DeepEMPR: coffee leaf disease detection with deep learning and enhanced multivariance product representation

Ahmet Topal[1], Burcu Tunga[1] and Erfan Babaee Tirkolaee[2,3,4]

[1] Department of Mathematics Engineering, Istanbul Technical University, Istanbul, Turkey
[2] Department of Industrial Engineering, Istinye University, Istanbul, Turkey
[3] Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan, Taiwan
[4] Department of Mechanics and Mathematics, Western Caspian University, Baku, Azerbaijan

ABSTRACT

Plant diseases threaten agricultural sustainability by reducing crop yields. Rapid and accurate disease identification is crucial for effective management. Recent advancements in artificial intelligence (AI) have facilitated the development of automated systems for disease detection. This study focuses on enhancing the classification of diseases and estimating their severity in coffee leaf images. To do so, we propose a novel approach as the preprocessing step for the classification in which enhanced multivariance product representation (EMPR) is used to decompose the considered image into components, a new image is constructed using some of those components, and the contrast of the new image is enhanced by applying high-dimensional model representation (HDMR) to highlight the diseased parts of the leaves. Popular convolutional neural network (CNN) architectures, including AlexNet, VGG16, and ResNet50, are evaluated. Results show that VGG16 achieves the highest classification accuracy of approximately 96%, while all models perform well in predicting disease severity levels, with accuracies exceeding 85%. Notably, the ResNet50 model achieves accuracy levels surpassing 90%. This research contributes to the advancement of automated crop health management systems.

Subjects Artificial Intelligence, Computer Vision, Data Mining and Machine Learning, Data Science, Neural Networks
Keywords Plant disease, Deep learning, Enhanced multivariance product representation, High dimensional model representation

Submitted 20 June 2024
Accepted 20 September 2024
Published 13 November 2024

Corresponding author
Ahmet Topal, atopal@itu.edu.tr

Academic editor
Bilal Alatas

Additional Information and Declarations can be found on page 21

DOI 10.7717/peerj-cs.2406

INTRODUCTION

The agricultural industry serves as a cornerstone of many economies worldwide, profoundly impacting the growth and prosperity of communities (*Pranto et al., 2021*). As the global population continues to rise, ensuring sustainable and efficient agricultural practices becomes more critical than ever. However, plant diseases significantly threaten to agricultural productivity and sustainability by compromising crop quality and yield. Despite existing crop protection measures, plant diseases still lead to substantial yield losses, estimated at 20% to 40% (*Richard, Qi & Fitt, 2022*). To address the challenges posed by plant diseases, there is a growing imperative to develop artificial intelligence (AI)-based applications capable of accurately and promptly diagnosing these ailments. Such applications represent a forward-thinking approach to agricultural technology, offering

How to cite this article Topal A, Tunga B, Tirkolaee EB. 2024. DeepEMPR: coffee leaf disease detection with deep learning and enhanced multivariance product representation. *PeerJ Comput. Sci.* **10**:e2406 http://doi.org/10.7717/peerj-cs.2406

smarter and more efficient means of diagnosing, monitoring, and managing plant health issues. By leveraging sophisticated deep learning (DL) models, these AI systems simplify the complex process of disease identification, empowering farmers and agronomists with timely insights into the specific pathogens affecting their crops. Recent advancements in AI have paved the way for automated systems designed to effectively manage crop health. These systems harness the power of DL to analyze vast amounts of data, thereby offering actionable recommendations for disease control and prevention. Consequently, the integration of AI technology holds tremendous potential in boosting crop yields and ensuring agricultural sustainability.

Recently, DL and machine learning (ML) have taken the forefront in the field of innovative AI solutions (*Ghadi et al., 2023*). Particularly in the field of image processing, ML has made significant strides through its sophisticated recognition and classification abilities. Traditional ML algorithms extract features from digital images and analyze these features to enhance their predictive capabilities. They can automatically identify patterns and relationships present in the image data. However, a limitation of traditional ML is its heavy reliance on the quality and relevance of manually engineered features. This reliance on manual feature engineering means that the success of ML algorithms in image processing is contingent upon the expertise of the humans who design and tune them.

On the other hand, DL, as a subfield of ML, possesses the ability to achieve high-performance modeling by utilizing the multi-layered structure of artificial neural networks. Convolutional neural networks (CNNs), as one of the DL architectures, are state-of-the-art tools for understanding the complexity and hidden motifs within image data (*Ölçer, Ölçer & Sümer, 2023*). Another aspect contributing to the popularity of CNNs is their ability to learn feature representations automatically, unlike traditional algorithms, which require manual feature extraction. Therefore, they are extensively used across various computer vision applications, such as classification, detection, and segmentation.

In recent years, research has increasingly focused on employing DL and ML techniques to recognize and diagnose plant diseases. This trend underscores the growing recognition of the potential of these advanced technologies in revolutionizing agricultural practices. In traditional ML-based studies, the system being proposed roughly consists of three key stages: preprocessing, feature extraction, and classification. *Neelakantan (2021)* collected a balanced dataset of 1,090 images to conduct research on the classification of diseases on tomato leaves. All images were annotated into the healthy class and four disease-related classes: septoria leaf spot, leaf curl, bacterial spot, and early blight. The performance analyses of supervised learning algorithms, including support vector machine (SVM) (*Vapnik, 2013*), K-nearest neighbor (*Cover & Hart, 1967*), decision tree (DT) (*Quinlan, 1986*), random forest (RF) (*Breiman, 2001*), and naive Bayes(NB), were evaluated based on the accuracy metric. Experimental findings indicated that RF was the top-performing classifier, achieving an accuracy of 89%. *Ramesh et al. (2018)* used the histogram of oriented gradient (HOG) (*Dalal & Triggs, 2005*) for the feature extraction phase in the binary plant disease classification and then implemented the RF, an ensemble classifier, during the training process. *Prajapati, Shah & Dabhi (2017)* designed a system that leverages the techniques of ML and image processing to enable the automated identification of diseased rice leaves. In the preprocessing step, the image backgrounds were removed using the S component of the image in the HSV domain. To eliminate bias during feature extraction, the diseased, non-diseased, and background segments were clustered using K-means (*MacQueen, 1967*). They then derived the features related to color, texture, and shape from those segments identified as diseased. During the classification step, the researchers experimented with three SVM models, each equipped with a different number of features, to study the impact of the parameters. Furthermore, ML research focused on different plant species, such as corn (*Panigrahi et al., 2020*), grape (*Kaur, Pannu & Malhi, 2019*), potato (*Singh & Kaur, 2021*), and rice (*Ahmed et al., 2019*), offers a broad spectrum of insights.

Mohanty, Hughes & Salathé (2016) performed a study on the recognition of 26 plant diseases using a publicly available database containing leaf images from 14 different plant species. By combining four main groups—deep learning architectures, training mechanisms, dataset types, and training-test set distributions—in various ways, they created and tested 60 different experimental configurations. Their results exhibited exceptional performance, with success rates reaching up to 99.35% in automated identification. A similar research was conducted by *Ferentinos (2018)* on a larger number of diseases and different plant species. Its strength was that it does not consider only laboratory-condition images but also includes real-condition images in the experimental phase. The results were exceptionally promising as the most successful model architecture attained a success rate of 99.53%. *Too et al. (2019)* adopted deep CNNs to create a model for recognizing plant diseases through leaf images. They built their model to identify 38 different classes including images of diseased and healthy leaves from 14 different plants. Their results showed that DenseNets were able to efficiently detect leaf diseases and classify test data with an accuracy of 99.75%. *Agarwal et al. (2020)* suggested a DL model that trains on tomato leaf images sourced from the Plant Village database (*Too et al., 2019*). The proposed architecture consists of three convolutional layers, with a max-pooling layer after each, and then two fully connected layers. The authors noted a significant finding that their model exhibited superior performance when compared to well-known CNN models like VGG16 (*Simonyan & Zisserman, 2014*), InceptionV3 (*Szegedy et al., 2016*), and MobileNet (*Howard et al., 2017*). *Nafi & Hsu (2020)* highlighted the minority of samples in the infected class, addressing the issue of class imbalance. In a comparative study by *Sujatha et al. (2021)*, they indicated that DL techniques surpassed ML techniques in terms of classification accuracy. They also pointed out that among the tested models, RF had the lowest classification accuracy at 76.8%, while VGG16 achieved the highest at 89.5%. *Shrivastava et al. (2019)* presented an approach integrating ML and DL, where AlexNet (*Krizhevsky, Sutskever & Hinton, 2012*), a deep neural network, acted as the feature extractor, and the SVM acted as the classifier. The entire dataset was partitioned into training and testing sets at varying ratios, specifically 80%–20%, 70%–30%, and 60%–40%. In the experimental study, it was outlined how the different data partitions impacted the performance of the predictive model. *Esgario, Krohling & Ventura (2020)* made a major contribution to the coffee farming industry with automated disease diagnosis. They worked on both disease classification and disease severity classification. Their results showed that they almost perfectly detected the disease in coffee plants, but did not meet expectations in determining the disease's severity.

Motivation for the study

Plant diseases present a major risk to agricultural productivity and sustainability by adversely affecting crop quality and yield. Despite current crop protection measures, plant diseases continue to cause significant yield losses. Furthermore, traditional methods for identifying leaf diseases rely heavily on manual inspection and expert knowledge, which can be time-consuming, costly, and prone to error. Additionally, existing automated techniques often struggle with accuracy due to the complex and variable nature of disease symptoms. Despite advancements in ML and computer vision, there remains a significant gap in developing robust, scalable, and accurate methods for disease identification.

Motivation for the choice of methods used

Over the past decade, CNNs have been commonly used to better process visual information. CNNs outperform traditional methods in visual information processing due to their ability to automatically learn hierarchical features from data. Furthermore, enhanced multivariance product representation (EMPR) is a technique for reducing the dimension by representing an N-direction tensor in the form of lower-dimensional tensors and specific support vectors. At the end of the EMPR process, the image data is separated into its high and low-frequency components. These components can be combined according to the task's requirements to better represent the original images for classification, detection, and similar processes.

Contributions

In this work, we advance the research on coffee leaf disease detection by utilizing deep CNN architectures. Our algorithm integrates two mathematical methods introduced in recent literature, EMPR and high-dimensional model representation (HDMR). HDMR represents high-dimensional functions as combinations of lower-dimensional functions, such as univariate and bivariate functions, along with their interactions. It is a powerful tool for analyzing high-dimensional datasets, simplifying the analysis process, and revealing underlying relationships and interactions within the data. EMPR, derived from the HDMR framework, is specifically designed for representing multivariate functions and tensor-type datasets (*Şen & Tuna, 2023*). It extends the capabilities of HDMR to handle complex multivariate functions and adapt to data structures characterized by multiple dimensions or variables. EMPR achieves this by employing a more specific decomposition approach, enhancing the representation of high-dimensional data in terms of simpler, less-variate structures (*Tuna et al., 2020*; *Tuna & Tunga, 2013*). The use of these mathematical methods has enabled us to achieve superior results in coffee leaf disease detection compared to previous studies (*Esgario, Krohling & Ventura, 2020*).

Our algorithm is founded on the creation of a new channel for each original image. This involves incorporating a series of EMPR components obtained from the EMPR method (*Tunga & Demiralp, 2010*), which is utilized in various fields in the literature (*Korkmaz Özay & Tunga, 2022*; *Tunga, 2015*). The created channel allows the hybridization of useful details from both high-frequency and low-frequency components. After creating the channel, a contrast enhancement process is performed using an HDMR-based contrast enhancement method (*Tunga & Kocanaogullari, 2017*). Following this, the enhanced channel is added to the original three-channel image, converting the baseline dataset into a dataset of coffee leaves with four channels. CNN architectures are applied to this dataset to accomplish the disease classification and stress severity estimation tasks.

Structure of the manuscript

The remainder of this paper is divided into five main sections. The second section reminds the mathematical background of EMPR concepts. The third section introduces the dataset under consideration. The fourth section elaborates on the methodology proposed in this study, providing detailed insight into the steps that comprise our investigative approach. The fifth section presents the experimental study and its results. The last section includes concluding remarks on the paper.

ENHANCED MULTIVARIANCE PRODUCT REPRESENTATION

EMPR is a technique for reducing the dimension by representing an N-direction tensor in the form of lower-dimensional tensors and specific support vectors. Employing the EMPR method, the representation of the N-dimensional tensor structure $\mathcal{X}$ reveals itself in the following manner:

$$\mathcal{X}_{i_1\ldots i_N} = \mathcal{X}^{(0)}\prod_{j=1}^{N} s_{i_j}^{(j)} + \sum_{j_1=1}^{N}\mathcal{X}_{i_{j_1}}^{(j_1)} \prod_{\substack{j=1\\ j\neq j_1}}^{N} s_{i_j}^{(j)} + \sum_{\substack{j_1,j_2=1\\ j_1<j_2}}^{N} \mathcal{X}_{i_{j_1},i_{j_2}}^{(j_1 j_2)} \prod_{\substack{j=1\\ j\neq j_1,j_2}}^{N} s_{i_j}^{(j)} + \cdots$$
$$+\mathcal{X}^{(j_1 j_2\ldots j_N)}_{i_{j_1} i_{j_2}\ldots i_{j_N}} \qquad (i_j = 1,2,\ldots,n_j, \quad j=1,2,\ldots,N). \tag{1}$$

This expansion involves a finite set of summations. $\mathcal{X}^{(0)}$ is a constant, $\mathcal{X}^{(j_1)}$ represents one-way tensors (or vectors), and $\mathcal{X}^{(j_1 j_2)}$ corresponds to two-way tensors, namely matrices. The components on the right-hand side can be viewed as tensors with an increasing number of dimensions.

The entities $s^{(j)}$ in the expansion are referred to as support vectors. The primary objective of this approach is to determine the fundamental structures of EMPR components by leveraging support vectors, with the aim of representing multivariate data with fewer variate components.

To uniquely obtain the components in the expansion, vanishing conditions are given through the support vectors:

$$\sum_{i_{j_l}=1}^{n_{j_l}} W_{i_{j_l}}^{(j_l)} s_{i_{j_l}}^{(j_l)} \mathcal{X}^{(j_1\ldots j_k)}_{i_{j_1}\ldots i_{j_k}} = 0 \quad (l=1,2,\ldots,k, \quad k=1,2,\ldots,N), \tag{2}$$

where $(W_{i_j}^{(j)})$s are weight vectors and these weight vectors satisfy the normalization conditions given below:

$$\sum_{i_j=1}^{n_j} W_{i_j}^{(j)} = 1, \quad \sum_{i_j=1}^{n_j} W_{i_j}^{(j)} \left(s_{i_j}^{(j)}\right)^2 = 1 \quad (j=1,2,\ldots,N). \tag{3}$$

The constant component of the EMPR, $\mathcal{X}^{(0)}$, is uniquely determined based on the conditions mentioned in Eq. (4):

$$\mathcal{X}^{(0)} = \sum_{i_1=1}^{n_1}\cdots\sum_{i_N=1}^{n_N}\left[\prod_{k=1}^{N} W_{i_k}^{(k)} s_{i_k}^{(k)}\right]\mathcal{X}_{i_1\ldots i_N}. \tag{4}$$

The univariate components represented by $\mathcal{X}^{(j)}$ are computed using Eq. (5) by excluding the pertinent direction:

$$\mathcal{X}_{i_j}^{(j)} = \sum_{i_1=1}^{n_1}\cdots\sum_{i_{j-1}=1}^{n_{j-1}}\sum_{i_{j+1}=1}^{n_{j+1}}\cdots\sum_{i_N=1}^{n_N}\left[\prod_{\substack{k=1\\ j\neq k}}^{N} W_{i_k}^{(k)} s_{i_k}^{(k)}\right]\mathcal{X}_{i_1\ldots i_N} - \mathcal{X}^{(0)} s_{i_j}^{(j)}$$
$$(i_j = 1,2,\ldots,n_j, \quad j = 1,2,\ldots,N). \tag{5}$$

The EMPR's bivariate components are calculated using Eq. (6) in a similar manner, excluding the two related directions:

$$\mathcal{X}_{i_j,i_k}^{(j,k)} = \sum_{i_1=1}^{n_1}\cdots\sum_{i_{j-1}=1}^{n_{j-1}}\sum_{i_{j+1}=1}^{n_{j+1}}\cdots\sum_{i_{k-1}=1}^{n_{k-1}}\sum_{i_{k+1}=1}^{n_{k+1}}\cdots\sum_{i_N=1}^{n_N}\left[\prod_{\substack{k=1\\ j\neq k}}^{N} W_{i_k}^{(k)} s_{i_k}^{(k)}\right]\mathcal{X}_{i_1\ldots i_N}$$
$$-\mathcal{X}_{i_k}^{(k)} s_{i_j}^{(j)} - \mathcal{X}_{i_j}^{(j)} s_{i_k}^{(k)} - \mathcal{X}^{(0)} s_{i_j}^{(j)} s_{i_k}^{(k)}$$
$$(j,k = 1,2,\ldots,N, \quad i_j = 1,2,\ldots,n_j, \quad i_k = 1,2,\ldots,n_k). \tag{6}$$

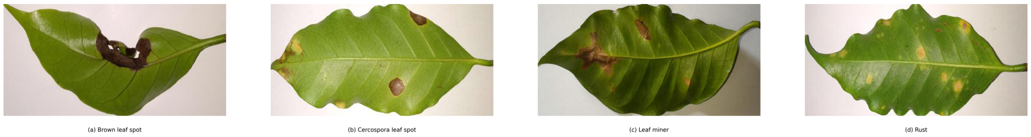
Other high-variate terms in the expansion are determined using the same philosophy.

Identifying the support vectors in the EMPR method is crucial as it directly impacts the method's performance (*Tunga & Demiralp, 2010*; *Tunga, 2015*). While there are no constraints on how support vectors are evaluated, in this study, we chose the support vectors by normalizing the directional averages of the dataset below. The calculation of support vectors is thoroughly outlined in Eq. (7). For more detailed information about support vectors, readers may refer to *Tuna et al. (2020)*, *Şen & Tuna (2023)*, and *Tunga (2015)*.

$$s_{i_j}^{(j)} = \frac{\sum_{i_1=1}^{n_1}\cdots\sum_{i_{j-1}=1}^{n_{j-1}}\sum_{i_{j+1}=1}^{n_{j+1}}\cdots\sum_{i_N=1}^{n_N}\mathcal{X}_{i_1\ldots i_N}}{\left[\sum_{i_j=1}^{n_j}\left[\sum_{i_1=1}^{n_1}\cdots\sum_{i_{j-1}=1}^{n_{j-1}}\sum_{i_{j+1}=1}^{n_{j+1}}\cdots\sum_{i_N=1}^{n_N}\mathcal{X}_{i_1\ldots i_N}\right]^2\right]^{\frac{1}{2}}}. \tag{7}$$

DATASET DESCRIPTION

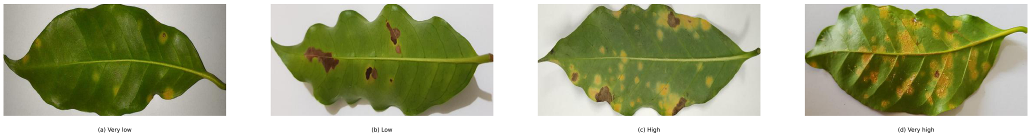
Our study focuses on the recognition of diseases in coffee leaves. We use the Arabica Coffee Leaves dataset (*Esgario, Krohling & Ventura, 2020*), which is a collection of high-resolution



(a) Brown leaf spot (b) Cercospora leaf spot (c) Leaf miner (d) Rust

Figure 1 A sample image of each type of disease affecting coffee plant leaves.

Full-size DOI: 10.7717/peerjcs.2406/fig-1



(a) Very low (b) Low (c) High (d) Very high

Figure 2 A sample image from each disease severity class.

Full-size DOI: 10.7717/peerjcs.2406/fig-2

images showcasing various leaf conditions. It equips us with the means to investigate and identify the severity levels and various types of diseases affecting coffee plants. Our objective is to improve the accuracy of disease diagnosis, thereby contributing to the efficient management and treatment in coffee farming.

The leaf images are captured using smartphones at various times throughout the year in Marechal Floriano County, located in the state of Espírito Santo, Brazil (*Esgario, Krohling & Ventura, 2020*). The dataset comprises a total of 1,685 images, categorized into two primary classes: one representing healthy images and the other representing diseased images. Each image has dimensions of 512 pixels in height and 256 pixels in width. It features three color channels providing color information and is represented in the RGB color space. We created two datasets from the original: one to address recognizing biotic stress and the other to assess stress severity. Diseased leaf images are annotated into biotic stress categories, such as leaf miner, rust, brown leaf spot, and Cercospora leaf spot. They are also labeled with different levels of stress severity, including very low, low, high, and very high. Figures 1 and 2 show example images from each class of biotic stress and stress severity, respectively, except for the healthy class.

We split the datasets into training, validation, and test sets, for model training and evaluation. Exactly 70% of the data is designated for the training set, which is used to train our models. A 15% portion of the data is allocated to the validation set for model selection. This validation set serves as a checkpoint to assess the model's performance during training, allowing us to make essential adjustments without impacting the integrity of the test data. It is also crucial for identifying and rectifying overfitting, where a model performs well on training data but poorly on unseen data. Finally, the remaining 15% of the data is reserved for the test set, used exclusively for evaluating the model's performance. The data division process has been conducted under the guidance of the reference research (*Esgario, Krohling & Ventura, 2020*).

The number of images in each class for the datasets on biotic stress and stress severity is presented in Table 1. The table clearly shows that the sample distribution among the

Table 1 **Dataset overview.**

Biotic stress	# of images	Stress severity	# of images
Brown leaf spot	348	Healthy	272
Cercospora leaf spot	147	High	101
Healthy	272	Low	332
Leaf miner	387	Very high	56
Rust	531	Very low	924

classes in both datasets is not uniform, leading to an imbalance between classes with one class containing a greater number of samples compared to the others. From a quantitative perspective, when the number of instances in the majority class (class with a greater number of examples) is divided by the number of instances in the minority class (class with fewer examples), imbalance ratios of 16.5 for severity and 3.61 for biotic stress are observed. This means that the biotic stress dataset is partially imbalanced, and the severity dataset is highly imbalanced (*Topal & Amasyali, 2021*). The high imbalance ratio makes it challenging for the classifier to learn the examples from the minority class. Therefore, it is required to use appropriate performance metrics that are more sensitive to the minority class.

PROPOSED METHODS

The main focus of the study is to develop approaches that will enhance the generalization and learning capacity of the model during the preprocessing stage. An additional channel based on the EMPR will be added to the images. This aims to examine the impact of using more features. Additionally, the contrast of this channel, which will be added to the original images, will be improved to various degrees based on high-dimensional model representation, and the effect of this enhancement will also be researched. The steps to be applied in the study to investigate the impact of these criteria are listed below:

1. The dataset to be used in the study is split into training, validation, and test sets in specific proportions. This splitting is necessary for training the model, validating it, and testing the results.
2. During the preprocessing stage, a method based on EMPR is used to add a new channel to each image. With this method, each image will become four-channel by adding a new channel to the original three-channel image. Additionally, the contrast of the added channel is improved at different levels.
3. Thus, a training set is generated for each level of contrast enhancement.
4. CNN models are trained separately on each generated training set.
5. The performance of the trained models is investigated on the validation set, and the model showing the best performance (*i.e.*, the one with the least loss value) for each level of contrast enhancement is selected.
6. Final performance analyses of the best-performing models are conducted on the test set using various metrics such as accuracy, precision, recall, and F1-score. These analyses are carried out to evaluate the generalization ability and performance of the models.

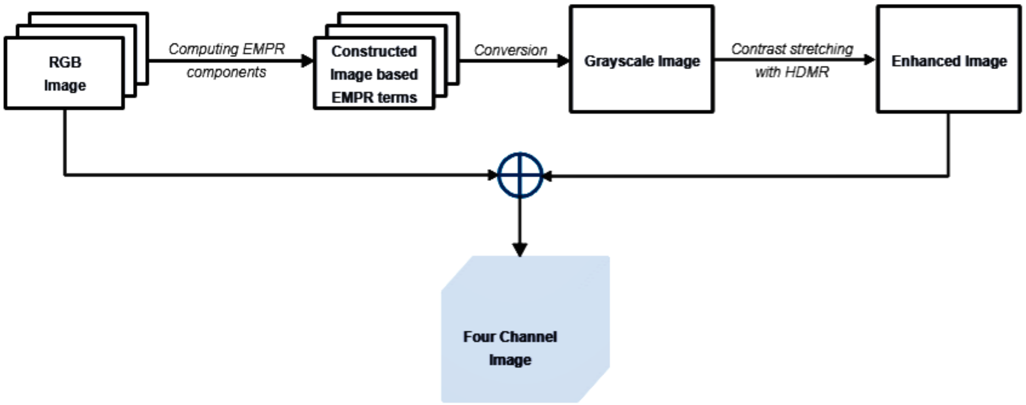

RGB Image
Computing EMPR components
Constructed Image based EMPR terms
Conversion
Grayscale Image
Contrast stretching with HDMR
Enhanced Image
Four Channel Image

Figure 3 **Preprocessing steps.**

Full-size DOI: 10.7717/peerjcs.2406/fig-3

7. Additionally, the ability of the models to distinguish between diseased and healthy classes is investigated using ROC-AUC curves. These evaluations help to demonstrate the classification performance of the model in more detail.

Preprocessing stage

In image classification tasks, the effectiveness of the results largely depends on the robustness of the preprocessing steps applied to the raw data. In our study, the proposed pipeline for preprocessing stages is illustrated in Fig. 3 and additional details are presented through the pseudocode available in Algorithm 1. Our steps prepare the raw image data for further analysis, improving features essential for classification. This approach enables classification algorithms to concentrate on the most important attributes of the data, a key factor in achieving high performance.

The pipeline begins with the computation of EMPR components from the original RGB image, which is represented by a 3-dimensional tensor. In EMPR frameworks, the terms $\mathcal{X}^{(1,2)}_{i_1,i_2}, \mathcal{X}^{(1,3)}_{i_1,i_3}, \mathcal{X}^{(2,3)}_{i_2,i_3}$, and $\mathcal{X}^{(1,2,3)}_{i_1,i_2,i_3}$ can be categorized as high-frequency components, and the terms $\mathcal{X}^{(0)}, \mathcal{X}^{(1)}_{i_1}, \mathcal{X}^{(2)}_{i_2}$, and $\mathcal{X}^{(3)}_{i_3}$ can be categorized as low-frequency components (*Korkmaz Özay & Tunga, 2022*). EMPR-based image is derived through Eq. (8):

$$\mathcal{G} = \mathcal{X}^{(3)}_{i_3} s^{(1)}_{i_1} s^{(2)}_{i_2} + \mathcal{X}^{(1,3)}_{i_1,i_3} s^{(2)}_{i_2} + \mathcal{X}^{(2,3)}_{i_2,i_3} s^{(1)}_{i_1} + \mathcal{X}^{(1,2,3)}_{i_1,i_2,i_3}$$
$$(i_1 = 1,2,\ldots,n_1, i_2 = 1,2,\ldots,n_2, \text{and } i_3 = 1,2,n_3 = 3), \tag{8}$$

where $\mathcal{X}^{(3)}_{i_3}, \mathcal{X}^{(1,3)}_{i_1,i_3}, \mathcal{X}^{(2,3)}_{i_2,i_3}$ and $\mathcal{X}^{(1,2,3)}_{i_1,i_2,i_3}$ denote the EMPR components; $s^{(1)}_{i_1}$ and $s^{(2)}_{i_2}$ denote the support vectors; and n_1, n_2, and n_3 represent the height, width and depth information of the original image, respectively.

It is not constructed to include solely low-frequency or high-frequency terms; it is designed to take advantage of the beneficial properties of both. High-frequency terms $\mathcal{X}^{(1,3)}_{i_1,i_3}, \mathcal{X}^{(2,3)}_{i_2,i_3}$, and $\mathcal{X}^{(1,2,3)}_{i_1,i_2,i_3}$ are used to better capture the edges, fine details and textural information in the image. On the other hand, since high-frequency terms are easily affected by noise, and low-frequency terms are robust against noise, omitting the term $\mathcal{X}^{(1,2)}_{i_1,i_2}$ and using the low-frequency term $\mathcal{X}^{(3)}_{i_3}$ relatively reduces the noise in the resulting

Algorithm 1 Preprocessing Stage

```
Require: $\mathcal{I} \leftarrow$ An RGB Image
Ensure: Four Channel Image
begin
    Find the height ($n_1$), width ($n_2$), and depth ($n_3$) of $\mathcal{I}$.
    Compute optimum support vectors $s_{i_1}^{(1)}$, $s_{i_2}^{(2)}$, and $s_{i_3}^{(3)}$.
    Compute constant ($\mathcal{X}^{(0)}$), univariate ($\mathcal{X}_{i_1}^{(1)}, \mathcal{X}_{i_2}^{(2)}, \mathcal{X}_{i_3}^{(3)}$), bivariate
  ($\mathcal{X}_{i_1,i_2}^{(1,2)}, \mathcal{X}_{i_2,i_3}^{(2,3)}, \mathcal{X}_{i_1,i_3}^{(1,3)}$), and trivariate ($\mathcal{X}_{i_1,i_2,i_3}^{(1,2,3)}$) EMPR terms of $\mathcal{I}$.
    Construct EMPR-based image $\mathcal{G}$.
    Convert $\mathcal{G}$ to the grayscale image $\mathcal{J}_{(g)}$.
    Compute constant, univariate, bivariate HDMR components of $\mathcal{J}_{(g)}$.
    for $i = 1 \rightarrow \alpha$ do
        Add the univariate and bivariate components to the $\mathcal{J}_{(g)}$.
    end for
    Put the enhanced grayscale image as a new channel behind the $\mathcal{I}$.
end
Notes:
                                        ▷ $\alpha$ : degree of contrast enhancement
        ▷ $i_1 = 1, 2, \ldots, n_1$, $i_2 = 1, 2, \ldots, n_2$, and $i_3 = 1, 2, n_3 = 3$
```

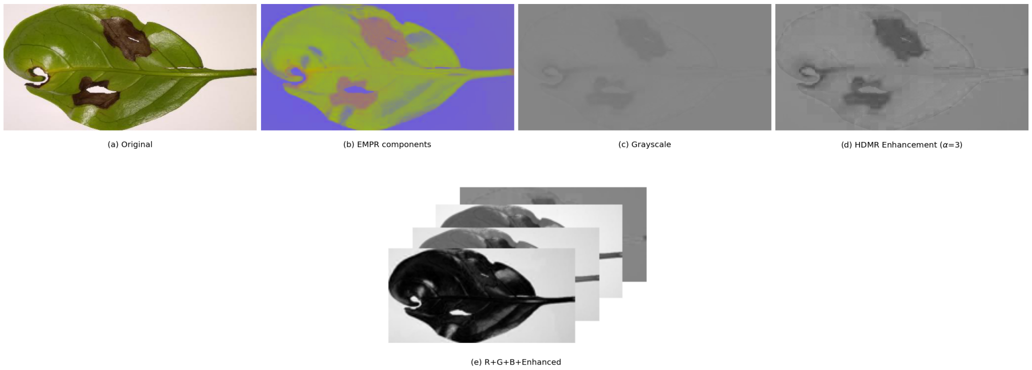
image and offers a gradual transition of colors. The key aspect of using high-frequency EMPR components predominantly is that the high-frequency details significantly improve the discrimination of features, aiding in more accurate and refined object detection, classification, and analysis. Furthermore, the superposition of EMPR terms incorporating the $\{i_3\}$-th position provides clearer visual differentiation between diseased and non-diseased regions in leaf images, as these terms emphasize color information well.

Following the EMPR construction, the image undergoes conversion to grayscale by applying the weights 0.2989 for the red channel, 0.5870 for the green channel, and 0.1140 for the blue channel. This well-known conversion formula is provided in *International Telecommunication Union (2011)*:

$$\mathcal{J}_{(g)} = 0.2989\, \mathcal{G}_{Red} + 0.5870\, \mathcal{G}_{Green} + 0.1140\, \mathcal{G}_{Blue}, \tag{9}$$

where $\mathcal{G}_{Red}, \mathcal{G}_{Green}$, and $\mathcal{G}_{Blue}$ correspond to the red, green, and blue channels of the image $\mathcal{G}$, respectively.

This conversion reduces the image from three color channels (red, green, and blue) to a single channel representing intensity. The pipeline culminates with the creation of an enhanced image that is not only visually more striking but also carries a greater informational load. To produce the enhanced image, the univariate and bivariate HDMR components of the grayscale image are first computed, and then they are added to the grayscale image at a specified level. Finally, this enhanced image is combined with the original RGB data to create a four-channel image. This process integrates both image construction based on EMPR components and the contrast enhancement provided by HDMR.



(a) Original

(b) EMPR components

(c) Grayscale

(d) HDMR Enhancement (α=3)

(e) R+G+B+Enhanced

Figure 4 Visualization of preprocessing steps on an example image.
Full-size DOI: 10.7717/peerjcs.2406/fig-4

Figure 4 shows a comprehensive illustration of the entire process in the pipeline on an example image. In this case, the alpha value for contrast stretching is set to 3.

Classification stage

Over the past decade, the field of CNNs has experienced remarkable evolution and innovation. A major factor in this development is the advent of CNN architectures, each with unique design and functionalities. Many of these architectures gained prominence through their success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a benchmark competition in the field. Even with their varying characteristics, these architectures all aim to better process visual information. Additionally, while some architectures may be widely used, the performance of one architecture over another can change depending on the dataset being used. In this study, we chose the AlexNet and VGG16 architectures, along with ResNet50, which is known for its strong results in (*Esgario, Krohling & Ventura, 2020*).

AlexNet: In 2012, Alex Krizhevsky and his team introduced a CNN model that was more sophisticated than LeNet and won the challenging ILSVRC in visual object recognition. It exhibited a high performance and achieved a lower error rate compared to existing models in the competition. The architecture of AlexNet is composed of eight layers in total and it accepts input images that are size of (224,224,3). The first five layers are specialized for convolution operations, and designed for feature extraction. These layers make use of a combination of max pooling and rectified linear unit (ReLU) activation functions to augment the learning process. Following these, there are three fully connected layers, which integrate the learned features for classification. Dropout is crucial in the fully connected layers to mitigate the problem of overfitting. The role of dropout here is to randomly disable a portion of neurons during the training, thus ensuring that the network does not become too dependent on any particular group of neurons.

VGG16: The VGG16 architecture was conceptualized and introduced by Karen Simonyan and Andrew Zisserman from the Visual Geometry Group Lab. In the 2014 ILSVRC competition, which was organized into two categories—object recognition and image classification—they won first place in object recognition and second in image classification. It accepts an input image of dimensions 224 × 224 pixels, with three color channels. The architecture is deep, comprising sixteen layers. The entire architecture has five groups of cascaded convolution layers. After each group of convolution layers, there is a max-pooling layer, which is implemented with a stride of (2,2), aiding in reducing the spatial size of the representation and in turn, the computational load. In the convolutional layers, receptive filters of different numbers, 64, 128, 256, and 512, are used. All these filters share a kernel size of (3,3). The final three layers are fully connected, with the first two layers having 4,096 neurons each. The last layer contains the number of neurons corresponding to the total number of classes in the classification task. The ReLU activation function is applied throughout the convolutional and hidden layers, while the softmax activation function is used in the last fully connected layer to compute the class probabilities. However, managing the VGG16 can be challenging due to its extensive structure, comprising approximately 138 million parameters.

ResNet50: ResNet50 is a version of the well-known ResNet architecture, which is short for "Residual Network". This deep architecture introduced a novel concept known as 'residual connections'. In a ResNet, the fundamental building block is the residual block, where the input to the block is connected to the output of the block, a connection known as the 'residual connection'. The key feature is that it allows the layers to learn the residual function that maps the input to the target output, rather than learning the complete mapping from input to output. This makes it easier to optimize and train very deep networks.

The ResNet50 model starts with a convolutional layer, followed by batch normalization, and then applies a ReLU activation function. It continues with four major sets of residual blocks. Each set comprises multiple residual blocks, and each block is made up of several convolutional layers followed by batch normalization and a ReLU activation function. After the residual blocks, global average pooling is typically used to minimize the spatial dimensions of the feature maps. The network ends with a fully connected layer that performs the final classification.

The transfer learning approach is used to increase the efficiency of the training and to achieve faster results. We utilized the pre-trained CNN architectures trained on the ImageNet database. Several modifications are introduced to these pre-trained models:

- The number of units in the last linear layer is adjusted to match the number of classes in our dataset.
- Since an additional channel is added to each RGB image, the number of input channels in the first convolutional layer configuration is increased from 3 to 4.
- The network weights associated with this newly added channel are set to be the mean of the weights from the existing three layers.

During the training, no layers are frozen, that is, we allowed every layer of the network to be trained and adjusted. This is implemented to ensure that the network can adapt more comprehensively to the specific requirements of our dataset and tasks.

There are two jobs in the dataset: disease class prediction and stress level estimation. The single-task CNN and the multi-task CNN are two common methodologies we might adopt. Their fundamental difference lies in the configuration of their feature extraction layers. The single-task approach uses separate convolution and pooling layers dedicated to each task, whereas the multi-task approach uses shared convolution and pooling layers across tasks. We have chosen to pursue a single-task learning approach. This strategy is centered on focusing our computational resources and design efforts on one specific task and supports a more targeted and efficient learning process on our dataset.

We determined an optimal set of hyperparameters to drive our DL models. Our choices included stochastic gradient descent (SGD) as the optimizer, with cross-entropy as the loss function. Training occurred over 30 epochs, each processing a batch size of 32 images. The learning rate is initiated at 0.001 and implemented a scheduled decrease by a factor of 1/2 every 10 epochs to fine-tune the learning process. Additionally, momentum is set to 0.9 for efficient convergence. To prevent overfitting and promote generalization in our model, we applied a weight decay of 0.0005.

Dealing with small and imbalanced datasets brings about primary concerns related to biases and the risk of overfitting. Data augmentation methods offer a powerful solution to address these issues by artificially expanding and diversifying the training dataset. In image classification, augmentations like rotation, flipping, and brightness adjustments help create a more balanced representation of different angles, orientations, and lighting conditions. These transformations mimic real-world variations and infuse variability into the training data. In our approach to data augmentation, a variety of standard transformations are applied to produce synthetic images using the original training dataset. This included implementing random horizontal flips and random vertical flips, which introduce changes in orientation and mirroring. Furthermore, random rotations are integrated to simulate various viewpoints. To further enhance the diversity of color variations in our data, random adjustments in brightness, contrast, and saturation are performed.

The diagram in Fig. 5 illustrates an overview of our image classification task. The capsule-shaped section presents the sequence of operations that occur during a single epoch within a model training cycle. Firstly, a set of augmented training images is fed into the pre-trained CNN architecture. Once the model is trained, it is evaluated using a validation set. During the performance evaluation of the trained model, the validation loss is computed and saved. This process continues until the maximum number of epochs is completed. Then, the model with the lowest validation loss across all epochs is recognized as the best-performing model. Finally, the selected model is evaluated on a testing set, with its performance in classification being measured using metrics like accuracy, precision, recall, and the F1-score.

EXPERIMENTAL RESULTS

Performance metrics

Performance metrics are essential in evaluating and guiding the development of ML and DL models. These benchmark functions act as quantifiable indicators of a model's performance, allowing practitioners to impartially evaluate the extent to which their models are meeting the desired objectives. Fundamentally, these metrics are instrumental in enabling well-informed decision-making processes during the critical phases of model selection and

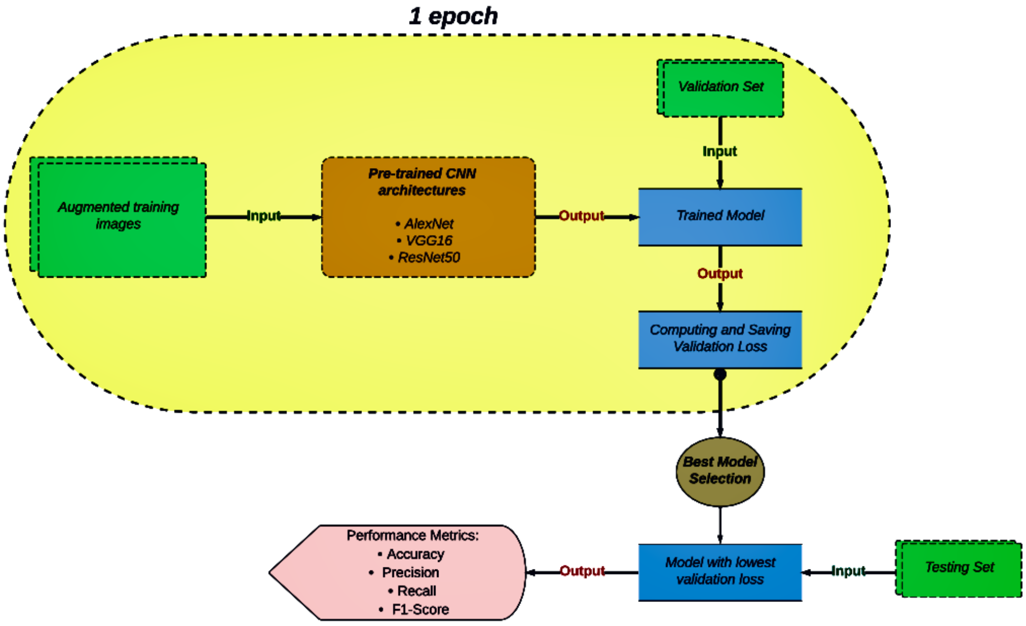
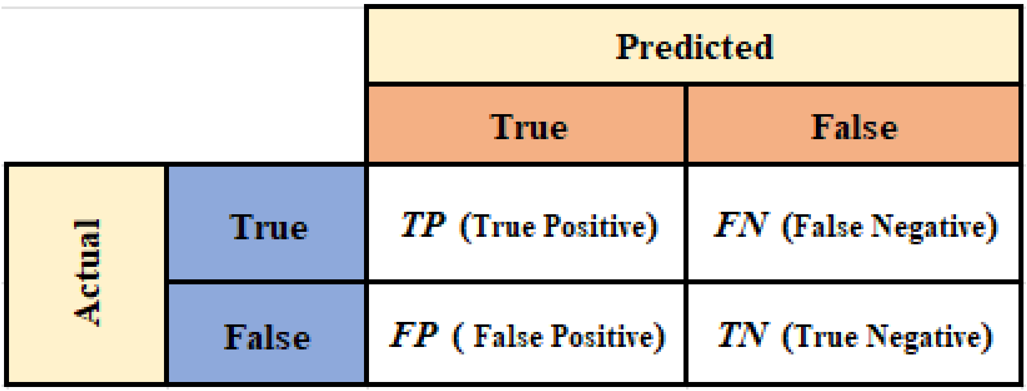


Figure 5 **Flowchart of the classification process.**

Full-size DOI: 10.7717/peerjcs.2406/fig-5

tuning. They assist in the precise identification of the most appropriate algorithms and parameter configurations for a specific task. In real-world applications, these metrics offer valuable insights to stakeholders regarding the reliability and effectiveness of the models contributing to informed decision-making. They also play a key role in the analysis and assessment of model behavior, permitting the identification and rectification of critical issues, including but not limited to overfitting, underfitting, and biases.

In our study, various performance metrics (*Bishop & Nasrabadi, 2006*) were employed to demonstrate the effectiveness and success of our methodology. As indicated in 'Dataset Description', our datasets are partially imbalanced and highly imbalanced. Although the accuracy metric is not typically useful for imbalanced datasets, it was utilized to demonstrate the superiority of our results compared to the reference study (*Esgario, Krohling & Ventura, 2020*). Furthermore, we chose the metrics of recall and precision, which were included to allow for a more accurate assessment of a model's performance in imbalanced datasets and to enable a direct comparison with the mentioned research. However, precision is insensitive to false negatives, which means it can provide an overly optimistic assessment when many relevant instances are missed. Conversely, recall is insensitive to false positives, potentially leading to a misleading sense of performance if many irrelevant instances are incorrectly classified as positive. We addressed these limitations in precision and recall by employing the F1-score. It harmonizes both and provides a more balanced evaluation. Moreover, due to the complexity and depth of the models, it is often necessary to use additional metrics. Therefore, we employed the area under the receiver operating characteristic (ROC) curve (*Fawcett, 2006*), known as AUC-ROC, to measure a model's ability to distinguish between classes.



		Predicted	
		True	False
Actual	True	*TP* (True Positive)	*FN* (False Negative)
	False	*FP* (False Positive)	*TN* (True Negative)

Figure 6 **Confusion matrix.**

Full-size DOI: 10.7717/peerjcs.2406/fig-6

The metrics referred to are computed through the confusion matrix, which is depicted in Fig. 6. The confusion matrix provides detailed information on true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), and it also suggests deeper insights into how well a classification model is performing. Accuracy quantifies the ratio of correct predictions out of total predictions and is formulated by Eq. (10):

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}. \tag{10}$$

Precision measures the proportion of true positive results among all positive predictions. This metric is necessary for understanding how successful the model is in correctly identifying positive cases. It is defined by Eq. (11):

$$Precision = \frac{TP}{TP + FP}. \tag{11}$$

Recall, also known as sensitivity, evaluates the model's ability to identify all relevant instances, indicating the percentage of true positives that are correctly identified and is defined by Eq. (12):

$$Recall = \frac{TP}{TP + FN}. \tag{12}$$

The F1-score is the harmonic mean of precision and recall. It offers a single metric that helps to convey the overall effectiveness of a model in classifying data correctly, balancing the trade-offs between false positives and false negatives. It is represented by Eq. (13):

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{13}$$

Results and discussions

All experiments in this study are conducted using the popular open-source ML library PyTorch. The Intel Iris Xe Graphics is used as the GPU. The hardware components feature an 11th Gen Intel® Core™ i5-1135G7 CPU with a base clock speed of 2.40 GHz.

Table 2 Biotic stress dataset results.

	Biotic stress dataset											
Metrics:	**Accuracy %**			**Precision %**			**Recall %**			**F1-Score %**		
α	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
AlexNet	94.86	94.47	94.07	93.92	94.55	95.29	91.79	89.84	89.51	92.71	91.54	91.44
VGG16	**95.65**	**95.65**	94.07	96.62	**96.65**	92.42	92.60	**93.38**	90.36	94.15	**94.73**	91.24
ResNet50	94.86	94.86	93.68	94.01	95.93	93.78	92.68	91.00	90.07	93.28	92.77	91.48

Notes.
The best result for each evaluation metric is highlighted in bold.

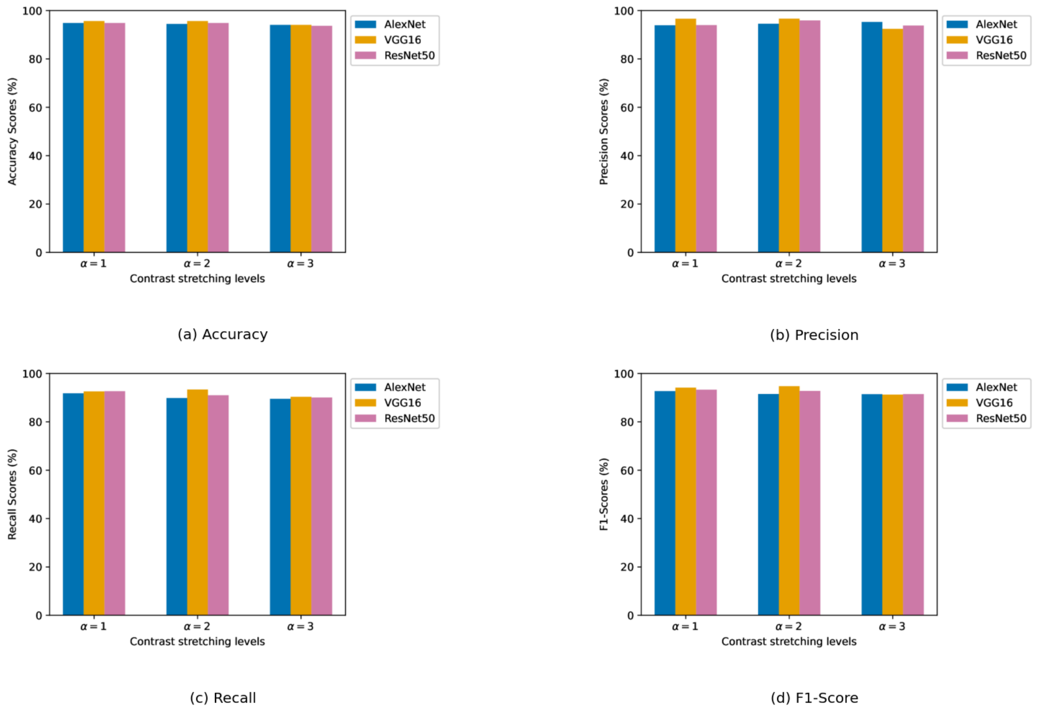
Table 3 Stress severity dataset results.

	Severity dataset							
Performance metrics	**Accuracy %**		**Precision %**		**Recall %**		**F1-Score %**	
α	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
AlexNet	88.14	88.14	83.55	84.89	**85.30**	84.61	84.28	84.62
VGG16	89.33	89.72	84.17	84.83	83.17	85.18	83.58	**84.83**
ResNet50	86.96	**90.11**	83.21	**86.35**	74.30	79.51	77.06	81.84

Notes.
The best result for each evaluation metric is highlighted in bold.

We experimented with different contrast enhancement levels. For the biotic stress dataset, the contrast enhancement levels are chosen as $\alpha \in \{1, 2, 3\}$. For the stress severity dataset, a slightly narrower range of contrast enhancement levels is used, with $\alpha \in \{1, 2\}$. Numerical results of the CNN models are presented in two tables. Table 2 details the results pertaining to the biotic stress dataset, while Table 3 presents the model results on the stress severity dataset. In both tables, results are organized according to the different levels of contrast enhancement applied. This shows a clear view of how contrast levels influence model performance. The best result for each evaluation metric is highlighted in bold font. Furthermore, the graphical results of CNN architectures in relation to each performance metric are depicted in Figs. 7A, 7B, 7C, and 7D obtained from the biotic stress dataset, and Figs. 8A, 8B, 8C, and 8D, obtained from the severity dataset. These graphs show how the models' success differs from one another in related levels of contrast enhancement.

Biotic stress dataset: VGG16 stood out as the architecture exhibiting the best results in all metrics. For VGG16, the best results in all metrics are produced at $\alpha = 2$. Its precision scored above 96% at α levels 1 and 2, outperforming the other models. At every alpha level, it achieved a higher recall value than other models, suggesting a more reliable detection of samples. AlexNet demonstrated high consistency in accuracy metric across all α levels with over 94% accuracy, even though there was a slight decrease as α increased. Its precision and recall also remained robust, with only minor fluctuations. ResNet50's performance is competitively comparable to AlexNet in terms of accuracy. It showed a notable increase in precision at α level 2, but its recall is remarkable, particularly at α level 1, which indicates a



(a) Accuracy

(b) Precision

(c) Recall

(d) F1-Score

Figure 7 **(A–D) Performance metrics comparison of CNN models across different contrast stretching levels for biotic stress dataset.**

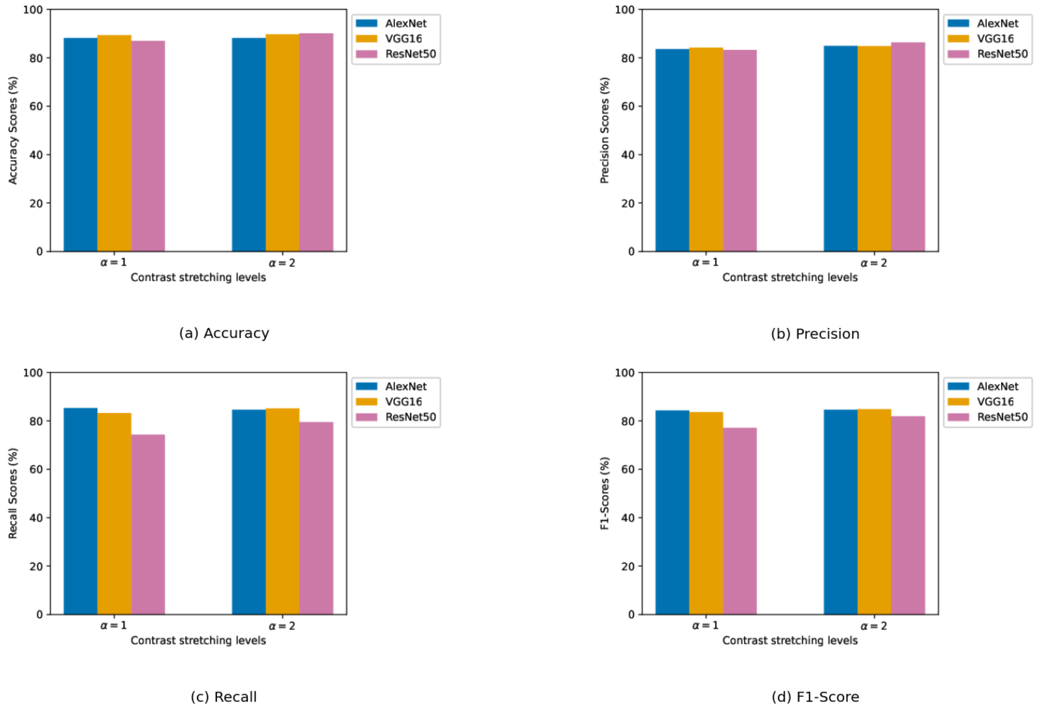
Full-size DOI: 10.7717/peerjcs.2406/fig-7

high rate of true positive detections. On the other hand, at the $\alpha = 3$, performance metrics results are generally poorer than those observed at the two other α levels.

The impact of EMPR-based channel increment on the performance of CNN models is analyzed by contrasting it with the findings of study *Esgario, Krohling & Ventura (2020)*. The addition of an extra channel led to significant improvements in the AlexNet model. It surpassed the accuracy, precision, and recall metrics of the reference study. Similarly, it had a positive effect on the VGG16 model, with a meaningful increment in the recall value. On the other hand, the accuracy and precision values of the VGG16 remained competitive with the results reported in *Esgario, Krohling & Ventura (2020)*. However, our ResNet50 model fell behind the results of the reference study.

Stress severity dataset: Unlike the biotic stress dataset, no individual model achieves the best result in overall performance metrics. ResNet50 excelled in accuracy and precision, AlexNet in recall, and VGG16 in F1-score. At $\alpha = 1$, VGG16 had the highest accuracy and precision with scores of 89.33% and 84.17%, respectively, while AlexNet led in recall and F1-score having a recall of 85.30% and an F1-score of 84.28%. Although ResNet50's results are close to those of AlexNet and VGG16 in terms of accuracy and precision, it significantly underperformed in recall by a wide margin. This is evident in the F1-score and caused the gap to widen.

When α is set to 2, there is a general increase in performance across all metrics for each deep architecture. ResNet50 benefited more from the increased α parameter and hit



(a) Accuracy

(b) Precision

(c) Recall

(d) F1-Score

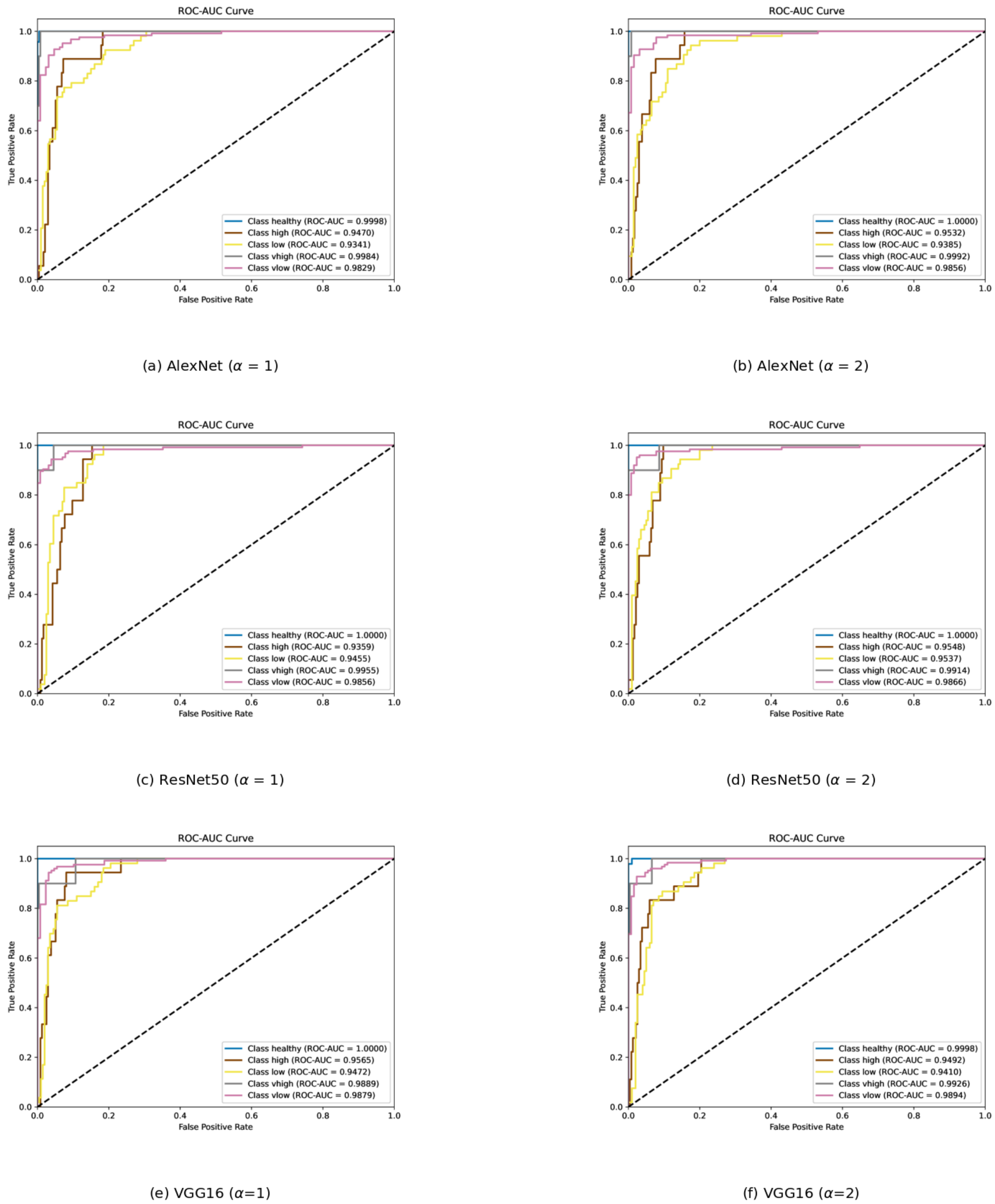
Figure 8 **(A–D) Performance metrics comparison of CNN models across different contrast stretching levels for severity dataset.**

Full-size DOI: 10.7717/peerjcs.2406/fig-8

the best results in accuracy and precision among others. Although ResNet50 has quite high accuracy for both α levels, its F1-score indicates that the model struggles to learn images due to the imbalanced structure of the dataset. VGG16 performed better than it did at $\alpha = 1$, and it secured the highest ranking in both recall and F1-score. AlexNet, despite lagging slightly behind in accuracy and precision, still demonstrated considerable effectiveness with substantial scores across all metrics.

Channel incrementation marked a major contribution when compared to the findings of the reference research. All CNN architectures contributed to the overall accuracy within the range of 3–6 points. While *Esgario, Krohling & Ventura (2020)* is unable to reach accuracy above 90%, our ResNet50 exceeded this threshold. On the other hand, EMPR-based incrementation addressed the problem of poor precision and recall results, especially due to the extreme imbalance of the severity dataset. For both metrics, most of our models got results above 80%, while VGG16 is the only model in the reference study to exceed this limit. Furthermore, AlexNet delivered groundbreaking outcomes. It jumped by about 10 points on average in precision and about 12 points on average in recall. The models are ranked on these two metrics as follows: for precision metric improvement, it is AlexNet > ResNet50 > VGG16, and for recall metric improvement, it is AlexNet > VGG16 > ResNet50.

To determine which class is learned by the classifier with difficulty, ROC curves for each classification model are presented in Fig. 9. The area under the ROC curve provides



(a) AlexNet (α = 1)

(b) AlexNet (α = 2)

(c) ResNet50 (α = 1)

(d) ResNet50 (α = 2)

(e) VGG16 (α=1)

(f) VGG16 (α=2)

Figure 9 **(A–D) ROC curves of each classification model for severity dataset.**
Full-size DOI: 10.7717/peerjcs.2406/fig-9

information about how well the classifier responds to various classes. The AUC value falls within the range of 0 to 1. The higher the AUC, the better the model's prediction. An AUC of 0.5 indicates a lack of discriminative power, while an AUC of 1 indicates perfect discrimination. A classifier with an AUC below 0.5 implies that it has performance worse than random guessing.

The "class healthy" curves are very close to the top left corner with an AUC of approximately 1 for all three models, indicating an excellent performance in distinguishing the healthy class from the rest. Although “class very high” is a minority class, it is also easily recognized by the models. Both classes are well-separated from the other categories. “Class high” and “Class low” have the lowest AUC scores and display less confident predictive abilities. There may be some overlaps in the features with other classes, which reduces their distinctiveness. Increasing α from 1 to 2, AlexNet and ResNet50 recognized “class low” and “class high” images better. This change in α appears to have a positive impact on AlexNet and ResNet50, allowing them to accurately detect “class low” and “class high” images. However, as alpha increases, VGG16 struggles to recognize both kinds of images.

CONCLUSION

Detecting diseases in coffee leaves is essential for maintaining the sustainability of coffee cultivation. Accurate and timely diagnosis not only aids in the effective management of crop health but also significantly contributes to the prevention of widespread agricultural losses. Our proposed study introduced a pre-processing strategy that combines the HDMR and EMPR concepts, and several parameters were key to advancing the diagnosis of diseases and disease severity in coffee leaves. In the pre-processing phase, in addition to the R, G, and B channels, a robust channel is formed by mixing low and high-frequency EMPR components, and it is further reinforced with contrast improvements based on HDMR. Additionally, the performance of deep CNN architectures has been investigated across various contrast enhancement levels for both tasks.

For the purpose of experimentation, well-known CNN architectures such as AlexNet, ResNet50, and VGG16 are implemented by adopting a single-task approach with transfer learning. Furthermore, data augmentation techniques are used to avoid the lack of learning due to the presence of fewer examples in one class compared to others. The testing results revealed that VGG16 performed better than previous research in obtaining the most accurate diagnoses for diseases in coffee leaves. This success was attained by incorporating a new EMPR-based channel and applying contrast enhancement at the $\alpha = 2$ level. When using the same parameter settings, but with any alpha level, AlexNet has significantly improved upon the disease identification results obtained in the reference study. Moreover, our work successfully addressed the challenge in stress severity estimation that the previous study could not handle and fell short of. All CNN architectures achieved not only higher precision and recall but also an accuracy of around 90%.

Although this study has demonstrated the effectiveness of the proposed method in enhancing the classification capability of deep architectures, several areas remain unexplored. One of the primary factors influencing classification success is the bias introduced by the background, which should be addressed in future research. Efforts could aim to adapt EMPR and HDMR to disease segmentation. Additionally, we believe that developing novel sampling techniques inspired by EMPR could help overcome the frequently observed between-class imbalance in such disease datasets.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by Istanbul Technical University within the scope of project number FHD-2024-45407. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript

Grant Disclosures

The following grant information was disclosed by the authors:
Istanbul Technical University: FHD-2024-45407.

Competing Interests

Erfan Babaee Tirkolaee is an Academic Editor for PeerJ.

Author Contributions

- Ahmet Topal conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Burcu Tunga conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Erfan Babaee Tirkolaee conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The data, algorithms and code are available at GitHub and Zenodo:

- https://github.com/ArticleCodeHub/DeepEMPR

- Topal, A. (2024). DeepEMPR. Zenodo. https://doi.org/10.5281/zenodo.13823450.

The code is available in the Supplemental File.

The data is also available at Figshare: Topal, Ahmet (2024). LeafData.zip. figshare. Dataset. https://doi.org/10.6084/m9.figshare.26060464.v1.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.2406#supplemental-information.

REFERENCES

Agarwal M, Singh A, Arjaria S, Sinha A, Gupta S. 2020. ToLeD: tomato leaf disease detection using convolution neural network. *Procedia Computer Science* **167**:293–301 DOI 10.1016/j.procs.2020.03.225.

Ahmed K, Shahidi TR, Alam SMI, Momen S. 2019. Rice leaf disease detection using machine learning techniques. In: *2019 international conference on sustainable technologies for industry 4.0 (STI)*. Piscataway: IEEE, 1–5.

Bishop CM, Nasrabadi NM. 2006. *Pattern recognition and machine learning*. Vol. 4(4). New York: Springer.

Breiman L. 2001. Random forests. *Machine Learning* **45**:5–32 DOI 10.1023/A:1010933404324.

Cover T, Hart P. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* **13(1)**:21–27 DOI 10.1109/TIT.1967.1053964.

Dalal N, Triggs B. 2005. Histograms of oriented gradients for human detection. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol. 1*. Piscataway: IEEE, 886–893.

Esgario JG, Krohling RA, Ventura JA. 2020. Deep learning for classification and severity estimation of coffee leaf biotic stress. *Computers and Electronics in Agriculture* **169**:105162 DOI 10.1016/j.compag.2019.105162.

Fawcett T. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* **27(8)**:861–874 DOI 10.1016/j.patrec.2005.10.010.

Ferentinos KP. 2018. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture* **145**:311–318 DOI 10.1016/j.compag.2018.01.009.

Ghadi YY, Mazhar T, Shah SFA, Haq I, Ahmad W, Ouahada K, Hamam H. 2023. Integration of federated learning with IoT for smart cities applications, challenges, and solutions. *PeerJ Computer Science* **9**:e1657 DOI 10.7717/peerj-cs.1657.

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. 2017. Mobilenets: efficient convolutional neural networks for mobile vision applications. ArXiv arXiv:1704.04861.

International Telecommunication Union. 2011. Studio encoding parameters of digital television for standard 4: 3 and wide-screen 16: 9 aspect ratios. In: *International Telecommunication Union, Radiocommunication Sector*. Geneva: International Telecommunication Union.

Kaur P, Pannu HS, Malhi AK. 2019. Plant disease recognition using fractional-order Zernike moments and SVM classifier. *Neural Computing and Applications* **31**:8749–8768 DOI 10.1007/s00521-018-3939-6.

Korkmaz Özay E, Tunga B. 2022. Hyperspectral image denoising with enhanced multivariance product representation. *Signal, Image and Video Processing* **16(4)**:1–7.

Krizhevsky A, Sutskever I, Hinton GE. 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* **25**:1097–1105 .

MacQueen J. 1967. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1(14)*. Oakland, CA, USA 281–297.

Mohanty SP, Hughes DP, Salathé M. 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science* **7**:1419 DOI 10.3389/fpls.2016.01419.

Nafi NM, Hsu WH. 2020. Addressing class imbalance in image-based plant disease detection: deep generative vs. sampling-based approaches. In: *2020 international conference on systems, signals and image processing (IWSSIP)*. Piscataway: IEEE, 243–248.

Neelakantan P. 2021. Analyzing the best machine learning algorithm for plant disease classification. *Materials Today: Proceedings* **80(3)**:3668–3671 DOI 10.1016/j.matpr.2021.07.358.

Ölçer N, Ölçer D, Sümer E. 2023. Roof type classification with innovative machine learning approaches. *PeerJ Computer Science* **9**:e1217 DOI 10.7717/peerj-cs.1217.

Panigrahi KP, Das H, Sahoo AK, Moharana SC. 2020. Maize leaf disease detection and classification using machine learning algorithms. In: *Progress in computing, analytics and networking: proceedings of ICCAN 2019*. Singapore: Springer, 659–669 DOI 10.1007/978-981-15-2414-1_66.

Prajapati HB, Shah JP, Dabhi VK. 2017. Detection and classification of rice plant diseases. *Intelligent Decision Technologies* **11(3)**:357–373 DOI 10.3233/IDT-170301.

Pranto TH, Noman AA, Mahmud A, Haque AB. 2021. Blockchain and smart contract for IoT enabled smart agriculture. *PeerJ Computer Science* **7**:e407 DOI 10.7717/peerj-cs.407.

Quinlan JR. 1986. Induction of decision trees. *Machine Learning* **1**:81–106.

Ramesh S, Hebbar R, Niveditha M, Pooja R, Shashank N, Vinod PV, Prasad Bhat N. 2018. Plant disease detection using machine learning. In: *2018 international conference on design innovations for 3Cs compute communicate control (ICDI3C)*. Piscataway: IEEE, 41–45.

Richard B, Qi A, Fitt BD. 2022. Control of crop diseases through Integrated Crop Management to deliver climate-smart farming systems for low-and high-input crop production. *Plant Pathology* **71(1)**:187–206 DOI 10.1111/ppa.13493.

Şen ME, Tuna S. 2023. Exploiting optimal supports in enhanced multivariance products representation for lossy compression of hyperspectral images. In: *2023 14th international conference on electrical and electronics engineering (ELECO)*. Piscataway: IEEE, 1–5.

Shrivastava VK, Pradhan MK, Minz S, Thakur MP. 2019. Rice plant disease classification using transfer learning of deep convolution neural network. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **42**:631–635.

Simonyan K, Zisserman A. 2014. Very deep convolutional networks for large-scale image recognition. ArXiv arXiv:1409.1556.

Singh A, Kaur H. 2021. Potato plant leaves disease detection and classification using machine learning methodologies. *IOP Conference Series: Materials Science and Engineering* **1022(1)**:012121.

Sujatha R, Chatterjee JM, Jhanjhi N, Brohi SN. 2021. Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocessors and Microsystems* **80**:103615 DOI 10.1016/j.micpro.2020.103615.

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. 2016. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. Piscataway: IEEE, 2818–2826.

Too EC, Yujian L, Njuki S, Yingchun L. 2019. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture* **161**:272–279 DOI 10.1016/j.compag.2018.03.032.

Topal A, Amasyali MF. 2021. When does synthetic data generation work? In: *2021 29th signal processing and communications applications conference (SIU)*. Piscataway: IEEE, 1–4.

Tuna S, Töreyin BU, Demiralp M, Ren J, Zhao H, Marshall S. 2020. Iterative enhanced multivariance products representation for effective compression of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing* **59(11)**:9569–9584 DOI 10.1109/TGRS.2020.3031016.

Tuna S, Tunga B. 2013. A novel piecewise multivariate function approximation method via universal matrix representation. *Journal of Mathematical Chemistry* **51**:1784–1801 DOI 10.1007/s10910-013-0179-y.

Tunga MA. 2015. A new approach for multivariate data modelling in orthogonal geometry. *International Journal of Computer Mathematics* **92(9)**:2011–2021 DOI 10.1080/00207160.2014.941825.

Tunga B, Demiralp M. 2010. The influence of the support functions on the quality of enhanced multivariance product representation. *Journal of Mathematical Chemistry* **48**:827–840 DOI 10.1007/s10910-010-9714-2.

Tunga B, Kocanaogullari A. 2017. Digital image decomposition and contrast enhancement using high-dimensional model representation. *Signal, Image and Video Processing* **12(2)**:299–306 DOI 10.1007/s11760-017-1158-8.

Vapnik V. 2013. *The nature of statistical learning theory*. New York: Springer Science & Business Media.