Comparative Analysis of CNN Models and Bayesian Optimization-Based Machine Learning Algorithms in Leaf Type Classification

Muhammet Fatih Aslan

Abstract— **In this study, the leaves are classified by various Machine Learning (ML) and Deep Learning (DL) based Convolutional Neural Networks (CNN) methods. In the proposed method, first, image pre-processing is performed to increase the accuracy of the posterior process. The obtained image is a grayscale image without noise as a result of the pre-processing. These pre-processed images are used in classification with ML and DL. The Speeded Up Robust Features (SURF) are extracted from the grayscale image for ML-based learning. The features are restructured as visual words using the Bag of Visual Words (BoVW) method. Then, histograms are generated for each image according to the frequency of the visual word. Those histograms represent the new feature data. The histogram features are classified by four different ML methods, Decision Tree (DT), k-Nearest Neighbor (KNN), Naive Bayes (NB) and Support Vector Machine (SVM). Before using the ML methods, the Bayesian Optimization (BO) method, which is one of the Hyperparameter Optimization (HO) algorithms, is applied to determine hyperparameters. In the classification process performed with four different ML algorithms, the best accuracy is achieved with the KNN algorithm at 98.09%. Resnet18, ResNet50, MobileNet, GoogLeNet, DenseNet, which are state-of-the-art CNN architectures, are used for DL-based learning. CNN models have higher accuracy than ML algorithms.**

Index Terms—**Bag of Visual Words, Bayesian Optimization, Convolutional Neural Networks, Deep Learning, Speeded Up Robust Features**

I. INTRODUCTION

IN AGRICULTURE, plants must be constantly observed to ensure the continuity of food production, and efforts must be made to ensure that different plant species do not become extinct. Plants are the essential source of life on earth. All living organisms on earth need nutrition and water. Plants release oxygen to the atmosphere through photosynthesis to meet these main needs. In addition, climate and the distribution of gases in the atmosphere become regular due to the plants.

MUHAMMET FATIH ASLAN, is with Department of Electrical Engineering, Karamanoglu Mehmetbey University, Karaman, Turkey, (e-mail: mfatihaslan@kmu.edu.tr).

https://orcid.org/0000-0001-7549-0137

Manuscript received Sep 12, 2022; accepted Nov 15, 2022.
DOI: 10.17694/bajece.1174242

Many animals in nature meet their sheltering needs in the favor of plants. Substantial needs of people and animal such as drugs, fuel, etc. are provided by plants [1]. and animals in nature need plants, hundreds of plants are extinct due to human activities over the past 200 years. The continuity of plant production has an enormous effect on the future of human generation. Therefore, various precautions should be taken to prevent the extinction of the plant. For this purpose, one of the precautions taken is the identification of plant species [2].

Recognition is the most effective one for the protection of plant species. If plant species can be recognized accurately, precautions can be taken for the endangered plant species so that, the generation of plant species can be protected. But given the existence of nearly 3 million named and classified plant species on earth, this is no easy task[3]. Because it requires in-depth knowledge and experience about botany and plant systematics [4]. In addition, classification techniques such as morphological anatomy, cell biology, molecular biology, and photochemistry are complex and challenging topics, and therefore not suitable for online applications [5, 6]. As a result, the detection of all plant species on earth is an important but difficult task.

For easier plant identification, using plant leaves has long been accepted by researchers [7]. The leaves on the plant are the strongest determinant of that plant type. Because unlike flowers and fruits which appear in a short period, they are both outnumbered and long-lasting [8]. The exact color of the leaves can differ depending on the climate. That's the reason why it is essential to make a classification according to both shape and tissue features. In this way, more reliable results can be achieved [9]. In addition to recognizing the plant species, leaves also provide the discovery of important information such as plant development and plant disease. However, given the large number of leaf species, leaf recognition by traditional methods is a time-consuming and difficult task for botanists. Recently, such classification tasks have become easier and faster to be solved automatically with computer-aided systems [10].

Due to the disadvantages of traditional methods, studies using automatic identification methods have increased. Gaston and O'Neill [11] stated that it is possible to identify the plants automatically with the recent developments in the field of Artificial Intelligence (AI) and Digital Image Processing. The classification studies of the plants gained popularity in the past decade with developments in both Machine Learning (ML) and Deep Learning (DL) fields. For example, in papers by Kulkarni, Rai, Jahagirdar and Upparamani [12], Kumar, Belhumeur, Biswas, Jacobs, Kress, Lopez and Soares [13] and Wäldchen, Rzanny, Seeland and Mäder [14] different computer vision and AI-based applications were introduced to improve leaf identification performance.

Plant identification studies provide significant benefits in applications for smart and precision agriculture. The development of automatic and rapid plant recognition applications can be used to know which plants have benefits and harms. In addition, leaf-based plant classification applications allow exciting applications for autonomous agricultural applications [15]. In this sense, the prominent application area is selective spraying. Vehicles that perform autonomous spraying according to the map of weed and crop distributions with selective spraying are an active field of application [16]. In addition, autonomous applications can be developed to monitor different crops grown in a greenhouse. Recently, different applications have been developed for precision agriculture with Unmanned Aerial Vehicles (UAV) and Unmanned Ground Vehicles (UGV). A comprehensive survey study on this was presented by Aslan, Durdu, Sabanci, Ropelewska and Gültekin [17]. As a result, plant identification applications are important and necessary in different fields for precision agriculture.

In this study, learning-based ML and DL methods are applied to identify leaves with high accuracy. For this, leaf images in the Folio [18] dataset are used. First, simple image processing steps are applied to raw images. In this way, irrelevant features that will negatively affect the artificial intelligence result are removed. ML and DL are implemented on these pre-processed images. While the feature extraction step is required first for traditional ML methods, feature extraction is performed automatically through deep layers in DL. In this study, to extract the necessary features for ML, Speeded Up Robust Features (SURF), a very powerful feature descriptor, is used. In addition, Bag of Visual Words (BoVW) is used to cluster a large number of features and express these clusters with histograms. Using the obtained histograms, the classification of leaf images is completed. In experimental studies, various ML methods such as Decision Tree (DT), k-Nearest Neighbor (KNN), Naive Bayes (NB) and Support Vector Machine (SVM) are applied with Bayesian Optimization (BO) to get the best performance from ML techniques. After the ML step, pre-processed leaves are classified with CNN models that can automatically extract features and perform classification. For this, frequently preferred methods such as Resnet18 [19], ResNet50 [19], MobileNet [20, 21], GoogLeNet [22] and DenseNet [23] are used. In the last step, the model that gives the most accurate result for the Folio dataset is also applied to the Swedish dataset to prove the result.

The hypothesis and novelty in this study are that traditional ML algorithms powered by SURF, BoVW and BO steps can perform close to popular CNN models in leaf recognition. The contributions of this article can be summarized as follows:

1) SURF features extracted from leaf images are strengthened with the BoVW method.

2) Different ML algorithms with powerful feature extraction and optimization steps are compared with different CNN models.

3) This study compares the high-level feature extraction power of CNN with ML methods including SURF, BoVW and BO.

4) At the end of the study, high leaf recognition successes are obtained.

The article is organized as follows. After the introduction, previous studies on leaf classification are discussed in section 2. Section 3 introduces leaf datasets used in learning algorithms. The pre-processing steps applied to leaf images, the BoVW approach, the ML and DL algorithms used, etc. are explained in section 4. Section 5 analyzes the ML and DL results obtained as a result of applying the proposed methods. Finally, section 6 concludes the work.

II. Related Works

Unfortunately, existing methods are not capable enough for the strong classification of the leaves [5, 24]. Therefore, many researchers aim to increase the accuracy of the classification by using different algorithms and their combinations. Vijaya Lakshmi and Mohan [25] suggested a new circle-based radii model. This model was based on the leaf's center point and leaf boundaries. 50 out of 220 classes from the ICL dataset were used. 44 features were extracted from each leaf. The SVM algorithm was used for classification. As a result of the study, a success of 93.33% was achieved. Koklu et al. [26] created grapevine leaf images and classified these images with the MobileNetv2 CNN model. They proposed three different approaches to classification. They implemented fine-tuned MobileNetv2 in the first, the MobileNetv2-SVM structure in the second, and the MobileNetv2-SVM-feature selection in the last. Also, different kernel functions for SVM were taken into account. At the end of the study, the CNN-SVM structure based on the selected features performed in the last step performed more successful classification with 97.60% accuracy. Sharma et al. [27] performed leaf classification based on color and texture features. They used HSV (hue, saturation, value) color space to extract color features and the Gray-Level Co-Occurrence Matrix (GLCM) algorithm to extract texture features. Extracted features were given as input to Artificial Neural Network (ANN), both individually and by fusion. At the end of the study, it was stated that the fusion features were stronger for leaf recognition. Arun and Viknesh [28] performed leaf classification on the Mendeley dataset using eleven different pre-trained CNN models such as AlexNet, Xception, ResNet50, and EfficientNet. In the comparisons made in the study, it was stated that the EfficientNet B5 model outperformed other models with 99.75% accuracy. Jin, Hou, Li and Zhou [24] emphasized leaf tooth features. The deformities in the leaf squares were removed with the Pauta criterion. After this, four-leaf tooth features (Leaf-obliqueness, Leaf-rate, Leaf-num and Leaf-sharpness, and Leaf-obliqueness) were obtained. As a result of the study, accuracy values between 73% and 80% were obtained. Saleem et al. [29], focused on handcrafted visual leaf features, feature extraction techniques, and classification methods for the analysis of visual leaf shape features in plant classification. In the experiments performed on the Flavia dataset, they presented a five-step algorithm for leaf recognition consisting of pre-processing and segmentation, feature extraction, etc. They used KNN, DT, NB, and multi-SVM classifiers in the proposed algorithm. At the end of the study, KNN was the most successful ML method with 98.75% accuracy. Also, at the end of the study, the proposed method was compared with AlexNet. The proposed method was more successful than AlexNet in the presence of less training data. Dudi and Rajesh [30] performed a DL-based plant identification application using the Swedish and Mendeley dataset. First, they applied pre-processing steps such as median filtering and histogram equalization to the leaves. These images were then classified by CNN optimized with the Shark Smell-based Whale Optimization Algorithm (SS-WOA). They also introduced a threshold dependent classification method to classify untrained images. The proposed SS-WOA application outperformed different ML methods such as NB, SVM for both trained and untrained data. Sladojevic et al. [31] applied DL that was recently overemphasized in the field of ML. In that study, recognition of plant disease was aimed. Hence, leaf recognition was done using Convolutional Neural Networks (CNN). With this method, 13 different healthy leaf species were introduced to CNN. So, diseased species were distinguished. As a result of the study, the accuracy rate between 91% and 98% was acquired. Kulkarni et al. [12] utilized Zernike moment features in addition to shape, color, vein, and texture features. As a classifier, Radial Basis Probabilistic Neural Network (RBPNN) was preferred. The accuracy rate of the study using Flavia data was 93.82%. Wagle et al. [32] designed a compact CNN with different convolutional layers, named N1, N2, and N3, for plant species classification. They tested this compact CNN model on the PlantVillage and Flavia datasets. They then compared their results with the AlexNet model with transfer learning. The designed CNN model was superior in terms of training time and accuracy. The classification accuracies achieved were above 99%. Bakhshipour and Jafari [33] presented a proposal for the separation of sugar beet and weed. In the feature extraction phase, shape factors, moment invariants, and Fourier descriptors were extracted from images. ANN and SVM were carried out in the classification phase. In that study, four common weed species were detected in sugar beet. Accuracy rates were 92.92% for ANN and 95.00% for SVM. Finally, in another study by Kho et al. [34], three different Ficus species leaves (F. benjamina, F. pellucidopunctata, and F. sumatrans) were classified. Morphological features, Hu moment features, texture features, and Histogram of Oriented Gradients (HOG) were employed in the feature extraction stage. ANN and SVM were applied to perform classification. As a result of both models, the accuracy rate was found as 83.3%.

In general, previous studies have used different feature extraction, optimization and classification algorithms for leaf classification. In previous studies, it is desired to increase the accuracy of leaf recognition by using different methods together (hybrid). Some studies perform classification with ML, others with DL. The learning algorithm used in leaf classification is vital for accuracy. No comprehensive study involving DL and ML comparison has been carried out so far. The superiority of DL algorithms over ML is known. In this sense, this study equips ML algorithms in the best way and compares them with different CNN models.

III. Dataset

A. Folio Dataset

In this study, the dataset from UCI[35] library is used. The Folio dataset includes 32 different images each belonging to different leaves. Each type has 20 images. Images are obtained from a mobile phone with a 1980x1024 resolution. In this study, 15 different types of leaves from the Folio [18] dataset are classified. The types of leaves are shown in Fig. 1 (a).

B. Swedish Leaf Dataset

The Swedish Leaf dataset [36] is a public dataset created by Linkoping University and the Swedish Museum of Natural History. The dataset contains a total of 15 different leaf types, and each type has 75 sample images. Some species are very similar in shape to each other. The image background is white. The leaf types of the Swedish Leaf dataset are shown in Fig. 1(b).

(a) Folio dataset

(b) Swedish Leaf dataset

Fig.1. Leaf datasets used in the study

IV. Methodology

The methodology part of this study deals with image processing, feature extraction, ML and DL methods. A general flow chart showing the application of these methods in this study is shown in Fig. 2. In this section, each method used is examined in detail and the outputs of the relevant method are shared.

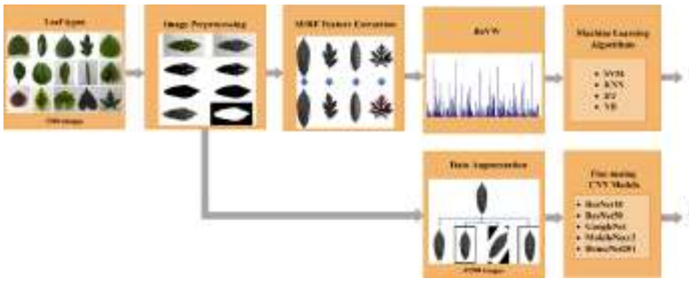


Fig.2. Methods used in the study

A. Image Processing

Most of the time, it is crucial to perform pre-processing operations in the original image to minimize the error of the feature extraction process. Using the advantage of pre-processing, some operations like the extraction of desired features, minimizing the noise can be achieved, and it directly affects the performance of the classification. This study presents a comparative analysis of ML and DL methods, which are frequently preferred in leaf classification applications. The main purpose of this study is to compare the results obtained from ML and DL-based algorithms. However, in order to increase the accuracy and reliability of both ML and DL-based algorithms, it is necessary to examine only leaf-containing pixels. In this sense, some image pre-processing steps are applied for raw leaf images. The image is belonging to the Croton plant, and each pre-processing step is illustrated in the order in Fig. 3.

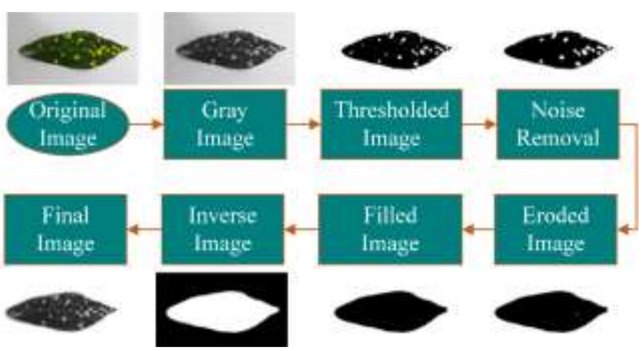


Fig.3. Pre-processing steps

As shown in Fig. 3, the original image is converted to a grayscale image to overcome the complexity derived from color and to minimize the noise. When Fig. 1 (a) is examined, it is obvious that backgrounds are completely white for some images while it is not for others. To improve recognition performance, features should be extracted mostly from the leaf. For this, a noise-free and uniform background should be provided for all leaves. For this reason, the background of all images should be white. The threshold value is determined by the Otsu method [37], and the leaf is completely separated from the background to cover the background with white color. The main purpose of the pre-processing is that the background should be white, and the leaves are gray-scale. For this reason, noise removal, morphological operations, filling, and inversion were performed on the image after the determination of the threshold value of the image. Eventually, the multiplication of achieved inverse image and gray image gives the final image. After this stage, most of the features extracted from the final image belong to the leaf. In summary, as a result of the pre-processing, the original image is subjected to specific processing, and thus, noise is removed, and gray space leaf images are obtained. Each of these processes is first applied to all leaves. The obtained results are recorded, and the features of these images are extracted. The original images and the final images of the Croton plant as a result of pre-processing are shown in Fig. 4.

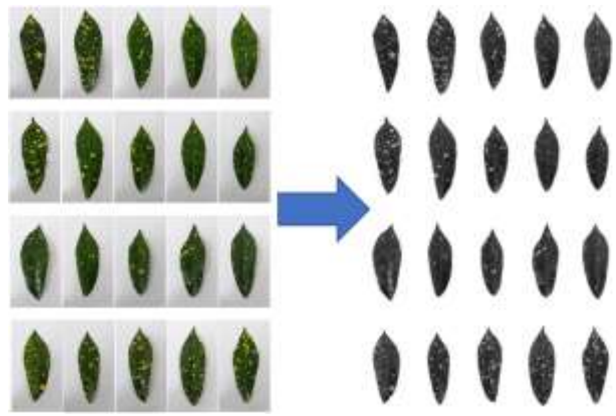


Fig.4. Original and final images of Croton plant

B. K-Means Algorithm

The K-Means algorithm is a frequently used learning method that has a simple structure. In this method, a large number of extracted features are divided into k groups. When the algorithm starts, first of all, centers are determined. Afterwards, data are classified according to these centers. According to the classification results, the center points are constantly updated. This process continues iteratively. Clustering is completed when the error between the specified center point and the data reaches the minimum value [38, 39].

C. Bag of Visual Words

The BoVW algorithm is an adapted version of the BoW (Bag of Words) algorithm to images. Using the BoW algorithm, it is possible to obtain a histogram of the words in a document. In this way, document classification can be performed. Similarly, with the BoVW, images can be classified by identifying visual words in an image and creating histograms [40]. In the BoVW method, the extracted features are clustered first. As a result of clustering, words are obtained, and each cluster represents a word. Finally, histograms are created according to the word frequency in the input image [41, 42].

D. Feature Extraction

In this study, SURF features are extracted from each pre-processed image. SURF is superior to SIFT (Scale Invariant Feature Transform) in terms of faster feature extraction [43]. Since there are 20 pieces from each leaf and the training rate is 80% in the classification, a total of 240 leaves are used to extract features. The total number of extracted features is 947085. In Fig. 5, the features of some different leaves are shown with red marks.

ML or DL models can be fed directly with SURF features. However, more successful results were obtained in studies [44-47] that applied SURF features together with the BoVW model. For the detection of SURF key points, the Hessian matrix is used. With the determinant of the Hessian matrix, it is decided whether a point will be chosen as the key point. Considering a pixel, its Hessian matrix is calculated as follows:

$$H\,(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \qquad (1)$$

In an I image, given the point $\mathbf{x} = [x, y]$, the Hessian matrix $H\,(\mathbf{x}, \sigma)$ at point $\mathbf{x}$ on the scale σ is defined as follows [48]:

$$H\,(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{yx} & L_{yy} \end{bmatrix} \qquad (2)$$

Here $L_{xx}(\mathbf{x}, \sigma)$ is the convolution of the image I at the $\mathbf{x}$ point and the Gaussian quadratic derivative $(\frac{\partial^2}{\partial x^2} G\,(x, y, \sigma))$. Others $(L_{xy},\ L_{yx},\ L_{yy})$ are calculated similarly. Since key point calculations will be made for each pixel, the SURF method is affected by noise in the image and faulty or missing features can be extracted [49-51]. Because SURF features are calculated for each pixel. Therefore, noise in the image and unnecessary blobs in the background should be removed. Various noise removal or denoising methods can be used for this purpose [52]. By removing the noises, only features are extracted from the main object. The number of features extracted with SURF may be different for each image. In other words, the number of pixels that are resistant to rotation and scale change in an image can be different. However, thanks to the use of these features with the BoVW model, the number of features will equate to the number of visual words.

As seen in Fig. 5, the background noise is completely removed, and the SURF features are only on the leaf. These SURF features are then processed in the BoVW algorithm to generate new feature data. In this way, it is aimed to represent more complex, and numerous SURF features in a more meaningful way with BoVW. Features are first clustered with BoVW. In this study, k-means is used as the clustering algorithm, and k value is set to 500. In other words, 947085 features are distributed to 500 clusters in k-means. Each cluster represents a word. The result of BoVW is expressed using histograms according to an image word frequency. However, the histogram only gives the word frequency in an image. Location information is not available on the histogram. The histogram obtained from a sample leaf image is shown in Fig. 6.

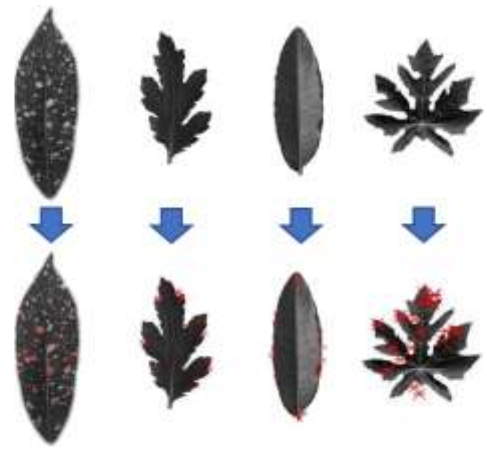


Fig.5. SURF features extracted from pre-processed images

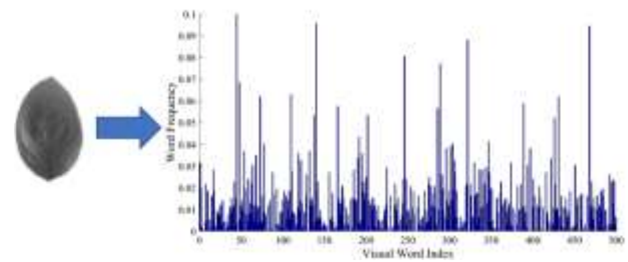


Fig.6. Visual word frequency histogram obtained from a leaf of Ashanti Blood

The histogram obtaining process in Fig. 6 is carried out for each leaf. As a result, a new numerical dataset in the size of 240x500 is generated for 240 images. These data can now be classified by four different ML methods.

E. Machine Learning-Based Leaf Classification

In this study, four different ML methods, DT, KNN, SVM, and NB, respectively are used for the classification of the leaves. In this part, methods are explained briefly.

1) Decision Tree

DT is a supervised learning method that can solve classification and regression problems by using trees, leaves, and branches in a representative manner. The tree structure is formed with DT according to the used dataset. A tree consists of leaves and branches. Leaves represent the label of the class while branches represent the cluster that forms the label class. Forming a tree continues iteratively until there are no criteria to divide the samples. After the forming phase of the tree, it is time for the pruning phase which increases the performance of the classification. A big dataset can be classified successfully by using DT [53-55].

2) k-Nearest Neighbors

The KNN algorithm is an unsupervised learning method that is easy to understand and implement. The algorithm works by considering the distance between the data of known class and a new sample for the classification operation. The number k represents the number of data closest to the new sample, and the class assignment is performed by considering the majority. Contrary to other methods, there is no time lost for the training process, in another saying there is no training process. Although KNN is a classical method, it is still preferred because of its accuracy and speed, and it was used frequently in applications such as pattern recognition and classification [56-58].

3) Support Vector Machine

SVM is a supervised learning method for classification and regression applications because of its high performance. The training process was performed in a multi-dimensional space by creating hyperplanes. The classes are separated with hyperplanes, and for a strong separation, hyperplanes should divide the class with an optimal distance. SVM includes the necessary calculations that form the hyperplanes to provide optimum separation. In addition, a multi-dimensional and non-linear dataset can also be classified with SVM. The kernel trick method is used for this purpose. In this way, the feature space is mapped to a higher space, and afterwards, hyperplanes are created [59-61].

4) Naive Bayes

NB is a probabilistic classification model based on the Bayes theorem. With probability calculations using data of a known class, the class of the new data can be calculated. In this method, classification is made by assuming that the features have connections and relations. One of the most important advantages of NB is that it can perform classification with little training data. Moreover, NB is less affected by noises [62-64].

F. Bayesian Optimization (BO)

BO [65] is a hyperparameter search algorithm. It was developed using Bayes' theorem. With Bayes, the posterior distribution is estimated using prior knowledge. The equation representing the general Bayes theorem is shown in Equation 3. According to Equation 3, posterior distribution ($P(X|Z)$) is directly proportional to likelihood ($P(Z|X)$) and a priori distribution (bias) ($P(X)$) [66].

$$P(X|Z) \propto P(Z|X)P(X) \tag{3}$$
$$x^* = \arg min f(x) \qquad x \in X \tag{4}$$

BO is often used among HO methods. The general calculation formula for this optimization method is as in Equation 4. In Equation 4, the value x^* minimizes the function $f(x)$. Here x represents a hyperparameter of the ML algorithm. $f(x)$ represents an objective function, also called the Gaussian process model, that strives to be minimized. The hyperparameter type x is searched in the X search space. Finding the optimal hyperparameter for ML algorithms in the search space is quite costly. Also, it would be costly to use their values in the whole space to find the optimum value of x as in Equation 4. An iterative approach is required to reduce cost. At this stage, BO converges to the optimum value by applying Bayes-based iterative optimization[67].

G. Deep Learning-Based Leaf Classification

This section presents the data augmentation step applied for leaf classification based on DL. It also briefly introduces the Resnet18 [19], ResNet50 [19], MobileNet [20, 21], GoogLeNet [22] and DenseNet [23] CNN models applied for classification.

1) Data Augmentation

It is a well-known fact that a rich dataset increases the success of DL classification. However, it may not always be possible to obtain a rich dataset. Therefore, the images in the dataset can be artificially augmented to obtain a rich dataset using various techniques [68, 69]. Data augmentation is actions performed on raw images to increase the amount of data in the dataset.

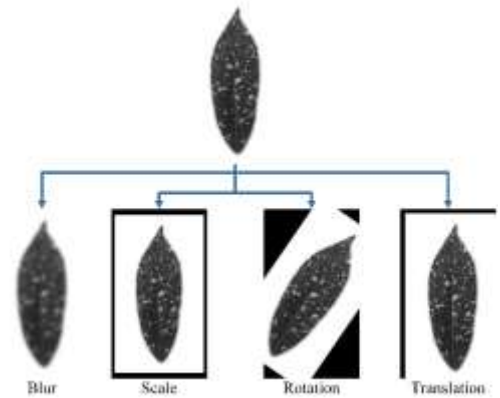


Fig. 7. Data augmentation techniques and sample result images

In this study, there are a total of 300 images, 20 of which belong to each of 15 different leaf types. Since this number of images is insufficient for a DL study, data augmentation techniques are applied. Four different data augmentation techniques (Rotation, Scale, Translation, and Blur) are applied to each image shown in Fig. 7 to ensure data diversity. With the rotation operation, the image is rotated around its center by a certain degree. The translation process is to shift the images right, left, up or down on a pixel basis. With scale operation, the scale of the image is changed at a certain interval, so that the image of the object in different dimensions is created. For blur, a low-pass filter is applied to reduce the high-frequency effect in the image. The total number of images is increased to 1500 by using data augmentation techniques. The lower and upper limit values of these data augmentation methods are shown in Table I.

TABLE I
THE LOWER AND UPPER LIMITS OF THE DATA AUGMENTATION TECHNIQUES USED

Parameter	**Lower Limit**	**Upper Limit**
Blur	1	4
Rotation	-45°	45°
Scale	0.8	1.2
Translation (pixel)	-20	+20

2) ResNet18 and ResNet50

CNN architectures are designed in a deeper structure over time. This complicates these architectures. Also, constantly increasing the depth does not always increase success. ResNet [19] is developed to avoid the vanishing gradient problem, which limits the learning of the network as the depth increases. To achieve this, residual blocks are used to transmit residual values to subsequent layers. ResNet18 and ResNet50 models with different depths such as 18 and 50 were developed using this principle.

3) MobileNet

MobileNet is a deep architecture with low computation and fast processing capability, designed for use on mobile devices. Expensive convolution layers are replaced with 1x1 and 3x3 convolution layers to reduce the number of parameters. In this way, fewer learning parameters are obtained. MobileNetv2, on the other hand, uses inverted residual blocks with bottleneck properties and thus has fewer parameters [70].

4) GoogLeNet

GoogLeNet is a 22-layer architecture developed by researchers at Google. It is the first version of the Inception network (Inception v1). In its architecture, it used an inception model consisting of convolution filters with dimensions of 1x1, 3x3 and 5x5. In this way, while increasing the depth of the network, the computational load is reduced [22].

5) DenseNet

It is similar to the ResNet architecture in general, but denser. Each convolution layer except the first layer receives feature maps of all previous convolution layers. That is, the feature maps of a convolution layer are given as input to all subsequent layers. DenseNet-201 is a deep CNN with a layer depth of 201 [23].

V. Results

This section evaluates the classification performance of ML and DL algorithms. Performance metrics used for this are given in Equations 5-10.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{8}$$

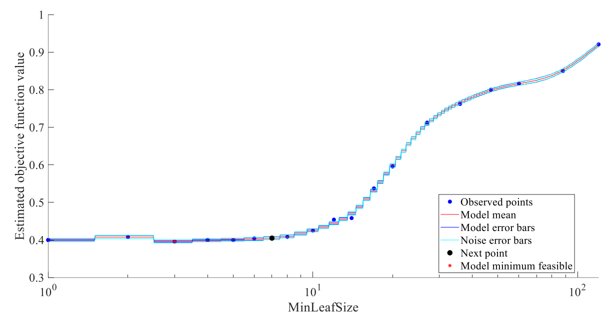
$$\text{F1} - \text{score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{9}$$

TP: True Positive TN: True Negative FP: False Positive FN: False Negative

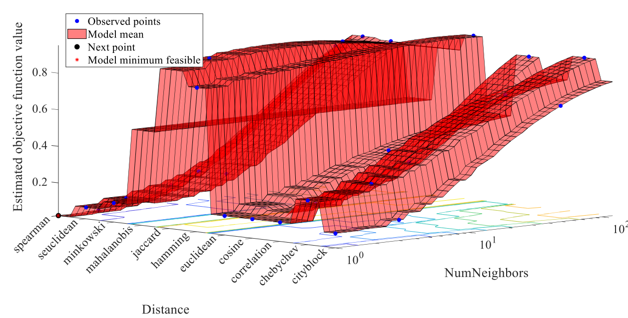
A. ML Results

DT, KNN, SVM, and NB have different hyperparameters that significantly affect the accuracy of the algorithm. When these hyperparameters are determined, optimal success can be obtained from the relevant ML method. With the Hyperparameter Optimization (HO) method, hyperparameter values that make system error minimum are determined [71]. BO [65] approach which is one of the HO methods is used in this study to obtain parameters that would give maximum success rates. For this reason, hyperparameters are first determined by the BO approach and then classification is made by using these hyperparameters. The hyperparameter values used for DT, KNN, NB, and SVM and the time taken to find the optimum hyperparameters (BO duration) are shown in Table II.

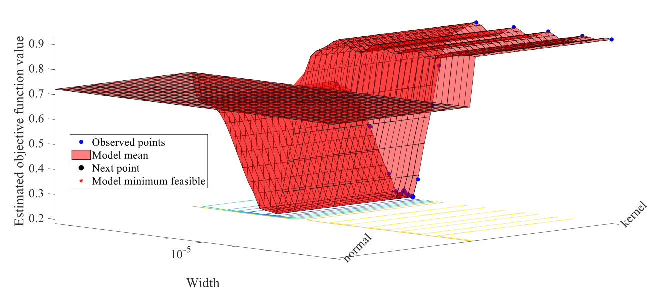
In addition, optimization graphs determining DT, KNN, and NB hyperparameters are shown in Fig. 8. Fig. 8 shows the objective function values according to the changes of the hyperparameters in three different ML algorithms. In these graphs, the values that minimize the objective function are searched in the search space. The hyperparameter value of each ML algorithm is calculated iteratively with BO and the objective function is recalculated accordingly. Instead of calculating the objective function with all space values, BO reaches the optimum value faster with an iterative approach. For each ML algorithm, the hyper parameter values that minimize the graph are recorded and ML classifications are made using these values. For SVM, the four-dimensional BO graph could not be plotted due to the number of variables (objective function, C, kernel scale, and coding method). However, results are obtained. The confusion matrices obtained after the classification using the specified parameter value are shown in Fig. 9, respectively. The metric values in Equations 1-6 are calculated to determine the classification performance. These values are shown in Table III.



(a) DT



(b) KNN



(c) NB

Fig. 8. BO results of ML algorithms

Considering the confusion matrices in Fig. 9 and the performance metrics in Table III, it is seen that KNN and SVM methods are more successful than other methods with an accuracy of 98.09%. Since KNN is the ML method that provides faster training, KNN is a more preferable method in this respect. It should be noted that these results are calculated using the ML parameters obtained with BO, that is, they are the optimum classification results.

TABLE II
HYPERPARAMETERS FOR ML ALG. AND BO DURATION VALUES

ML Alg.	DT	KNN	NB	SVM
Hyperpara-meters	Min. leaf size: 3	Number of neighbors: 1 Distance type: spearman	Distribution type: kernel Kernel width: 0.00308	Coding method: one-versus-all Box Constraints (C): 0.84721 Kernel scale: 0.0336963
BO duration (sec.)	61.86	41.944	3664.401	1616.419

TABLE III
PERFORMANCE METRICS OF ML ALGORITHMS

ML algorithm	**Acc.**	**Spec.**	**Prec.**	**Sens.**	**F1-Score**
DT	0.6536	0.9752	0.7213	0.6536	0.6536
KNN	0.9809	0.9986	0.9851	0.9809	0.9805
NB	0.8963	0.9926	0.9173	0.89635	0.8958
SVM	0.9809	0.9986	0.9851	0.9809	0.9805

B. DL Results

In the next step after data augmentation, five different popular CNN models are fed with augmented leaf images. These are ResNet18, ResNet50, GoogLeNet, DenseNet201, and MobileNetv2, which are frequently used in DL studies. The architecture of each model consists of different numbers and types of layers. The input image size for each model is 224 × 224. Therefore, each image is resized and given to the models. Of 1500 leaf images, 80% were used for training and 20% for testing.

Before training, the dataset is divided into small groups. Learning is done in these small groups. This parameter, called mini-batch size, determines how many data the CNN model can process at the same time. The groups separated by mini-batch are trained sequentially and the weights of the network are updated. The number of times that all training data is given to the network is called the epoch. In this way, the network is repeatedly trained and the weights are updated to reduce the error. The update rate of the weights is determined by the Learning Rate. The parameter values used for training in all CNN models are as follows: Execution Environment: GPU, Max. Epoch: 20, Learning Rate: 0.001, and Mini Batch Size: 32. The optimization algorithm used to train and reduce the loss value is Stochastic Gradient Descent with Momentum (SGDM). Result values are obtained after training and testing for each CNN model. The training of CNN models is carried out on a laptop computer with Intel Core i7-7700HG CPU, NVIDIA GeForce GTX 1050 4 GB graphics card and 16 GB RAM. The confusion matrices obtained according to the ResNet18, ResNet50, GoogLeNet, DenseNet201, and MobileNetv2 classification results are shown in Fig. 10. In addition, accuracy, specificity, precision, F1-score, and sensitivity performance metrics were calculated according to these error matrices and these values are given in Table IV.

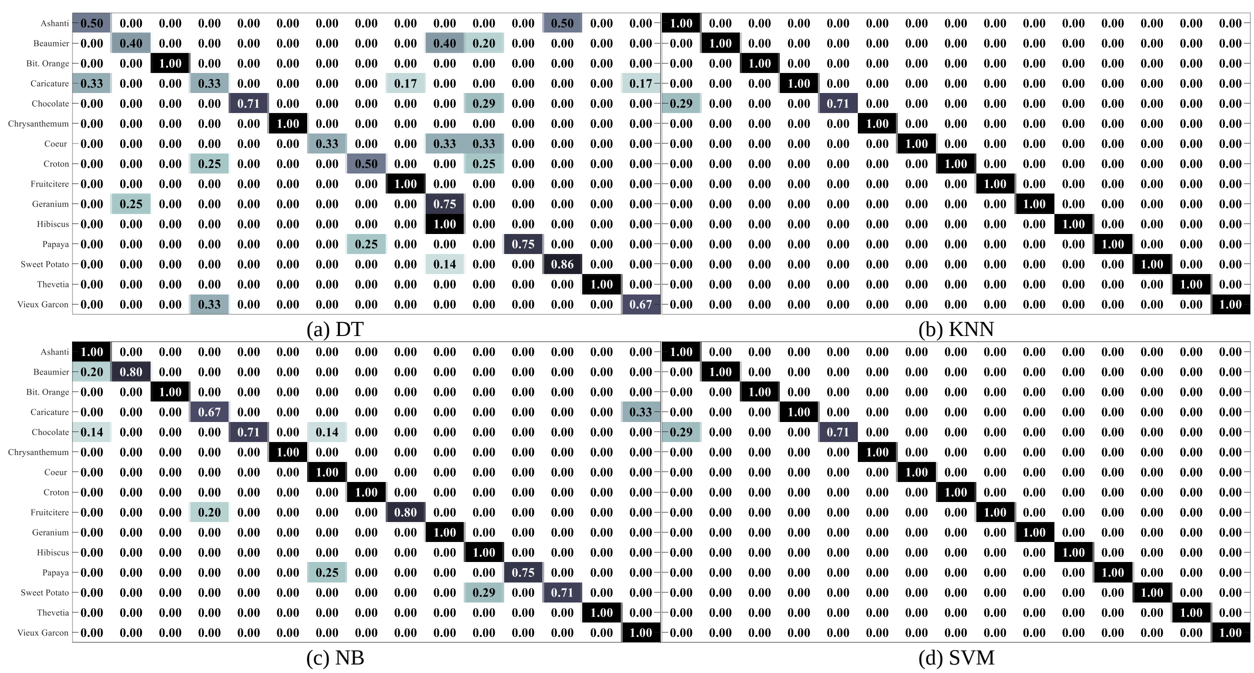
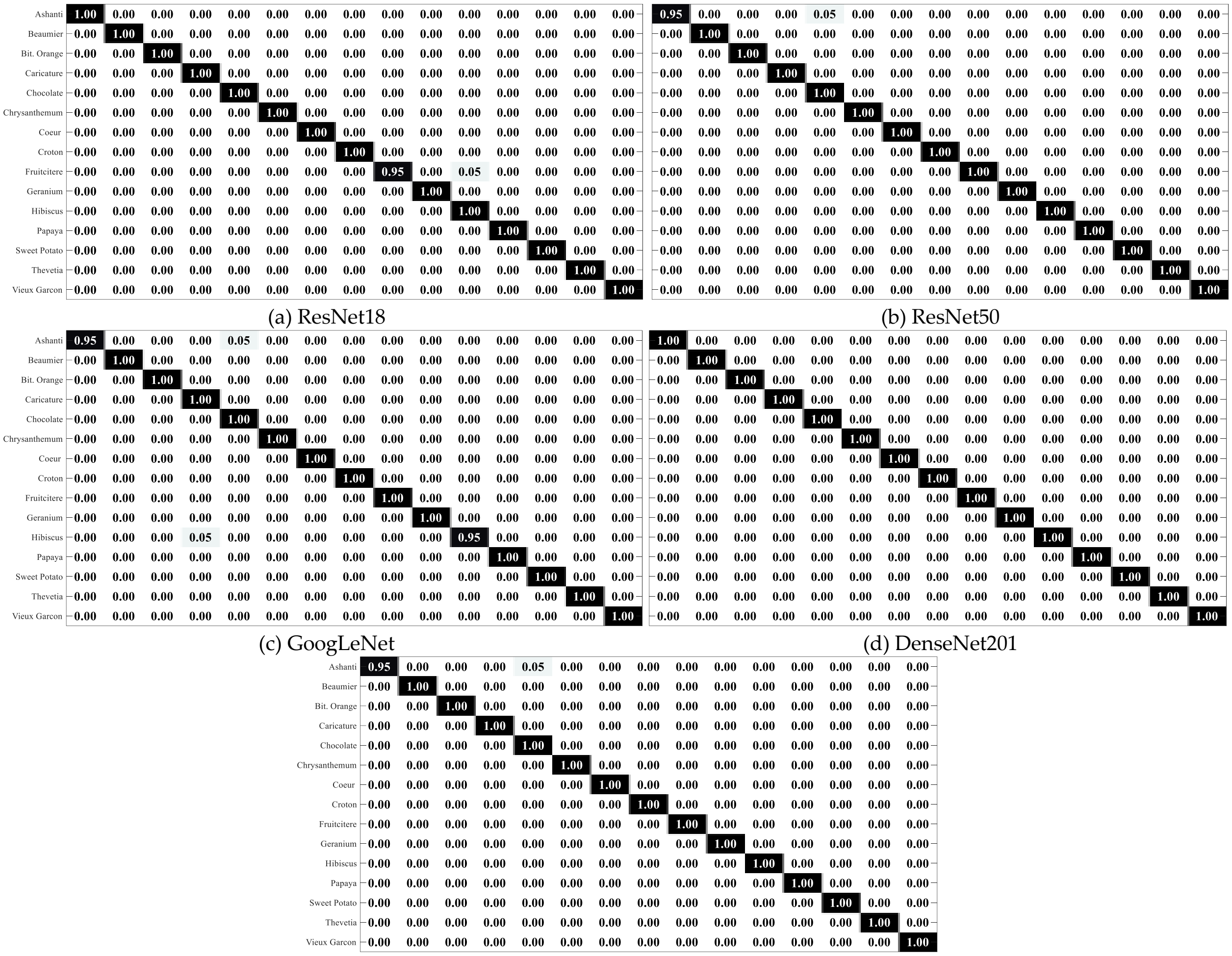


Fig. 9. Confusion matrices of ML algorithms



(a) ResNet18 (b) ResNet50

(c) GoogLeNet (d) DenseNet201

(e) MobileNetv2

Fig. 10. Confusion matrices of DL algorithms

TABLE IV

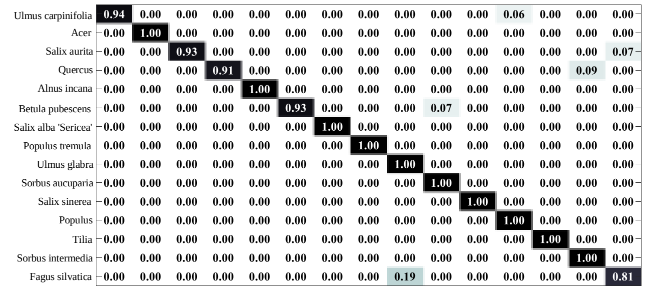
PERFORMANCE METRICS OF CNN MODELS

Model	Acc.	Spec.	Prec.	Sens.	F1-Score
ResNet18	0.9967	0.9998	0.9970	0.9968	0.9968
ResNet50	0.9967	0.9998	0.9974	0.9970	0.9971
GoogLeNet	0.9933	0.9995	0.9923	0.9938	0.9928
DenseNet201	1	1	1	1	1
MobileNetv2	0.9967	0.9998	0.9974	0.9970	0.9971

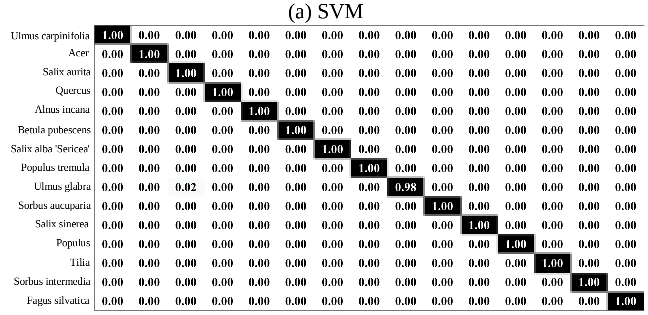
When the confusion matrices in Fig.10 and the classification results in Table IV are examined, it is seen that all the CNN models used provide superior success. DenseNet201, which has the densest connectivity among CNN models, correctly classified all leaves.

In addition, in this study, DL and ML algorithms, which provide the highest classification (SVM for ML, DesneNet201 for DL) are also tested on the Swedish leaf dataset. First, each of the leaf images in the Swedish dataset (see Fig.1(b)) is pre-processed as in the Folio dataset. 80% of these images are used for training and the rest for testing. For ML, SURF features are extracted from the processed images. Then these features are processed with the BoVW method and new word features are created. Afterwards, the BO-optimized SVM method classifies leaf images with 96.89% accuracy. The confusion matrix obtained as a result of classification with SVM is shown in Fig. 11(a). Other metric values are also shown in Table V.

For DL, the number of pre-processed images is increased by the data augmentation techniques shown in Fig. 7. As a result of increasing the data, the total number of processed leaf images increases to 5625. After the training and testing steps, the confusion matrix obtained with the DenseNet201 network for the test data is shown in Fig. 11(b). In addition, other performance metrics are shared in Table V. Accordingly, Swedish leaf dataset images are classified with 99.91% accuracy with DenseNet201. The resulting accuracy and other metrics prove the robustness of the proposed method and support the results obtained on the Folio dataset.



(a) SVM



(b) DenseNet201

Fig. 11. Confusion matrices obtained with DenseNet201and SVM for the Swedish Leaf dataset

TABLE V

PERFORMANCE METRICS OF SVM AND DENSENET201 MODEL FOR THE SWEDISH LEAF DATASET

Model	Acc.	Spec.	Prec.	Sens.	F1-Score
SVM	0.9689	0.9978	0.9706	0.9685	0.9680
DenseNet201	0.9991	0.9999	0.9992	0.9988	0.9990

VI. Conclusion, Discussion and Future Works

In this study, leaf types are classified using ML and DL-based approaches. For a more comprehensive classification, four different ML methods, namely DT, KNN, SVM, and NB, and five different DL methods, ResNet18, ResNet50, GoogLeNet, DenseNet201, and MobileNetv2, are applied. Folio dataset images are used in all these learning-based algorithms. Then the Swedish leaf dataset is also used to prove the robustness of the proposed method. No features are directly extracted from the raw leaf images in this dataset. Firstly, image pre-processing steps are applied to the raw leaf images. Noise and background are removed during pre-processing so that only features are extracted based on the leaves. In this way, the basis for a more successful classification is formed. Pre-processed images are saved, and classifications are performed using these images.

In ML-based classification, the SURF features are extracted from the saved images. Thus, scale-invariant and rotation-invariant features are obtained. However, there are many feature points around 947085 with SURF. BoVW is used to classify these features in a shorter time and to reduce their size by combining them to be more meaningful. SURF features are clustered with BoVW and then histograms are generated according to the number of clusters in each image. That means each image is expressed by the histogram of visual words. K-means is used for clustering. As a result, the histograms are classified by four different ML methods (DT, KNN, SVM, NB). BO method, which is one of the HO algorithms, is used with ML methods to obtain the best results. In this way, optimum hyperparameters are determined for each ML method. As a result, for DT, KNN, SVM, and NB, the classification accuracy rates are 65.36%, 98.09%, 98.09%, and 89.63%, respectively. KNN is a faster learning algorithm than SVM. So, the highest performance in terms of both accuracy and time is provided by KNN.

In the next stage, the types of leaves are classified with different state-of-the-art CNN models. New leaf images have been obtained using data augmentation techniques. In this way, the number of data important for deep network training is increased. Finally, the augmented images were classified using five different CNN models: ResNet18, ResNet50, GoogLeNet, DenseNet201, and MobileNetv2. The classification accuracy rates of these CNN models are 99.67%, 99.67%, 99.33%, 100%, and 99.67%, respectively. In addition, at the end of the study, ML and DL models that successfully classify the Folio dataset are analyzed on the Swedish Leaf dataset. As a result, leaf images in the Swedish Leaf dataset are classified with 99.91% accuracy with DenseNet201 and 96.89% with SVM.

The results show that CNN-based methods for leaf classification are more successful than traditional image processing and ML methods. Finally, it is considered that these results can be useful and encouraging for future studies. As more complex features are obtained from more images with DL, higher accuracy is expected. However, the success of KNN and SVM methods is quite high despite fewer features and fewer images. The combination of SURF, BOVW, and BO methods provided a powerful feature map for ML algorithms.

As a result of both ML and DL experiments, a successful leaf recognition application was carried out for pre-processed leaf images. The highest leaf recognition accuracy was 98.09% with ML and 100% with DL. These accuracies were obtained with KNN/SVM and DenseNet201 models, respectively. The performance of ML methods powered by SURF, BoVW, and BO close to CNN models supports the initial hypothesis.

Despite these successful results, ML and DL applications have some limitations. For a real-time application, extraction of SURF features and BO would cause significant delays and thus ML implementations would be insufficient for practical implementation. Both ML and DL applications should be able to recognize faster. Although data augmentation significantly impacts the success of DL, these images are artificial. Therefore, performing DL experiments on a larger dataset will provide more reliable results.

Considering the implementation and results of ML and DL, ML provided a more erroneous result than DL despite the SURF, BoVW, and BO implementation steps. However, data augmentation was performed for DL as an additional step. Ultimately, however, DL-based CNN models proved to be both an easier and more successful tool for leaf recognition. DL-based solutions will provide stronger recognition if larger datasets are used. However, increasing the accuracy does not only depend on the dataset. The design of the deep architecture for training is also important. For this reason, more successful results can be obtained with different deep architectures in future studies. In addition, CNN-SVM networks, which have come to the fore recently and which extract features with DL and perform classification with SVM, can provide stronger leaf recognition performance. Besides architecture, the fusion of features extracted from leaves in different ways can also enhance recognition performance. In this context, the fusion of DL and ML features may provide a stronger representation. In addition to all these, the current success can be increased by applying the HO applied in ML algorithms to CNN models similarly.

REFERENCES

[1] J. S. Cope, D. Corney, J. Y. Clark, P. Remagnino, and P. Wilkin, “Plant species identification using digital morphometrics: A review,” Expert Systems with Applications, vol. 39, no. 8, pp. 7562-7573, 2012.

[2] Z.-Q. Zhao, L.-H. Ma, Y.-m. Cheung, X. Wu, Y. Tang, and C. L. P. Chen, “ApLeaf: An efficient android-based plant leaf identification system,” Neurocomputing, vol. 151, pp. 1112-1119, 2015.

[3] B. Harish, A. Hedge, O. Venkatesh, D. Spoorthy, and D. Sushma, "Classification of plant leaves using Morphological features and Zernike moments." pp. 1827-1831.

[4] C. Zhao, S. S. Chan, W.-K. Cham, and L. Chu, “Plant identification using leaf shapes—A pattern counting approach,” Pattern Recognition, vol. 48, no. 10, pp. 3203-3215, 2015.

[5] X. Wang, J. Liang, and F. Guo, “Feature extraction algorithm based on dual-scale decomposition and local binary descriptors for plant leaf recognition,” Digital Signal Processing, vol. 34, pp. 101-107, 2014.

[6] K. K. Thyagharajan, and I. Kiruba Raji, “A Review of Visual Descriptors and Classification Techniques Used in Leaf Species Identification,” Archives of Computational Methods in Engineering, vol. 26, no. 4, pp. 933-960, 2019/09/01, 2019.

[7] J. Chaki, R. Parekh, and S. Bhattacharya, “Plant leaf classification using multiple descriptors: A hierarchical approach,” Journal of King Saud University - Computer and Information Sciences, vol. 32, no. 10, pp. 1158-1172, 2020/12/01/, 2020.

[8] A. Aakif, and M. F. Khan, “Automatic classification of plants based on their leaves,” Biosystems Engineering, vol. 139, pp. 66-75, 2015.

[9] L. Longlong, J. M. Garibaldi, and H. Dongjian, “Leaf classification using multiple feature analysis based on semi-supervised clustering,” Journal of Intelligent & Fuzzy Systems, vol. 29, no. 4, pp. 1465-1477, 2015.

[10] J. Su, M. Wang, Z. Wu, and Q. Chen, “Fast Plant Leaf Recognition Using Improved Multiscale Triangle Representation and KNN for Optimization,” IEEE Access, vol. 8, pp. 208753-208766, 2020.

[11] K. J. Gaston, and M. A. O'Neill, “Automated species identification: why not?,” Philosophical Transactions of the Royal Society of London B: Biological Sciences, vol. 359, no. 1444, pp. 655-667, 2004.

[12] A. Kulkarni, H. Rai, K. Jahagirdar, and P. Upparamani, “A leaf recognition technique for plant classification using RBPNN and Zernike moments,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 1, pp. 984-988, 2013.

[13] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares, "Leafsnap: A computer vision system for automatic plant species identification," Computer vision–ECCV 2012, pp. 502-516: Springer, 2012.

[14] J. Wäldchen, M. Rzanny, M. Seeland, and P. Mäder, “Automated plant species identification—Trends and future directions,” PLoS computational biology, vol. 14, no. 4, pp. e1005993, 2018.

[15] D. Hall, C. McCool, F. Dayoub, N. Sunderhauf, and B. Upcroft, "Evaluation of features for leaf classification in challenging conditions." pp. 797-804.

[16] B. Liu, and R. Bruch, “Weed detection for selective spraying: A review,” Current Robotics Reports, vol. 1, no. 1, pp. 19-26, 2020.

[17] M. F. Aslan, A. Durdu, K. Sabanci, E. Ropelewska, and S. S. Gültekin, “A Comprehensive Survey of the Recent Studies with UAV for Precision Agriculture in Open Fields and Greenhouses,” Applied Sciences, vol. 12, no. 3, pp. 1047, 2022.

[18] T. Munisami, M. Ramsurn, S. Kishnah, and S. Pudaruth, “Plant leaf recognition using shape features and colour histogram with K-nearest neighbour classifiers,” Procedia Computer Science, vol. 58, pp. 740-747, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." pp. 770-778.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks." pp. 4510-4520.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions." pp. 1-9.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." pp. 4700-4708.

[24] T. Jin, X. Hou, P. Li, and F. Zhou, “A novel method of automatic plant species identification using sparse representation of leaf tooth features,” PLoS One, vol. 10, no. 10, pp. e0139482, 2015.

[25] B. Vijaya Lakshmi, and V. Mohan, “Plant leaf image detection method using a midpoint circle algorithm for shape-based feature extraction,” Journal of Modern Applied Statistical Methods, vol. 16, no. 1, pp. 26, 2017.

[26] M. Koklu, M. F. Unlersen, I. A. Ozkan, M. F. Aslan, and K. Sabanci, “A CNN-SVM study based on selected deep features for grapevine leaves classification,” Measurement, vol. 188, pp. 110425, 2022/01/01/, 2022.

[27] G. Sharma, A. Kumar, N. Gour, A. K. Saini, A. Upadhyay, and A. Kumar, “Cognitive framework and learning paradigms of plant leaf classification using artificial neural network and support vector machine,” Journal of Experimental & Theoretical Artificial Intelligence, pp. 1-26, 2022.

[28] Y. Arun, and G. S. Viknesh, "Leaf Classification for Plant Recognition Using EfficientNet Architecture." pp. 1-5.

[29] G. Saleem, M. Akhtar, N. Ahmed, and W. S. Qureshi, “Automated analysis of visual leaf shape features for plant classification,” Computers and Electronics in Agriculture, vol. 157, pp. 270-280, 2019/02/01/, 2019.

[30] B. Dudi, and V. Rajesh, “Optimized threshold-based convolutional neural network for plant leaf classification: a challenge towards untrained data,” Journal of Combinatorial Optimization, vol. 43, no. 2, pp. 312-349, 2022/03/01, 2022.

[31] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep neural networks based recognition of plant diseases by leaf image classification,” Computational intelligence and neuroscience, vol. 2016, 2016.

[32] S. A. Wagle, R. Harikrishnan, S. H. M. Ali, and M. Faseehuddin, “Classification of Plant Leaves Using New Compact Convolutional Neural Network Models,” Plants, vol. 11, no. 1, pp. 24, 2022.

[33] A. Bakhshipour, and A. Jafari, “Evaluation of support vector machine and artificial neural networks in weed detection using shape features,” Computers and Electronics in Agriculture, vol. 145, pp. 153-160, 2018.

[34] S. J. Kho, S. Manickam, S. Malek, M. Mosleh, and S. K. Dhillon, “Automated plant identification using artificial neural network and support vector machine,” Frontiers in Life Science, vol. 10, no. 1, pp. 98-107, 2017.

[35] "UCI Machine Learning Repository," 07.05.2022, 2022; https://archive.ics.uci.edu.

[36] "Swedish Leaf Dataset," 10.08.2022, 2022; https://www.cvl.isy.liu.se/en/research/datasets/swedish-leaf/.

[37] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, 1979.

[38] A. K. Jain, “Data clustering: 50 years beyond K-means,” Pattern recognition letters, vol. 31, no. 8, pp. 651-666, 2010.

[39] K. R. Žalik, “An efficient k′-means clustering algorithm,” Pattern Recognition Letters, vol. 29, no. 9, pp. 1385-1391, 2008.

[40] H. Kato, and T. Harada, "Image reconstruction from bag-of-visual-words." pp. 955-962.

[41] A. Bosch, X. Muñoz, and R. Martí, “Which is the best way to organize/classify images by content?,” Image and vision computing, vol. 25, no. 6, pp. 778-791, 2007.

[42] S. Xu, T. Fang, D. Li, and S. Wang, “Object classification of aerial images with bag-of-visual words,” IEEE Geoscience and Remote Sensing Letters, vol. 7, no. 2, pp. 366-370, 2010.

[43] M. Paul, R. K. Karsh, and F. A. Talukdar, "Image hashing based on shape context and speeded up robust features (SURF)." pp. 464-468.

[44] F. Baig, Z. Mehmood, M. Rashid, M. A. Javid, A. Rehman, T. Saba, and A. Adnan, “Boosting the performance of the BoVW model using SURF–CoHOG-based sparse features with relevance feedback for CBIR,” Iranian Journal of Science and Technology, Transactions of Electrical Engineering, vol. 44, no. 1, pp. 99-118, 2020.

[45] M. Ghalan, and R. K. Aggarwal, "Multifold Classification for Human Action Recognition." pp. 1-6.

[46] T. Sunitha, and T. Sivarani, “Novel content based medical image retrieval based on BoVW classification method,” Biomedical Signal Processing and Control, vol. 77, pp. 103678, 2022.

[47] E. R. Vimina, and K. P. Jacob, “Feature fusion method using BoVW framework for enhancing image retrieval,” IET Image Processing, vol. 13, no. 11, pp. 1979-1985, 2019.
[48] P. Panchal, S. Panchal, and S. Shah, “A comparison of SIFT and SURF,” International Journal of Innovative Research in Computer and Communication Engineering, vol. 1, no. 2, pp. 323-327, 2013.
[49] N. Y. Khan, B. McCane, and G. Wyvill, "SIFT and SURF Performance Evaluation against Various Image Deformations on Benchmark Dataset." pp. 501-506.
[50] E. Karami, S. Prasad, and M. Shehata, “Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images,” arXiv preprint arXiv:1710.02726, 2017.
[51] S. Routray, A. K. Ray, and C. Mishra, "Analysis of various image feature extraction methods against noisy image: SIFT, SURF and HOG." pp. 1-5.
[52] L. Cao, J. Ling, and X. Xiao, “Study on the Influence of Image Noise on Monocular Feature-Based Visual SLAM Based on FFDNet,” Sensors, vol. 20, no. 17, pp. 4922, 2020.
[53] D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan, “Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks,” Expert Systems with Applications, vol. 41, no. 4, pp. 1937-1946, 2014.
[54] N. Patel, and S. Upadhyay, “Study of various decision tree pruning methods with their empirical comparison in WEKA,” International journal of computer applications, vol. 60, no. 12, 2012.
[55] A. Priyama, R. G. Abhijeeta, A. Ratheeb, and S. Srivastavab, “Comparative analysis of decision tree classification algorithms,” International Journal of Current Engineering and Technology, vol. 3, no. 2, pp. 334-337, 2013.
[56] N. Bhatia, “Survey of nearest neighbor techniques,” arXiv preprint arXiv:1007.0085, 2010.
[57] T. Cover, and P. Hart, “Nearest neighbor pattern classification,” IEEE transactions on information theory, vol. 13, no. 1, pp. 21-27, 1967.
[58] J. Kim, B.-S. Kim, and S. Savarese, "Comparing image classification methods: K-nearest-neighbor and support-vector-machines." pp. 133-138.
[59] A. Ben-Hur, and J. Weston, "A user’s guide to support vector machines," Data mining techniques for the life sciences, pp. 223-239: Springer, 2010.
[60] N. Reljin, and D. Pokrajac, "Classification of performers using support vector machines." pp. 165-169.
[61] X. Yang, R. Zhang, Z. Zhai, Y. Pang, and Z. Jin, “Machine learning for cultivar classification of apricots (Prunus armeniaca L.) based on shape features,” Scientia Horticulturae, vol. 256, pp. 108524, 2019.
[62] J. Abellán, and J. G. Castellano, “Improving the Naive Bayes Classifier via a Quick Variable Selection Method Using Maximum of Entropy,” Entropy, vol. 19, no. 6, pp. 247, 2017.
[63] M. Ahmed, M. Shahjaman, M. Rana, M. Mollah, and N. Haque, “Robustification of Naïve Bayes Classifier and Its Application for Microarray Gene Expression Data Analysis,” BioMed research international, vol. 2017, 2017.
[64] S. Mukherjee, and N. Sharma, “Intrusion detection using naive Bayes classifier with feature reduction,” Procedia Technology, vol. 4, pp. 119-128, 2012.
[65] Y. Ozaki, M. Yano, and M. Onishi, “Effective hyperparameter optimization using Nelder-Mead method in deep learning,” IPSJ Transactions on Computer Vision and Applications, vol. 9, no. 1, pp. 20, 2017.
[66] M. F. Aslan, A. Durdu, A. Yusefi, K. Sabanci, and C. Sungur, “A tutorial: Mobile robotics, SLAM, bayesian filter, keyframe bundle adjustment and ROS applications,” Robot Operating System (ROS), pp. 227-269, 2021.
[67] M. F. Aslan, K. Sabanci, A. Durdu, and M. F. Unlersen, “COVID-19 diagnosis using state-of-the-art CNN architecture features and Bayesian Optimization,” Computers in Biology and Medicine, vol. 142, pp. 105244, 2022/03/01/, 2022.
[68] A. Mikołajczyk, and M. Grochowski, "Data augmentation for improving deep learning in image classification problem." pp. 117-122.
[69] C. Shorten, and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” Journal of Big Data, vol. 6, no. 1, pp. 1-48, 2019.
[70] J.-E. Kim, N.-E. Nam, J.-S. Shim, Y.-H. Jung, B.-H. Cho, and J. J. Hwang, “Transfer Learning via Deep Neural Networks for Implant Fixture System Classification Using Periapical Radiographs,” Journal of Clinical Medicine, vol. 9, no. 4, pp. 1117, 2020.
[71] T. Hinz, N. Navarro-Guerrero, S. Magg, and S. Wermter, “Speeding up the Hyperparameter Optimization of Deep Convolutional Neural Networks,” International Journal of Computational Intelligence and Applications, pp. 1850008, 2018.

BIOGRAPHY

MUHAMMET FATIH ASLAN received his B.Sc. and master's degrees from Selcuk University (SU), Konya, Turkey in 2016 and 2018, respectively. He completed his Ph.D. in Konya Technical University (KTUN) in 2022, in the department of electrical and electronics engineering. He has been working as a researcher at Karamanoglu Mehmetbey University (KMU) since 2017. His research interests include mobile robotics, real-time autonomous applications, image processing, machine learning, computer vision and object tracking.