European Journal of
Engineering and Applied Sciences

ISSN: 2651-3412 (Print) & 2667-8454 (Online)
Journal homepage: http://dergipark.gov.tr/EJEAS
Published by Çorlu Faculty of Engineering, Tekirdağ Namık Kemal University

European J. Eng. App. Sci. 6(1), 32-40, 2023

*Research Article*

# The Efficiency of Transfer Learning and Data Augmentation in Lemon Leaf Image Classification

**Ahmet Saygılı**[1, *]

[1] **Department of Computer Engineering, Çorlu Engineering Faculty, Tekirdağ Namık Kemal University, Tekirdağ, Turkey, 59860**
[1]**asaygili@nku.edu.tr**

*Received: 30.06.2023* *Accepted: 25.07.2023* *DOI: 10.55581/ejeas.1321042*

***Abstract:*** Leaf diseases in trees and plants are important factors that directly affect the yield of agricultural products. This problem may cause a decrease in the production capacity and profitability of farmers. For this reason, computer-aided detection and classification systems are needed to accurately detect plant diseases. In recent years, learning algorithms and image-processing techniques have been used effectively in the agricultural sector. In this study, the efficiency of transfer learning and data augmentation methods on a dataset consisting of lemon leaf images is examined and the classification of diseased and healthy lemon leaf images is performed. In our study, VGG16, ResNet50, and DenseNet201 transfer learning methods were applied both with and without data increment, and the effect of data augmentation on performance was evaluated. Among the deep transfer learning methods used, DenseNet201 gave the highest accuracy rate with 98.29%. This study shows that transfer learning methods can effectively distinguish between diseased and healthy lemon leaves. It has also been observed that data augmentation does not always provide performance improvement. In future studies, it is predicted that it will be appropriate to evaluate the effect of data augmentation more effectively by applying deep transfer learning methods to plants with different class numbers.

***Keywords:*** *Data Augmentation, Image Processing, Lemon Leaf Diseases, Transfer Learning*

## Limon Yaprağı Görüntü Sınıflandırmasında Transfer Öğrenme ve Veri Artırımın Etkinliği

***Öz.*** Ağaç ve bitkilerde yaprak hastalıkları, tarımsal ürünlerin verimini doğrudan etkileyen önemli faktörlerdir. Bu sorun, çiftçilerin üretim kapasitelerinin ve karlılık düzeylerinin düşmesine neden olabilmektedir. Bu nedenle bitki hastalıklarını doğru bir şekilde tespit edebilmek için bilgisayar destekli tespit ve sınıflandırma sistemlerine ihtiyaç duyulmaktadır. Son yıllarda öğrenme algoritmaları ve görüntü işleme teknikleri tarım sektöründe etkin bir şekilde kullanılmaktadır. Bu çalışmada, limon yaprağı görüntülerinden oluşan bir veri kümesi üzerinde transfer öğrenme ve veri artırma yöntemlerinin etkinliği incelenerek hastalıklı ve sağlıklı limon yaprağı görüntüleri sınıflandırılması işlemi yapılmaktadır. Çalışmamızda VGG16, ResNet50 ve DenseNet201 transfer öğrenme yöntemleri hem veri artırımlı hem de artırımsız olarak uygulanmış ve veri artırmanın performansa etkisi değerlendirilmiştir. Kullanılan derin transfer öğrenme yöntemleri arasında en yüksek doğruluk oranını %98,29 ile DenseNet201 vermiştir. Gerçekleştirilen bu çalışma, transfer öğrenme yöntemlerinin hastalıklı ve sağlıklı limon yapraklarını etkili bir şekilde ayırt edebildiğini göstermektedir. Veri artırmanın her zaman performans iyileşmesi sağlamadığı da gözlemlenmiştir. Gelecekteki çalışmalarda derin transfer öğrenme yöntemleri farklı sınıf sayılarına sahip bitkilerde uygulanarak veri artırmanın etkisinin daha etkili bir şekilde değerlendirilmesinin uygun olacağı öngörülmektedir.

* **Corresponding author:**
E-mail address: asaygili@nku.edu.tr (A.Saygılı)

***Anahtar Kelimeler:*** *Veri Artırımı, Görüntü İşleme, Limon Yaprağı Hastalıkları, Transfer Öğrenme*

## 1. Introduction

Plants protect the ozone layer because they provide food for all terrestrial living organisms and are responsible for filtering out the sun's harmful UV radiation [1]. The Food and Agriculture Organization of the United Nations (FAO) recommends that agricultural production should increase by 70% by 2050 to meet the world's food needs [2]. However, agriculture has been an important source of economic growth for countries. The farmer selects the required product according to the soil type, weather conditions, and economic value of the place. Alternative methods are sought to increase food production due to increasing population and climate changes [3]. Information technologies have a great contribution in this sense. Machine learning and deep learning methods and automatic disease detection systems are mechanisms created for this purpose. Diseases in the leaves of trees and plants are factors that directly affect the yield of agricultural products [4]. This is an important problem that may lead to a decrease in the production capacity and profitability of farmers. The detection of diseases of plants is determined as a result of the visual examinations of the farmers. An error in these determinations can lead to negative results. For this reason, computer-aided detection and classification systems are needed to accurately detect plant diseases. Learning algorithms and image processing methods, which have been actively used in almost all fields in recent years, can also be used effectively in this field. There are many studies done in this area.

Today, deep learning techniques are successfully applied in many areas in the field of image classification. However, the use of these techniques often requires large datasets and may encounter performance problems when we do not have sufficient training data. Transfer learning methods solve or at least alleviate such problems with data augmentation techniques. In this article, we will examine the effectiveness of transfer learning and data augmentation methods on an image dataset consisting of lemon leaves. Transfer learning is the reuse of a previously trained model to solve the task on the target dataset. Data augmentation is a technique that allows the creation of new samples by applying various transformations to the existing data set. In this study, transfer learning methods were used to distinguish the diseased and healthy images of lemon leaves from each other. Transfer learning methods were applied with and without data augmentation, and the effect of data augmentation on success was tried to be evaluated. Models were created using deep transfer learning methods VGG16, ResNet50, and DenseNet201. Therefore, our contribution to this article;

- Models were created with transfer learning methods to classify the disease from lemon leaves.
- The effects of data augmentation on classification success were analyzed.
- DenseNet201, one of the CNN models based on transfer learning, gave a very high accuracy rate for plant leaf disease classification.
- A study has been presented that can be used for early diagnosis of lemon leaf diseases.

The details of our article are explained in the following sections. Chapter 2 shows a brief literature review of different disease detection patterns from plant leaves. Chapter 3 describes the material and method. Then, experimental results and analysis are presented in section 4. Finally, section 5 presents the results of the study and future work.

## 2. Literature Review

There are many studies in the literature using the leaves of plants. The first of these is the study of Subramanian et al. [5]. In this study, Subramanian et al. suggest the use of Deep Learning models based on Transfer Learning for the classification of lemon leaf diseases. According to the results obtained in the study, it is stated that the use of Deep Learning models based on transfer learning can be an effective and cost-effective method in the classification of lemon leaf diseases. Among the models used, Xception gave the highest accuracy rate with 94.34%. Banni et al. [6] proposed a model using GLCM algorithms to detect citrus leaf disease. The accuracy rate obtained in the study is around 85.71%. In the study of Sardogan et al. [7], a method is proposed to detect and classify tomato leaf diseases using Convolutional Neural Network (CNN) and Learning Vector Quantification (LVQ) algorithms. Tomato leaves are divided into four different classes: bacterial spot, late blight, Septoria leaf spot, and yellow curly leaf diseases. The average success rate for all leaf varieties was 86%.

Rastogi et al. [8] propose a method for the automatic detection and grading of leaf diseases in agriculture using digital image processing and machine vision technology. They evaluated the proposed system in two stages. In the first stage, plant recognition is carried out according to the characteristics of the leaf, which includes pre-processing of leaf images, feature extraction, and Neural Network-based training and classification for leaf recognition. In the second stage, segmentation of the diseased area on the leaf with the K-Means method, feature extraction of the defective part, and classification of the disease with the ANN method is performed. In the study of Padol et al. [9], a grape leaf disease detection system with an SVM classification method is suggested. In the study, first of all, the diseased area is segmented with K-means. Color and texture properties are then extracted. Finally, the SVM classification technique is used to detect the type of leaf disease. The accuracy rate of the proposed system was 88.89%. Ahmed et al. In their study [10], they present a system for detecting three common diseases (leaf spot, bacterial leaf blight, and brown spot diseases) in rice plants using machine learning techniques. After the preprocessing step, the dataset is classified by several different machine learning algorithms including KNN, J48, Naive Bayes, and Logistic Regression. When the decision tree algorithm was applied to the test data set after 10 times cross-validation, an accuracy rate of over 97% was obtained.

Agarwal et al. [11] propose a deep learning-based approach to detect and classify tomato leaf diseases using Convolutional Neural Network (CNN). Experimental results show that the proposed model outperforms pre-trained models such as VGG16, InceptionV3, and MobileNet, with a classification accuracy ranging from 76% to 100% for different classes and an average accuracy of 91.2% for 9 diseases and 1 healthy. Irmak and Saygılı [12] discuss the use of convolutional neural networks (CNNs) for the automatic classification of major tomato leaf diseases that significantly affect tomato efficiency. In the study, a CNN model is proposed for the classification of tomato leaves. As a result of the study, a test accuracy of 97.05% was obtained. Kurmi et al. [13] performed disease detection on 3 different plants (bell pepper, potato, and tomato) using the PlantVillage dataset. Success rates with CNN are 95.80%, 94.10 and 92.60, respectively. Chen et al. [14] aimed to detect disease from potato leaves using the same data set. They also achieved a success rate of 97.73% in the study where they used the MobileNet V2 method. Elfatimi et al. [15] detected disease from bean leaves in their study on the data set they obtained from the Kaggle platform. The success rate obtained by using the MobileNet method in the study is 97.00%.

**3. Material and Methods**

The dataset used in our study is a dataset consisting of diseased and healthy plant images on the Kaggle platform [16]. In this data set, as seen in Figure 1, there are diseased and healthy leaf images of different plants. In our study, diseased and healthy image data of lemon images were used. The lemon dataset contains 159 images of healthy leaves and 77 images depicting diseased leaves.

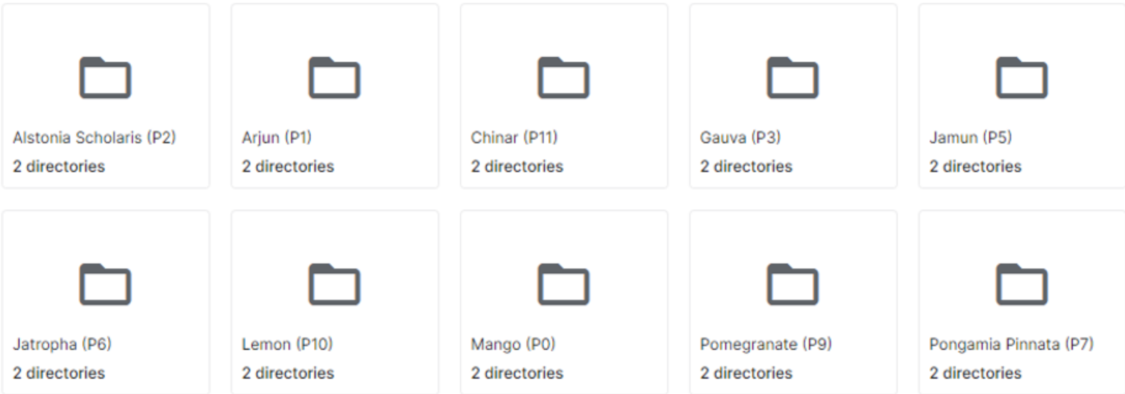


***Fig. 1.*** *Plant species included in the data set*

Examples of diseased and healthy images in the data set are shown in Figure 2. It can also be seen with the naked eye that there are various differentiations on the diseased leaves.

In our study, the classification process was carried out on the lemon dataset by using the transfer learning methods VGG16, ResNet50, and DenseNet201. The VGG16 method is a convolutional neural network (CNN) model used in the field of deep learning [17].

VGG16 is a CNN model with 16 layers, as seen in Figure 3. One of the most important features of VGG16 is that it has a very simple architecture. The model consists of successive layers and mostly uses small 3x3 filters. VGG16 contains 13 convolutional layers, followed by 3 fully connected layers. Fully connected layers perform the classification. VGG16 is trained on the ImageNet dataset. ImageNet is a dataset of millions of images and includes many different classes of objects. During the training of VGG16, it was aimed to extract the features of these images. VGG16 is often used in image classification tasks. It performs particularly well in tasks such as object recognition and classification. It can also be used with transfer learning methods, i.e. it can be adapted to a different dataset by taking the previously trained network of VGG16 and retraining its last layers for another task. It was used in this way in our study as well.

ResNet50 is short for Residual Network and is a type of convolutional neural network (CNN) introduced by He Kaiming et al. in 2015 [18]. As shown in Figure 4, ResNet50 is a 50-layer convolutional neural network (48 convolutional layers, a MaxPool layer, and an average pool layer). The key difference of ResNet50 from other models is an innovative way to add more convolution layers to a CNN without falling into the vanishing gradient problem, using a concept called short-cut connections. A shortcut link bypasses some layers and turns a normal network into a residual one. In this way, ResNet50 can maintain its performance during training while creating a deeper network. In addition, ResNet50 can perform better than other models, although it has fewer parameters.

DenseNet201 [19], shown in Figure 5, is a 201-layer convolutional neural network (CNN). The most important difference of DenseNet201 from other models is an artificial neural network called Dense Convolutional Network (DenseNet), which connects each layer to a feed-oriented mode. The main idea behind DenseNet is the dense connectivity model, where each layer receives input from all previous layers and passes its feature maps to all subsequent layers. This dense connectivity facilitates feature reuse and improves the flow of information across the network. DenseNet-201 specifically refers to the DenseNet variant with 201 layers, including convolutional, pooling, heap normalization, and fully connected layers.

The number of layers and parameters of these three transfer learning methods are shown in Table 1.

Diseased Healthy

*Fig. 2. Diseased and healthy images in the lemon dataset*

**Table 1** Layer and parameter numbers of the methods

| Model | VGG16 | ResNet-50 | DenseNet201 |
|---|---|---|---|
| Number of Layers | 16 | 50 | 201 |
| Number of Parameters (in million) | 138 | 23 | 20 |

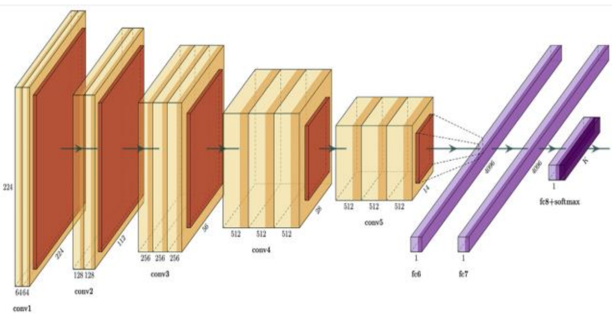


*Fig. 3. VGG16 layers [20]*

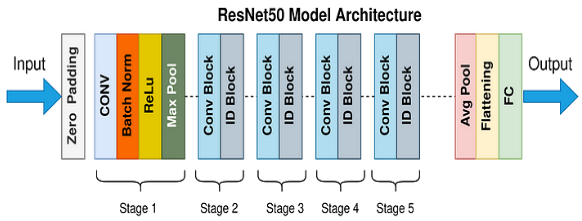


*Fig. 4. ResNet50 layers [21]*

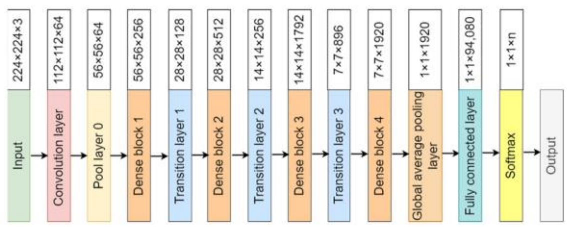


*Fig. 5. DenseNet201 layers [22]*

Since the effect of augmentation on success was investigated in the study, the effect of augmentation on success was examined by applying Reflection, Translation, Scale, and Rotation. In our study, the steps performed in the augmentation process on the MATLAB platform are seen in the code snippet below.

**Table 2** Image Augmentation Code:

```
pixelRange = [-30 30];
scaleRange = [0.9 1.1];
rotationRange = [0 360];
imageAugmenter = imageDataAugmenter( ...
RandXReflection', true, ...
'RandXTranslation',pixelRange, ...
'RandYTranslation',pixelRange, ...
'RandXScale',scaleRange, ...
'RandYScale',scaleRange, ...
'RandRotation', rotationRange)
```

- pixelRange = [-30 30]; This sets the spacing for random scrolling in the x and y directions. The image will be randomly shifted between -30 and 30 pixels in both directions.
- scaleRange = [0.9 1.1]; This sets the random scaling range in the x and y directions. The image will be scaled randomly between 90% and 110% of its original size.
- rotationRange = [0 360]; This sets the interval for random spins. The image will be randomly rotated between 0 and 360 degrees.
- 'RandXReflection', true, This means the image can be randomly mirrored (flipped) along the x-axis.
- 'RandXTranslation',pixelRange, This sets the range to the previously defined 'pixelRange' for random scrolling in the x direction.
- 'RandYTranslation',pixelRange, This sets the range to the previously defined 'pixelRange' for random scrolling in the y direction.
- 'RandXScale',scaleRange, This sets the random scaling range in the x direction to the previously defined `scaleRange`.
- 'RandYScale',scaleRange, This sets the random scaling range in the y direction to the previously defined `scaleRange`.
- 'randRotation', rotateRange); This sets the range to the predefined 'rotationRange' for random spins.

In summary, as a result of the code snippet given above, it will create an "imageDataAugmenter" object that can be randomly flipped, scaled, rotated, and mirrored. This is an efficient method used to augment the image dataset in deep learning applications.

**4. Experimental Results**

In our study, Accuracy, Sensitivity, Specificity, and F1 Score metrics were used for performance measurement. These metrics are calculated in Formulas (1), (2), (3), and (4). In the formulas, TP refers to correctly predicted positives, TN refers to correctly predicted negatives, FP refers to incorrectly predicted positives, and FN refers to incorrectly predicted negatives.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} * 100 \quad (1)$$

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})} * 100 \quad (2)$$

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})} * 100 \quad (3)$$

$$\text{F1 Score} = \frac{2\text{TP}}{(2\text{TP} + \text{FP} + \text{FN})} \quad (4)$$

Accuracy, sensitivity, specificity, and F1 score are commonly used performance measures to evaluate the effectiveness of classification models. Accuracy measures the overall accuracy of the model's predictions by calculating the ratio of correctly classified samples to the total number of samples in the dataset. Provides an overview of the model's performance. Accuracy is useful when the dataset is balanced. However, it can be misleading when the dataset is unbalanced, as the model can achieve high accuracy by only predicting the majority class. Sensitivity measures the model's ability to accurately identify positive samples. Calculates the ratio of true positive estimates to the total number of true positive samples. Precision is especially important when the cost of false negatives is high. For example, in medical diagnosis, it is crucial to accurately identify individuals with a disease to provide timely treatment. Specificity measures the model's ability to accurately identify negative samples. Calculates the ratio of true negative estimates to the total number of true negative samples.

Specificity is important when the cost of false positives is high. For example, in airport security, accurate identification of innocuous objects as negative can help prevent unnecessary delays and inconvenience for passengers. The F1 score is a harmonic mean of precision and recall (precision). It provides a single measurement that balances both measurements. It is particularly useful when there is an imbalance between classes in the dataset. The F1 score ranges from 0 to 1; where 1 indicates the best possible model performance. Together, these metrics provide a comprehensive assessment of the model's performance (F1 score), taking into account different aspects such as overall accuracy, ability to identify positive patterns (sensitivity), ability to identify negative patterns (specificity), and a balanced measure of precision and recall.

In our study, the 5-fold cross-validation test, the details of which are shown in Figure 6, was applied in the classification process. The results of the metrics obtained in each fold of the cross-validation test are given in Table 3 for VGG16, Table 4 for ResNet50, and Table 5 for DenseNet201.


***Fig. 6.*** 5-Fold Cross Validation Schema

Looking at the results obtained in Tables 3, 4, and 5, it is seen that different values can be obtained in the same method for each fold. The reason for this is that the randomly determined training and test subsamples in each fold are different. For this reason, in such studies, it is important for the reliability and accuracy of the results to be cross-validated, such as 5-fold or 10-fold, instead of separating the data as 30%-70% test and training data.

Table 6 shows the results of the transfer learning methods used in our study, with and without data augmentation. When the results in the table are evaluated, it does not seem possible to say that data augmentation should be applied or it should not be applied. Because when we look at the Accuracy rates, it is seen that the success of the three methods applied decreases when the data is increased. When the sensitivity values are examined, it is seen that the success rate of 74.99% in the VGG16 method increased to 86.87% after the data increase. In ResNet50 and DenseNet201, it is observed that data increase decreases the Sensitivity value. Looking at the F1 Score metric, which produces more realistic results in unbalanced data sets, it is observed that data augmentation in VGG16 and ResNet50 methods increases success. In DenseNet201, on the other hand, the success decreased as a result of the data increase.

Data augmentation does not always guarantee an increase in success in transfer learning methods. While data augmentation is a powerful technique that can help improve generalization and performance in many situations, its effectiveness depends on several factors and there are cases where it may not lead to a significant improvement. In certain situations, aggressive data augmentation can lead to overfitting, especially when the target task has relatively small amounts of data. Overfitting occurs when the model learns to memorize augmented examples rather than understanding underlying patterns. Highly complex models may have sufficient capacity to learn from limited original data without requiring extensive data augmentation. In such cases, augmentation may not yield significant benefits. If the target task data is unbalanced, data augmentation may not be able to deal with this problem effectively and may even exacerbate the class instability problem.

Since the numbers of diseased and healthy images in the data set do not show a balanced distribution, it is thought that it is more accurate to evaluate according to the F1 Score among the metrics. As can be seen from Table 6, the most successful F1 Score of 0.98 was obtained with the DenseNet201 method. The lowest results in F1 score values were obtained in the VGG16 method.

Table 8 shows the time complexity of the 3 different methods used in the study. As seen in the table, ResNet50 was the method that performed the transactions in the shortest time. One of the most important reasons for this is the number of layers and parameters given in Table 1. ResNet50 was the fastest method with 23 million parameters and 50 layers.

**Table 3** VGG16 Performance Measurement Results

| | **VGG16 no Augmentation** | | | | **VGG16 with Augmentation** | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc. | Sen. | Spe. | F1 | Acc. | Sen. | Spe. | F1 |
| Fold1 | 44,68 | 21,88 | 93,33 | 0,35 | 55,32 | 75,00 | 13,33 | 0,70 |
| Fold2 | 91,49 | 93,75 | 86,67 | 0,94 | 72,34 | 81,25 | 53,33 | 0,80 |
| Fold3 | 91,49 | 87,50 | 100,00 | 0,93 | 68,09 | 100,00 | 0,00 | 0,81 |
| Fold4 | 100,00 | 100,00 | 100,00 | 1,00 | 82,98 | 90,63 | 66,67 | 0,88 |
| Fold5 | 80,85 | 71,87 | 100,00 | 0,84 | 82,98 | 87,50 | 73,33 | 0,88 |
| Mean | 81,70 | 75,00 | 96,00 | 0,81 | 72,34 | 86,88 | 41,33 | 0,81 |

**Table 4** ResNet50 Performance Measurement Results

| | **ResNet50 no Augmentation** | | | | **ResNet50 with Augmentation** | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc. | Sen. | Spe. | F1 | Acc. | Sen. | Spe. | F1 |
| Fold1 | 91,48 | 96,87 | 80,00 | 0,93 | 89,36 | 96,87 | 73,33 | 73,3333 |
| Fold2 | 93,61 | 100,00 | 80,00 | 0,95 | 93,61 | 100,00 | 86,66 | 86,6667 |
| Fold3 | 91,48 | 100,00 | 73,33 | 0,94 | 100,00 | 100,00 | 100,00 | 100,00 |
| Fold4 | 93,61 | 90,62 | 100,00 | 0,95 | 100,00 | 100,00 | 100,00 | 100,00 |
| Fold5 | 93,61 | 100,00 | 80,00 | 0,95 | 100,00 | 100,00 | 100,00 | 100,00 |
| Mean | 92,76 | 97,50 | 82,66 | 0,94 | 96,59 | 99,37 | 92,00 | 92,00 |

**Table 5** DenseNet201 Performance Measurement Results

| | **DenseNet201 no Augmentation** | | | | **DenseNet201 with Augmentation** | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc. | Sen. | Spe. | F1 | Acc. | Sen. | Spe. | F1 |
| Fold1 | 93,61 | 100,00 | 80,00 | 0,95 | 89,36 | 96,87 | 73,33 | 0,92 |
| Fold2 | 97,87 | 100,00 | 93,33 | 0,98 | 89,36 | 100,00 | 66,66 | 0,92 |
| Fold3 | 100,00 | 100,00 | 100,00 | 1,00 | 95,74 | 100,00 | 86,66 | 0,96 |
| Fold4 | 100,00 | 100,00 | 100,00 | 1,00 | 87,23 | 96,87 | 66,66 | 0,91 |
| Fold5 | 100,00 | 100,00 | 100,00 | 1,00 | 95,74 | 100,00 | 86,66 | 0,96 |
| Mean | 98,29 | 100,00 | 94,66 | 0,98 | 91,48 | 98,75 | 76,00 | 0,94 |

**Table 6** Performance Measurement Results of All Methods

| Method | Augmentation | Accuracy | Sensitivity | Specificity | F1 Score |
|---|---|---|---|---|---|
| **VGG16** | No | 81,70 | 74,99 | 96,00 | 0,81 |
| **VGG16** | Yes | 72,34 | 86,87 | 41,33 | 0,81 |
| **ResNet50** | No | 96,59 | 99,37 | 92,00 | 0,92 |
| **ResNet50** | Yes | 92,76 | 97,50 | 82,66 | 0,94 |
| **DenseNet201** | No | 98,29 | 100,00 | 94,66 | 0,98 |
| **DenseNet201** | Yes | 91,48 | 98,75 | 76,00 | 0,94 |

**Table 7** Comparing similar studies in the literature

| Study | Image Type | Method(s) | Accuracy Rate (%) |
|---|---|---|---|
| [5] | Lemon | Xception | 94.34 |
| [6] | Citrus | GLCM | 85.71 |
| [7] | Tomato | Convolutional Neural Network (CNN) and Learning Vector Quantification (LVQ) algorithms | 86.00 |
| [9] | Grape | K-means<br>SVM | 88.89 |
| [10] | Rice | KNN, J48, Naive Bayes, and Logistic Regression | 97.00 |
| [11] | Tomato | VGG16, InceptionV3, and MobileNet, | 91.2 |
| [12] | Tomato | CNN | 97.05 |
| This Study | Lemon | VGG16, ResNet50, and DenseNet201 | 98.29 |

**Table 8** Time complexity of the methods

| Method | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
|---|---|---|---|---|---|
| **VGG16** | 28 Min 56 Sec | 26 Min 55 Sec | 27 Min 6 Sec | 27 Min 31 Sec | 26 Min 56 Sec |
| **ResNet50** | 19 Min 14 Sec | 17 Min 42 Sec | 17 Min 28 Sec | 17 Min 29 Sec | 17 Min 30 Sec |
| **DenseNet201** | 34 Min 3 Sec | 26 Min 47 Sec | 27 Min 45 Sec | 31 Min 43 Sec | 30 Min 37 Sec |

## 5. Conclusion and Discussion

In our study, the classification process was carried out with three different deep transfer learning methods (VGG16, ResNet50, and DenseNet201) using lemon leaf images. While performing these processes, the results were also evaluated by increasing the data. The results obtained in the lemon leaf data set did not reveal a clear finding for data augmentation. While some methods have more successful results with data augmentation, more successful results have been obtained in some methods without data augmentation. In the deep transfer learning methods used, the highest accuracy rate was obtained with the DenseNet201 method, with an accuracy rate of 98.29% and an F1 Score of 0.98. Among the three methods used, the lowest measurement metrics were obtained with the VGG16 method. As a result of the study, it has been seen that transfer learning methods can distinguish diseased and healthy lemon leaves at a high rate. It is also one of the points obtained as a result that data augmentation does not always increase success.

When Table 7 is examined, it is seen that the leaves of different types of plants are classified in the studies carried out in the literature. In Table 7, it is seen that deep learning and transfer learning methods are frequently preferred, however, classical methods continue to be used. When the success rates in Table 7 are examined, it is seen that our study ranks first among the studies compared.

In future studies, the status of data augmentation can be evaluated by applying more transfer learning methods to different image data. Again, since the binary classification process was performed in this study, it was concluded that it would be appropriate to evaluate the situation of data augmentation in classification probes belonging to more than two classes in future studies.

### Author Contribution

Data curation – Ahmet Saygılı (AS); Formal analysis - AS; Investigation - AS; Experimental performance - AS; Data collection - AS; Processing - AS; Literature review - AS; Writing - AS; Review and editing - AS.

### Declaration of Competing Interest

The authors declared no conflicts of interest with respect to the research, authorship, and/or publication of this article.

### References

[1] Ahmad, I., Hamid, M., Yousaf, S., Shah, S. T., Ahmad, M. O. (2020). Optimizing pretrained convolutional neural networks for tomato leaf disease detection, *Complexity*, vol. 2020, 1-6.

[2] FAO. "A Third More Mouths to Feed, FAO, Roma, Italy." http://www.fao.org/news/story/en/item/35571/icode/, Access Date: 02.06.2023.

[3] Sujatha, R., Chatterjee, J. M., Jhanjhi, N., Brohi, S. N. (2021). Performance of deep learning vs machine learning in plant leaf disease detection, *Microprocessors and Microsystems*, vol. 80, p. 103615.

[4] Liliane, T. N. & Charles, M. S. (2020). Factors affecting yield of crops, *Agronomy-climate change & food security*, p. 9, 2020.

[5] Subramanian, B., Jayashree, S., Kiruthika, S., Miruthula, S. (2019), Lemon leaf disease detection and classification using SVM and CNN, *International Journal of Recent Technology and Engineering*, vol. 8, no. 4, pp. 11485-11488.

[6] Banni, R. & Sksvmacet, L. (2018), Citrus leaf disease detection using image processing approaches, *International Journal of Pure and Applied Mathematics*, vol. 120, no. 6, pp. 727-735.

[7] Sardogan, M., Tuncer, A., Ozen, Y. (2018) Plant leaf disease detection and classification based on CNN with LVQ algorithm, in 2018 3rd international conference on computer science and engineering (UBMK), 2018: IEEE, pp. 382-385.

[8] Rastogi, A., Arora, R., Sharma, S. Leaf disease detection and grading using computer vision technology & fuzzy logic, in 2015 2nd international conference on signal processing and integrated networks (SPIN), 2015: IEEE, pp. 500-505.

[9] Padol, P. B. & Yadav, A. A., SVM classifier based grape leaf disease detection, in 2016 Conference on advances in signal processing (CASP), 2016: IEEE, pp. 175-179.

[10] Ahmed, K., Shahidi, T. R., Alam, S. M. I., Momen, S. Rice leaf disease detection using machine learning techniques, in 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 2019: IEEE, pp. 1-5.

[11] Agarwal, M., Singh, A., Arjaria, S., Sinha, A., Gupta, S. (2020) ToLeD: Tomato leaf disease detection using convolution neural network, *Procedia Computer Science*, vol. 167, pp. 293-301.

[12] Irmak, G., Saygili, A. (2020). Tomato leaf disease detection and classification using convolutional neural networks," in 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 2020: IEEE, pp. 1-5.

[13] Kurmi, Y., Saxena, P., Kirar, B. S., Gangwar, S., Chaurasia, V., Goel, A. (2022), Deep CNN model for crops' diseases detection using leaf images, *Multidimensional Systems and Signal Processing*, vol. 33, no. 3, pp. 981-1000.

[14] Chen, W., Chen, J., Zeb, A., Yang, S., Zhang, D. (2022). Mobile convolution neural network for the recognition of potato leaf disease images, *Multimedia Tools and Applications*, vol. 81, no. 15, pp. 20797-20816.

[15] Elfatimi, E., Eryigit, R., Elfatimi, L. (2022). Beans leaf diseases classification using MobileNet models, *IEEE Access*, vol. 10, pp. 9471-9482, 2022.

[16] Anonymous. Healthy vs. Diseased Leaf Image Dataset (Public Domain). https://www.kaggle.com/datasets/amandam1/healthy-vs-diseased-leaf-image-dataset, Access Date: 05.05.2023.

[17] Simonyan, K. &. Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[18] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778.

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

[20] Sahinbas, K. &. Catak, F. O. (2021). Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images," in Data science for COVID-19: Elsevier, pp. 451-466.

[21] S. Mukherjee. "ResNet50." https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758, Access Date: 05.06.2023.

[22] Sharma, N., Saba, L., Khanna, N. N., Kalra, M. K., Fouda, M. M., Suri, J. S. (2022). Segmentation-Based Classification Deep Learning Model Embedded with Explainable AI for COVID-19 Detection in Chest X-ray Scans, *Diagnostics*, vol. 12, no. 9, p. 2132, 2022.