.37	97.45	98.09	91.23
Unweighted Average Recall	96.14	97.00	87.95	63.09
Average F1 Score	96.69	97.21	92.47	71.38
Geometric Mean	96.08	96.99	87.57	57.67

When the results given in Table 2 are examined, it is seen that the developed method reaches an accuracy value higher than 96% on both verification techniques for Dataset-1. In addition, this situation is valid for all metric values. When the similar situation is analyzed for Dataset-2, it is seen that the accuracy values are 96.94% and 91.73% for 80:20 hold-out CV and 10-fold CV, respectively. The obtained results show that the proposed method can classify cotton plant diseases with high accuracy. Both of the open access datasets used in this research have an unbalanced distribution and contains a limited amount of raw images. However, the developed model has managed to overcome all these problems.

Discussion

The dataset used in the study is shared as open access and basically contains 2137 images. This dataset, which contains images of disease types of cotton plants, has 7 classes including healthy class images. These classes are Healthy Leaf, Bacterial Blight, Curl Virus, Herbicide Growth Damage, Leaf Hopper Jassids, Leaf Redding and Leaf Variegation, respectively. In this context, the comparison results with other studies using the same dataset are given in Table 3.

Table 3. Comparison results (for Dataset-1)

Author(s) and Year	Dataset	Method	Validation	Result(s)
Bishshash et al., 2024	7000 augmented images, 7 class	Data augmentation and Inception V3	80:20 hold-out CV	Acc.=96.03

Our Method	2137 raw images, 7 class	DenseNet201, ResNet50, IChi2, SVM	80:20 hold-out CV	Acc.=96.96
				APre.=97.37
				UAR.=96.14
				AF1.=96.69
				Gm.=96.08

			10-fold CV	Acc.=97.29
				APre.=97.45
				UAR.=97.00
				AF1.=97.21
				Gm.=96.99

*Acc.=Accuracy, Apre.=Average Precision, UAR.=Unweighted Average Recall, AF1.=Average F1 Score, Gm.=Geometric Mean

The studies given in Table 3 are the studies that used the same dataset as the dataset used in this research. As can be seen from the table, Bishhash et al. [18] applied only the 80:20 hold-out CV strategy as the verification technique. Two different verification techniques were used in our research, and the classification performance in both approaches was higher than the other research. In addition, Bishhash et al. [18] trained the InceptionV3 architecture they used in their research end-to-end. Deep learning architectures need a lot of data to perform well. For this reason, the data augmentation approach was used in their research, and thus both the number of images was increased and a balanced dataset was obtained. In our research, the raw image dataset was used directly without using any data augmentation approach. Despite this situation, high classification success was achieved in all performance metric values. As stated in other sections, the IChi2 [21] approach was used in this study, and thus the most informative features were selected. An advanced version of the Chi2 algorithm was used in the feature selection procedure. This approach, called Iterative Chi2, aims to determine the optimum number of features to be selected. In the test operations performed, the IChi2 algorithm selected a total of 788 features from the feature vector of length 5840. In the process of selecting the feature vector, the iteration range was determined to be between 100-1000. In other words, the first 100 features with the highest weight were directly selected and then the remaining features were tested iteratively to determine the optimum point. A graph of this test operation is given in Figure 6.

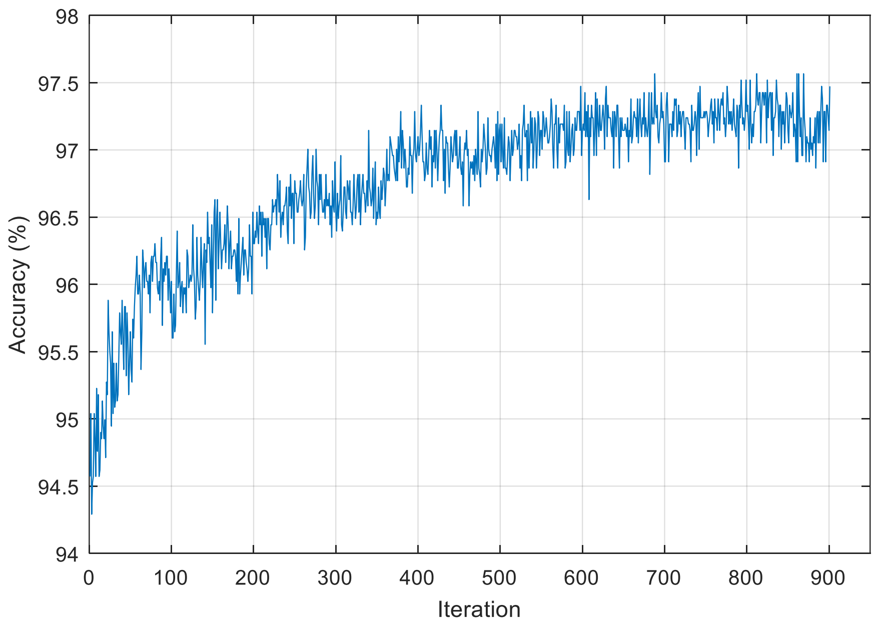


Figure 6. Changes in the classification accuracies obtained with the IChi2 [21] method

As given in Figure 6, the developed model showed an increase from 94% to approximately 97%. In the test process given here, the 10-fold CV method was used as a verification technique and a total of 788 features were selected according to the accuracy values obtained in Figure 6. In the last test phase of the model, classification algorithms were considered. At this stage, Decision Tree (DT), SVM, kNN and Neural Network (NN) algorithms were tested respectively. The results of this test process are as given in Figure 7.

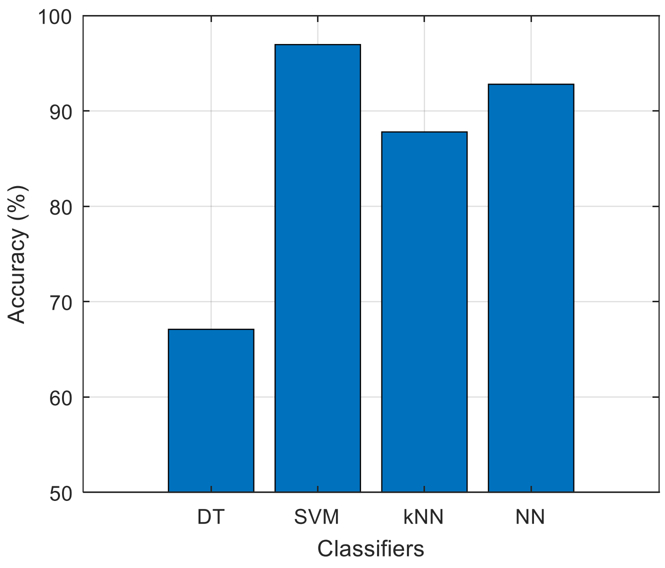


Figure 7. Performance of classification algorithms on the Dataset-1

As given in Figure 7, the highest classification accuracy was obtained with the SVM algorithm, and the results given in this graph were obtained by applying the 10-fold CV strategy. In this test process, where a total of 788 features were classified, the lowest accuracy value of 67.1% was obtained with the DT algorithm.

The proposed model is particularly designed to address computational efficiency concerns by avoiding full model training and instead leveraging pre-trained deep networks for feature extraction. This approach is computationally much lighter compared to fine-tuning or training a deep model from scratch. Additionally, after feature extraction, the feature selection process further optimizes the model by reducing the number of dimensions fed into the classifier. To evaluate the feasibility of real-time applications, the inference time per image was measured, showing that the feature extraction and classification pipeline can process an image in approximately 2.75 seconds on a standard CPU configuration. This demonstrates that the model can be adapted for real-world applications with further optimizations, such as hardware acceleration via GPUs or edge AI solutions. To further evaluate the effectiveness of the proposed approach, additional experiments were conducted using DenseNet201 and ResNet50 separately, without deep feature concatenation. In these scenarios, deep features were extracted from each network individually, followed by feature selection with the IChi2 algorithm and classification using the SVM classifier. This experiment was designed to determine whether the integration of DenseNet201 and ResNet50 significantly contributes to classification performance or if a single network is sufficient for accurate disease detection. The comparative results of these experiments are presented in Figure 8.

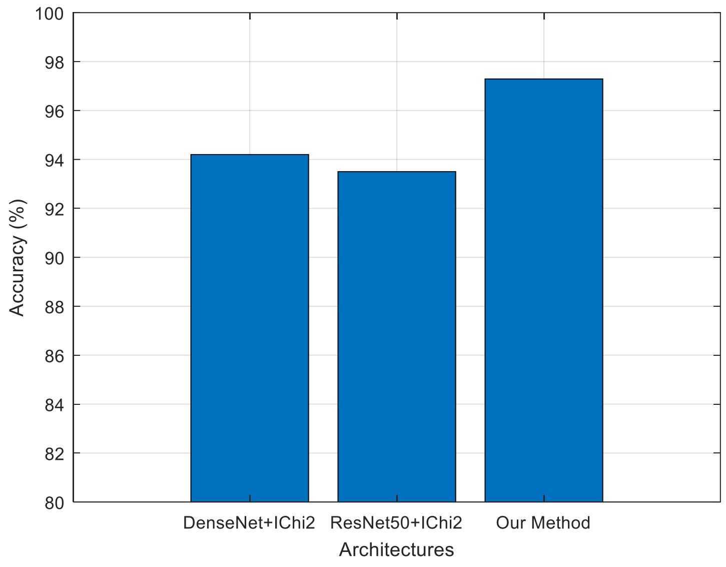


Figure 8. Performance of classification algorithms on the Dataset

As shown in the figure, DenseNet201 alone achieved an accuracy of approximately 94.2%, while ResNet50 alone reached 93.5%. Although both architectures performed well individually, their accuracy was notably lower than the 97.29% achieved when using their combined feature representations. This highlights the complementary nature of the features extracted from both networks, reinforcing that their integration enables a more comprehensive and discriminative feature representation. Furthermore, despite the datasets being imbalanced and no data augmentation being applied, the proposed approach successfully achieved high accuracy across both datasets. These findings confirm that the combination of deep feature extraction and feature selection plays a critical role in maximizing classification performance while maintaining computational efficiency.

One of the key contributions of this study is demonstrating that high classification accuracy can be achieved without the use of data augmentation techniques. Although data augmentation is commonly employed to balance datasets and improve model performance, the proposed approach successfully classified imbalanced datasets from two different regions with high accuracy rates, achieving over 97% classification success. This result is particularly significant as it shows that the method can effectively extract discriminative features without artificially increasing the number of training samples.

The fact that both datasets were imbalanced yet still yielded high classification performance further emphasizes the strength of the proposed model. Instead of relying on data augmentation, this study focused on feature selection and efficient model design to enhance classification performance. This approach ensures that the model is not overly dependent on synthetic data generation and can generalize well to real-world scenarios. The ability to classify raw images with high accuracy is a valuable outcome, as it demonstrates the potential for real-world applicability without the need for complex preprocessing techniques.

Conclusion

Ensuring continuity in the agricultural sector is a very important issue for today's world. In particular, it is necessary to use scientific approaches to protect the ecological balance, obtain high-yield products and transfer soils efficiently to future generations. Today, artificial intelligence technologies are actively used by many different disciplines and very successful results are obtained. In this study, it was aimed to detect the diseases of the cotton plant, which has a very important place among agricultural plants.

Within the scope of this research, two open-access datasets were tested with the developed model, and multi-class classification was performed. The first dataset consists of 7 classes, including 1 healthy and 6 diseased categories, and contains a total of 2137 images collected using a simple phone camera. Similarly, the second dataset includes 980 images, covering both healthy and diseased cotton leaves with an imbalanced class distribution. The images in both datasets were collected under diverse environmental conditions to enhance model robustness. With the proposed architecture, deep feature extraction was performed on these datasets, and classification was conducted using the SVM algorithm. The developed model was designed within a lightweight framework, avoiding computationally expensive end-to-end training. As artificial intelligence-based methods typically require large amounts of data for optimal performance, this study aimed to classify raw images without applying any data augmentation techniques. Despite the challenges posed by imbalanced datasets, the proposed approach successfully achieved classification accuracy exceeding 97%. The results obtained in this research demonstrate that the developed method can be effectively utilized for detecting cotton plant diseases across different datasets and environmental conditions.

Ethics committee approval and conflict of interest statement

Ethics committee approval is not required for this study. The author declares that there is no conflict of interest with any person/institution in the prepared article.

Authors' Contributions

The author designed the model and processes, created the analysis methods, developed the software code to implement the processes, performed the tests, evaluated the results and contributed to manuscript preparation.

References

[1] R. F. Caldeira, W. E. Santiago, and B. Teruel, "Identification of cotton leaf lesions using deep learning techniques," *Sensors*, vol. 21, no. 9, 2021, doi: 10.3390/s21093169.

[2] P. R. Rothe and R. V. Kshirsagar, "Cotton leaf disease identification using pattern recognition techniques," *2015 Int. Conf. Pervasive Comput. Adv. Commun. Technol. Appl. Soc. ICPC 2015*, vol. 00, no. c, pp. 1–6, 2015, doi: 10.1109/PERVASIVE.2015.7086983.

[3] B.-A. S. Bashish Al Dheeb,Braik Malik, "Detection and Classification of Leaf Diseases using K-means-based Segmentation and Neural-networks-based Classification," vol. 10, 2011.

[4] T. Rumpf, A. K. Mahlein, U. Steiner, E. C. Oerke, H. W. Dehne, and L. Plümer, "Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance," *Comput. Electron. Agric.*, vol. 74, no. 1, pp. 91–99, 2010, doi: 10.1016/j.compag.2010.06.009.

[5] C. Hillnhütter and A. K. Mahlein, "Neue Ansätze zur frühzeitigen Erkennung und Lokalisierung von Zuckerrübenkrankheiten," *Gesunde Pflanz.*, vol. 60, no. 4, pp. 143–149, 2008, doi: 10.1007/s10343-008-0196-0.

[6] A. Jenifa, R. Ramalakshmi, and V. Ramachandran, "Cotton Leaf Disease Classification using Deep Convolution Neural Network for Sustainable Cotton Production," *2019 Int. Conf. Clean Energy Energy Effic. Electron. Circuit Sustain. Dev. INCCES 2019*, pp. 19–21, 2019, doi: 10.1109/INCCES47820.2019.9167715.

[7] B. S. Prajapati, V. K. Dabhi, and H. B. Prajapati, "A survey on detection and classification of cotton leaf diseases," *Int. Conf. Electr. Electron. Optim. Tech. ICEEOT 2016*, pp. 2499–2506, 2016, doi: 10.1109/ICEEOT.2016.7755143.

[8] N. Parashar and P. Johri, "Deep Learning for Cotton Leaf Disease Detection," *Proc. - 2nd IEEE Int. Conf. Device Intell. Comput. Commun. Technol. DICCT 2024*, no. Dl, pp. 158–162, 2024, doi: 10.1109/DICCT61038.2024.10533021.

[9] M. M. Islam *et al.*, "A deep learning model for cotton disease prediction using fine-tuning with smart web application in agriculture," *Intell. Syst. with Appl.*, vol. 20, no. January, p. 200278, 2023, doi: 10.1016/j.iswa.2023.200278.

[10] M. Azath, M. Zekiwos, and A. Bruck, "Deep Learning-Based Image Processing for Cotton Leaf Disease and Pest Diagnosis," *J. Electr. Comput. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/9981437.

[11] C. K. Rai and R. Pahuja, "Classification of Diseased Cotton Leaves and Plants Using Improved Deep Convolutional Neural Network," *Multimed. Tools Appl.*, vol. 82, no. 16, pp. 25307–25325, 2023, doi: 10.1007/s11042-023-14933-w.

[12] A. Kaur, V. Kukreja, M. Kumar, A. Choudhary, and R. Sharma, "A Fine-tuned Deep Learning-based VGG16 Model for Cotton Leaf Disease Classification," *2024 5th Int. Conf. Emerg. Technol. INCET 2024*, pp. 1–6, 2024, doi: 10.1109/INCET61516.2024.10593164.

[13] H. Kukadiya, N. Arora, D. Meva, and S. Srivastava, "An ensemble deep learning model for automatic classification of cotton leaves diseases," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 33, no. 3, pp. 1942–1949, 2024, doi: 10.11591/ijeecs.v33.i3.pp1942-1949.

[14] B. Kavinandhan, R. Pranav, and M. Ganesan, "A Hybrid Approach for Cotton Leaf Disease Detection using DCGAN and Diverse CNN Models," *2024 5th Int. Conf. Innov. Trends Inf. Technol. ICITIIT 2024*, pp. 1–8, 2024, doi: 10.1109/ICITIIT61487.2024.10580758.

[15] C. K. Rai and R. Pahuja, *An ensemble transfer learning-based deep convolution neural network for the detection and classification of diseased cotton leaves and plants*, no. 0123456789. Springer US, 2024. doi: 10.1007/s11042-024-18963-w.

[16] A. Saleh *et al.*, “Machine Learning-based classification of cotton diseases using mobilenet and Support Vector Machine,” *2024 Int. Telecommun. Conf. ITC-Egypt 2024*, pp. 165–171, 2024, doi: 10.1109/ITC-Egypt61547.2024.10620532.

[17] W. S. Noble, “What is a support vector machine?,” *Nat. Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006.

[18] P. Bishshash, M. A. S. Nirob, M. H. Shikder, M. A. H. Sarower, D. T. Bhuiyan, and S. R. H. Noori, “A Comprehensive Cotton Leaf Disease Dataset for Enhanced Detection and Classification,” *Data Br.*, vol. 57, p. 110913, 2024, doi: 10.1016/j.dib.2024.110913.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[20] 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*, 2016, pp. 770–778.

[21] M. Erten and T. Tuncer, “Automated differential diagnosis method for iron deficiency anemia and beta thalassemia trait based on iterative Chi2 feature selector,” *Int. J. Lab. Hematol.*, vol. 44, no. 2, pp. 430–436, 2022.