



Research article

Feature fusion based artificial neural network model for disease detection of bean leaves

Eray Önlér*

Tekirdag Namik Kemal University, Faculty of Agriculture, Biosystem Engineering Department, Suleymanpasa, Tekirdag, Turkey

* **Correspondence:** Email: erayonler@nku.edu.tr; Tel: +905457687128.

Abstract: Plant diseases reduce yield and quality in agricultural production by 20–40%. Leaf diseases cause 42% of agricultural production losses. Image processing techniques based on artificial neural networks are used for the non-destructive detection of leaf diseases on the plant. Since leaf diseases have a complex structure, it is necessary to increase the accuracy and generalizability of the developed machine learning models. In this study, an artificial neural network model for bean leaf disease detection was developed by fusing descriptive vectors obtained from bean leaves with HOG (Histogram Oriented Gradient) feature extraction and transfer learning feature extraction methods. The model using feature fusion has higher accuracy than only HOG feature extraction and only transfer learning feature extraction models. Also, the feature fusion model converged to the solution faster. Feature fusion model had 98.33, 98.40 and 99.24% accuracy in training, validation, and test datasets, respectively. The study shows that the proposed method can effectively capture interclass distinguishing features faster and more accurately.

Keywords: leaf disease; artificial neural network; feature fusion; deep neural networks; computer vision

1. Introduction

In order to feed the world population, productivity in agricultural production should be increased by 60% annually until 2050 [1]. Plant diseases are one of the most important agricultural problems affecting crop yield and quality. Studies have shown that plant diseases reduce crop yields by 20–40% [2]. Most of the strategies used to prevent plant diseases today are based on the use of

chemical pesticides [2]. It has been determined that 78–79% overdose pesticides are used in pesticide applications without considering the plant need and the prevalence of the disease [3]. Excessive use of chemicals causes the emergence of disease and pest species that are more resistant to pesticides [4,5]. In addition, due to climate change, there are changes in the emergence time and periods of diseases [6–8]. For these reasons, there is a need for systems that will detect the location of the disease and apply chemicals only to these areas, thus preventing excessive pesticide use.

According to past studies, 42% of agricultural production is in loss due to plant leaf diseases [9]. Detection of leaf diseases is commonly made with the diagnosis of an experienced plant pathologist. However, this method can be used in a limited way due to the small number of specialists and the slow diagnosis made in this way. Complex systems are frequently modelled using machine learning techniques [10,11]. The complexity of agriculture makes machine learning techniques well-suited for detecting leaf diseases with expert-level precision and significantly faster speed. [12]. Since there are many different diseases, variations within the disease, and plant varieties, more studies are needed to obtain a generalizable leaf disease classification solution in this regard [13]. The generalization of a machine learning model is that the classification success achieved during training and validation is also sustainable on new samples that the model has never seen [14].

Artificial neural network models for computer vision need very large image datasets for training [15]. However, collecting and tagging large image datasets is expensive in terms of both time and money [16]. Methods such as transfer learning, data regularization and data augmentation are used to obtain a model that can generalize the maximum from the available data [17–19]. In addition, it has been seen in various studies that fusion of descriptive features obtained by different methods can increase classification accuracy and generalization success, and that the machine learning model will be converged to the solution faster [20–22].

1.1. Literature review

Chen et al. [23] used AlexNet modified architecture-based CNN on the Android platform to predict tomato leaf diseases using training and testing data containing 18,345 and 4585 images respectively. The best model accuracy is 98%.

Fan et al. [24] presented a general framework for identifying plant diseases by fusing deep feature descriptors and traditional handcrafted features. Extensive experiments have been conducted to validate the efficacy of the proposed method, which achieves classification accuracies of 99.79, 92.59 and 97.12% on three datasets (Two apple leaf datasets and one coffee leaf dataset). They used InceptionV3 architecture as deep feature descriptors.

Elfatimi et al. [25] suggested a method for classifying bean leaf diseases based on a publicly available bean leaf image dataset using MobileNetV2 model architecture as a deep feature extractor. The trained model achieved 97% accuracy on training data and 92% accuracy on test data.

Harakannavar et al. [26] conducted a research by using machine learning and image processing to detect leaf diseases in tomato plants. The extracted features are classified using machine learning approaches such as Support Vector Machine (SVM), Convolutional Neural Network (CNN) and K-Nearest Neighbor (K-NN). The accuracy of the proposed model is tested using SVM (88%), K-NN (97%) and CNN (99.6%). The proposed model extracts informative features using computer vision techniques such as RGB conversion, Histogram Equalization, K-means clustering, contour tracing, Discrete Wavelet Transform, Principal Component Analysis, and GLCM (gray level co-occurrences

matrix). They suggest that they can improve accuracy by using fusion techniques in their future works.

Annrose et al. [27] proposed a hybrid deep learning model with an Archimedes optimization algorithm (HDL-AOA) for the classification of soybean diseases. The hybrid deep learning strategy is built of Wavelet packet decomposition (WPD) and long short-term memory (LSTM). The WPD disassembles the input pictures into four subseries, and four LSTM networks were created. The HDL-AOA model is applied in a cloud-based framework for collaboration and achieves a lower MAPE (mean absolute percentage error) than other existing techniques such as RNN, DCNN, LSTM and CNN. The suggested HDL-AOA model has an accuracy of 98.23%.

Singh et al. [28] employed an artificial intelligence and computer vision approaches to construct and develop an intelligent leaf disease classification system. The PlantVillage data set (for apple, maize, potato, tomato, and rice plants) photos are augmented, and deep features are retrieved using a convolutional neural network (CNN). The picture was then subjected to preprocessing and feature extraction via color moments, HOG, and GLCM. For the selection of these hybrid features, binary particle swarm optimization is utilized, followed by random forest classification for classification results. Five convolutional neural network architectures were trained and evaluated, namely LeNet, ShuffleNet, AlexNet, EffNet, and MobileNet. MobileNet achieved the highest accuracy (96.1%).

1.2. Proposed method

In this study, we aimed to detect the disease in bean leaves, which is one of the commonly produced cultivars, via digital images. For this purpose, we used an image dataset consisting of healthy, bean rust and angular leaf spot classes for the training of the machine learning model. In order to increase the classification accuracy of the developed model and the generalization success on the test dataset, an artificial neural network model was created and trained by fusion of descriptive vectors obtained by transfer learning feature extraction and HOG feature extraction methods. CNN-based machine learning models trained in the detection of leaf diseases will converge to the solution in less time and with more accuracy by employing fewer parameters according to the proposed method.

2. Materials and methods

2.1. Dataset

The bean leaf dataset consists of images taken with smartphone cameras. The captured images consist of bean rust, angular leaf spot and healthy classes (Figure 1). The dataset was collected by the Makerere AI research lab and tagged by experts at the National Crops Resources Research Institute (NaCRRI) [29]. Images are 500×500 pixels in RGB (red-green-blue) digital format.

We divided the image dataset into training, test and validation datasets to be used in the training, validation and testing of the machine learning model. There are 1034 images in the training dataset, 128 images in the test dataset, and 134 images in the validation dataset. The total number of images in the healthy, bean rust and angular leaf spot classes are given in Table 2. We used training and validation datasets in the training of the machine learning model. We tested the developed model using a test dataset that has not been used in training before.

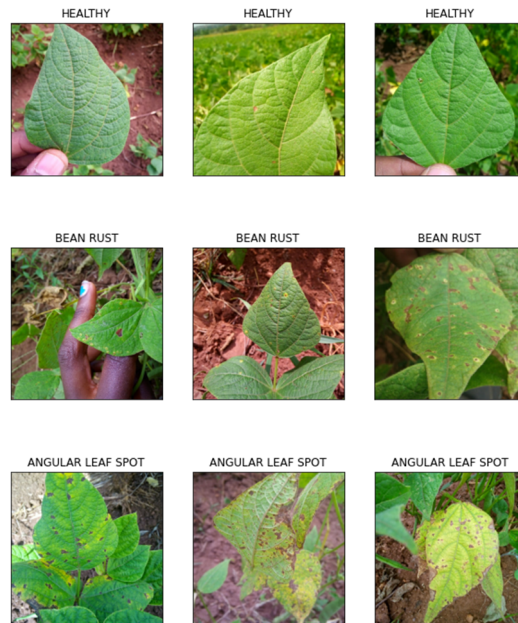


Figure 1. Image samples from the dataset.

Table 2. Number of images by class.

Healthy	Bean rust	Angular leaf spot
428	436	432

2.2. Data augmentation

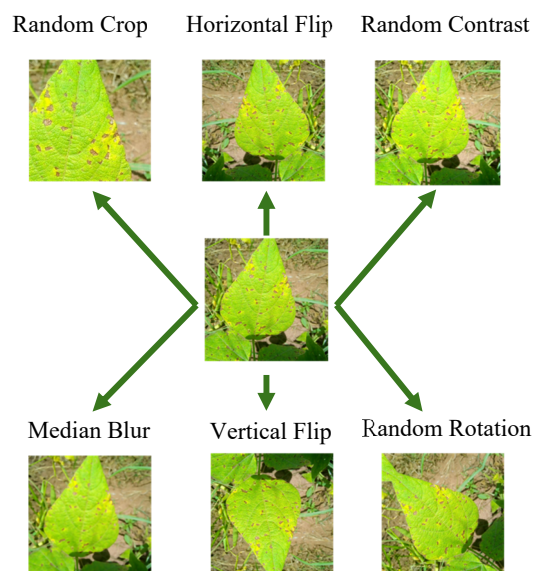


Figure 2. Some results of data augmentation applied on the original image.

Data augmentation is a technique to improve the diversity of datasets. It is used in machine

learning and deep learning to increase the size of training data, which can be very useful in cases where the training dataset is too small. Data augmentation involves adding more samples to the original dataset so that each sample has a different representation from the others. This can be done by translating/rotating/cropping/scaling/randomly changing pixels of an image [30]. In the data augmentation phase, each image in the training and validation dataset was subjected to randomly selected augmentation process (es) before entering to the artificial neural network (Figure 2). Test dataset representing real-world, unseen data. Consequently, we did not apply data augmentation to the test dataset for this reason.

2.3. HOG feature extraction

The histogram of oriented gradients (HOG) is an image feature descriptor and widely used in computer vision applications [31]. It simplifies images by discarding extraneous information and retaining only local texture and appearance information. Since diseased areas on the leaf have different local texture characteristics from healthy areas, HOG is effective for use in disease detection. The HOG function in the scikit-image 0.19.3 library was used with its default settings for HOG feature extraction (Figure 3) [32].

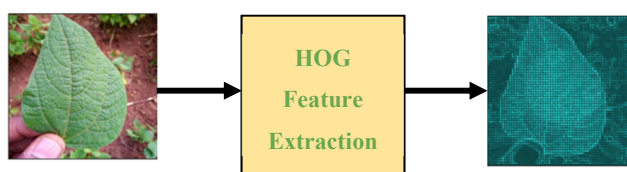


Figure 3. HOG feature extraction.

The calculation in HOG feature extraction was performed as follows.

1) RGB image ($R(r, c)$, $G(r, c)$, $B(r, c)$) (r : row, c : column) firstly converted to grayscale image ($I(r, c)$)

$$I(r, c) = 0.299 \times R(r, c) + 0.587 \times G(r, c) + 0.114 \times B(r, c) \quad (1)$$

2) The grayscale image was normalized with gamma correction to reduce the effects of change in brightness.

$$I(r, c) = I(r, c)^{1/2} \quad (2)$$

3) Gradients were calculated separately for each pixel in the horizontal ($G_x(r, c)$) and vertical ($G_y(r, c)$) directions.

$$G_x(r, c) = I(r, c + 1) - I(r, c - 1) \quad (3)$$

$$G_y(r, c) = I(r - 1, c) - I(r + 1, c) \quad (4)$$

4) Then, the magnitude (μ) and direction angle (θ) values that make up the gradient matrix were calculated for the gradient values in the horizontal and vertical directions.

$$\mu = \sqrt{G_x^2 + G_y^2} \quad (5)$$

$$\theta = |\tan^{-1}(G_y/G_x)| \quad (6)$$

5) The gradient matrix was divided into blocks of 8x8 cells and 9 bins histograms (corresponding to 20° per bin) were generated for each cell in the block. The bin that the cell belongs to was calculated according to the direction angle. The value in each bin was calculated according to magnitude. By summing these values, the histogram of that block was found and histogram matrix was created.

6) After the histogram matrix calculation was completed, the blocks in the histogram matrix were scanned with a 3 × 3 kernel so that the kernel shifted one block after each calculation. The histograms of the 9 blocks in the 3x3 kernel were combined and a feature vector (f_{b_i}) of 81 × 1 size was created for each step.

$$f_{b_i} = [b_1, b_2, b_3, \dots, b_{81}] \quad (7)$$

7) All feature vectors are normalized with the L2 norm

$$f_{b_i} = \frac{f_{b_i}}{\sqrt{\|f_{b_i}\|^2 - \varepsilon}} \quad (8)$$

The HOG feature extraction of the images of Figure 1 are given in Figure 4. We performed HOG feature extraction of all images in the dataset Later, we used these images in the training of the artificial neural network.

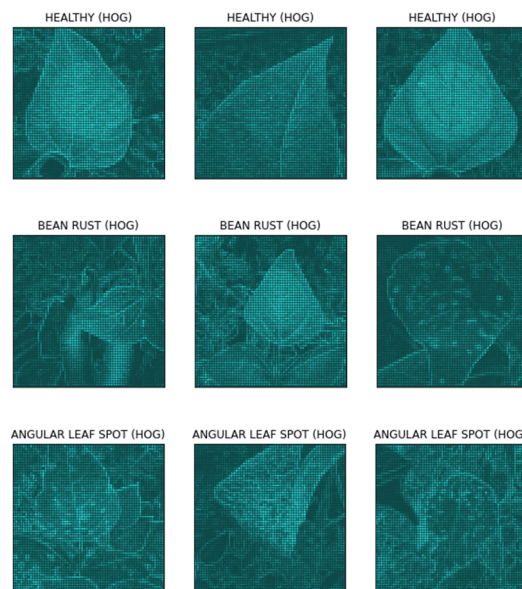


Figure 4. Visualization of HOG feature extraction samples.

2.4. Artificial neural networks

Artificial neural networks are a type of machine learning algorithm inspired by the way the human brain works. Artificial neural networks are composed of layers of neurons, with each neuron connecting to the neurons in the previous layer. This allows for the information processed in one layer to be passed on to subsequent layers, helping the network learn how to solve certain problems by recognizing patterns in the data [33].

We used TensorFlow 2.10.0 library to create artificial neural networks and NVIDIA A100 SXM4 40 GB GPU to accelerate training. We coded with Python 3.8.8 in Google Colab environment. We used the following layers in the artificial neural network models: Convolutional layer: A 2-D convolutional layer applies sliding convolutional filters to two-dimensional input. The convolution operation is performed by sliding the filters along the input horizontally and vertically, and calculating the dot product of the weights and input. A bias term is added afterwards. Dense layer: A dense layer, also known as a fully connected layer, connects every neuron in the preceding layer to every neuron in the present one. Flatten layer: These layers flatten the multi-dimensional input tensors into a single dimensional array for inputting it to the next layer. Batch normalization layer: Each input is normalized and zero-centered across all mini-batches by batch normalization layer. This layer is helping to reduce the risk of vanishing/exploding gradients problems at the artificial neural network training. Dropout layer: The Dropout layer randomly sets the input units to 0 with a frequency of rate at each step during training time, which prevents overfitting. Global Average Pooling 2D layer: A 2-D global average pooling layer computes a mean of the height and width dimensions of the input to downsample it. Concatenation layer: The concatenation layer concatenates a list of inputs [34].

A loss function is a function that compares the target (y_i) and predicted output (\hat{y}_i) values; measures how well the artificial neural network models the training data. Because we have sparse labels we choose sparse categorical cross-entropy as a loss function.

$$Loss = - \sum_{i=1}^{Output\ Size} y_i \times \log \hat{y}_i \quad (9)$$

Optimizers are algorithms used to update the parameters of an artificial neural network such as weights and biases to reduce the losses. Adam optimizer [35] used in all trials.

An activation function in an artificial neural network defines how the weighted sum of the input (x) is transformed into an output. ReLU (rectified linear unit) activation function is used in conv and dense layers in all models. In the last layer, the softmax activation function (σ) is used to calculate which class the sample will belong to (k) based on feature vector (\vec{z}).

$$ReLU = f(x) = x^+ = \max(0, x). \quad (10)$$

$$Softmax = \sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (11)$$

Accuracy is a metric that generally describes how the model performs across all classes. It is calculated as the ratio between the number of correct predictions to the total number of predictions.

$$Accuracy = \frac{Correct\ predictions}{Total\ predictions} \quad (12)$$

2.5. Artificial neural network with HOG feature extraction

We created the artificial neural network model using HOG feature extraction as in Figure 5. Images already have compressed information as they are converted with HOG transform. For this reason, we created a simpler artificial neural network structure. After the RGB images were converted to HOG images, the data was subjected to augmentation and the artificial neural network was trained with these images. We selected batch size as 32 and the training of the model continued for 100 epochs. We used sparse categorical cross-entropy as the loss function and Adam as the optimizer by setting the learning rate to 0.001. We used the softmax activation function in the output layer, and the ReLU activation function in the dense and convolution layers.

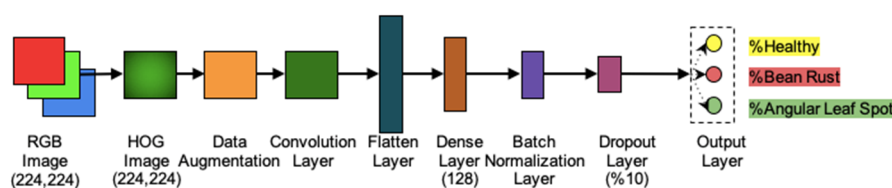


Figure 5. Structure of artificial neural network with HOG feature extraction.

2.6. Artificial neural network with transfer learning feature extraction

Transfer learning allows us to build better performing artificial neural network models faster and with a better accuracy by using previously pre-trained models. In transfer learning, we used a pre-trained model which is trained on a large and general enough dataset to serve as a generic model for our needs. We can use these pre-trained models without having to train a model from scratch on a large dataset.

MobileNet V2 model was developed at Google [36], pre-trained on the ImageNet dataset with 1.4M images and 1000 classes of web images. Since ImageNet is a large dataset in terms of number of images and classes, models trained with this dataset and giving good results can be easily adapted for other datasets.

We created an artificial neural network using transfer learning feature extraction as in Figure 6. We chose 32 as the batch size in the artificial neural network. In this artificial neural network, we first subjected RGB images to data augmentation, then we extracted features with MobileNet V2 feature extractor, transformed the extracted features into vector by the global average pooling layer and passing them through various dense, batch normalization, dropout layers. All MobileNet V2 feature extractor weights are frozen for 5 epochs in order to heat the weights of other layers. Learning rate was 0.001 during initial 5 epochs. Then we applied the first fine-tuning process for 45 epochs by unfreezing the last 4 layers of MobileNet V2, reducing the learning rate to 0.0001. Later we applied the second fine-tuning process for 50 epochs by unfreezing the last 6 layers of MobileNet V2 and reducing the learning rate to 0.0001. So, we used 100 epochs in total for training.

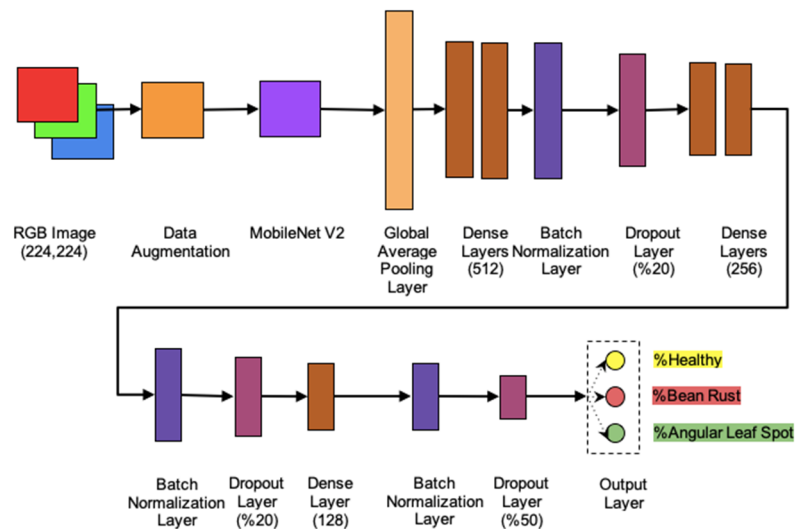


Figure 6. Structure of artificial neural network with transfer learning feature extraction.

2.7. Artificial neural network with feature fusion

Feature fusion is the process of combining features extracted each of based on a different feature extraction method. Combining features can improve the accuracy of a classification model. We performed feature fusion by concatenating the descriptive feature vectors obtained from the artificial neural network which is using transfer learning feature extraction and the artificial neural network which is using HOG feature extraction.

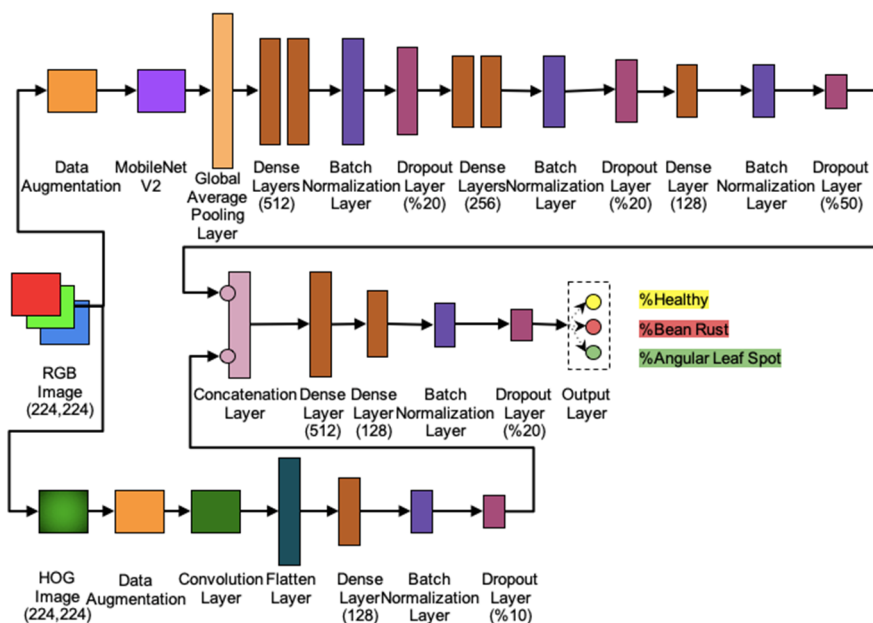


Figure 7. Structure of artificial neural network with feature fusion.

The artificial neural network using feature fusion is shown in Figure 7. This network consists of

two separate branches through which HOG and RGB images pass. After the descriptive feature vectors which contains 128 neurons obtained from each of these branches are combined with a concatenation layer, they reach the output layer by passing through the dense, batch normalization layer, and dropout layers. By using the softmax activation function in the output layer, an estimate is made about which class the input image belongs to. The images were given to the network in batches of 32 images during the training. After training with a learning rate of 0.001 for 5 epochs, we unfreeze the last 4 layers of the MobileNet V2 feature extractor on the transfer learning feature extraction branch to apply the first fine tuning. We changed the learning rate to 0.0001 and trained the model for 45 epochs. Then, for the second fine tuning, we unfreeze the last 6 layers of the MobileNet V2 feature extractor, changed the learning rate to 0.00001 and trained the model for 50 more epochs.

3. Results

Figures 8 and 9 show the results of accuracy and loss results obtained from training and validation datasets of the artificial neural network using only HOG feature extraction. The accuracy and loss results in the training and validation datasets are not far from each other. Accuracy results remained approximately the same from the start of training and showed minor oscillations. Although the loss values followed a similar structure, especially after the 85th epoch, the loss values of the training dataset remained constant, while the loss value of the validation dataset made some peaks.

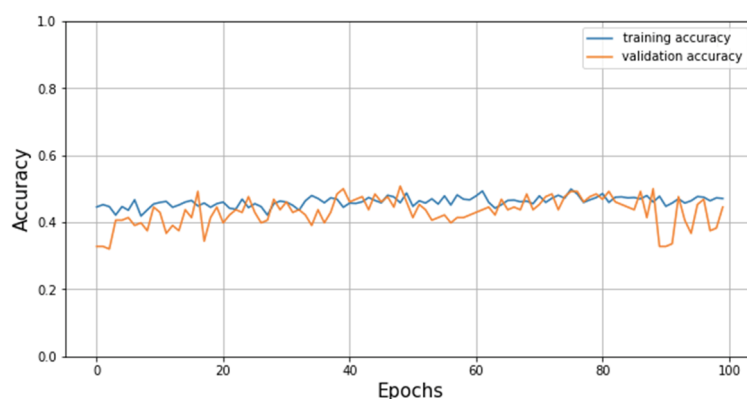


Figure 8. Accuracy results of artificial neural network with HOG feature extraction.

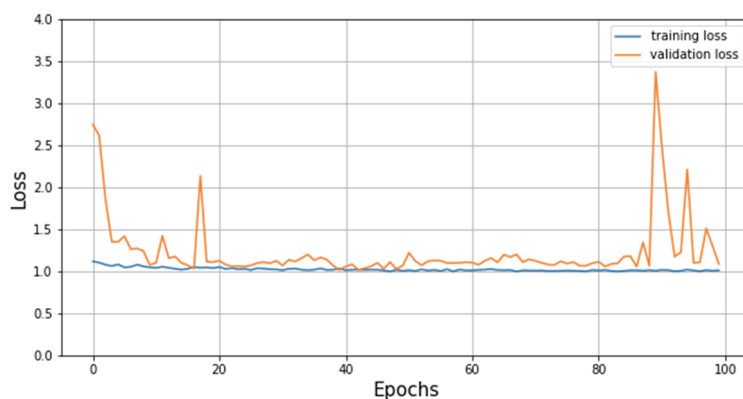


Figure 9. Loss results of artificial neural network with HOG feature extraction.

Figures 10 and 11 show the graphs of the accuracy and loss results obtained from the training and validation datasets of the artificial neural network using only transfer learning feature extraction. The training and validation accuracy values, which converged during the first 35 epochs, then began to diverge from each other. In addition, after the 80th epoch, both the validation and training set accuracy values remained approximately constant, sitting on a plateau around 92 and 96%, respectively.

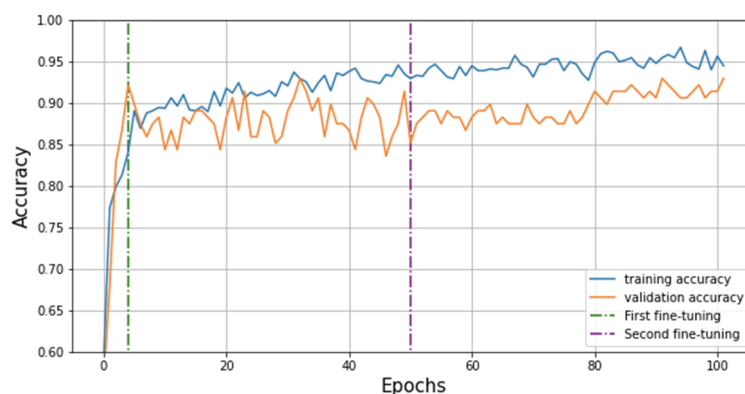


Figure 10. Accuracy results of artificial neural network with transfer learning feature extraction.

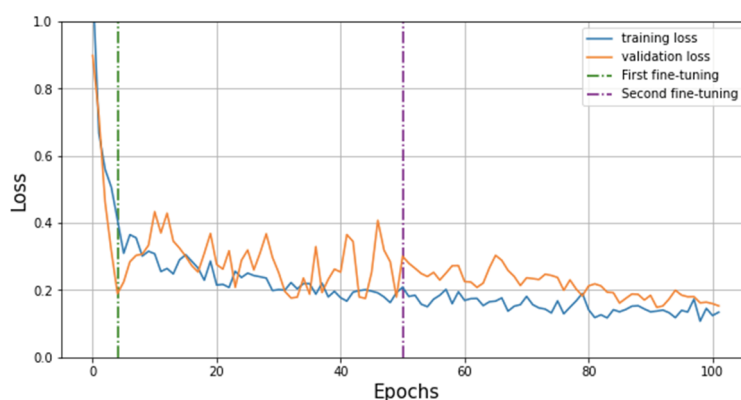


Figure 11. Loss results of artificial neural network with transfer learning feature extraction.

Figures 12 and 13 show the graphs of the accuracy and loss results obtained from the training and validation datasets of the artificial neural network using feature fusion, in which HOG and transfer learning feature extraction methods are used simultaneously. When we examine the accuracy and loss results of the feature fusion model, it is seen that the training and validation accuracy and loss values converge after the 20th epoch, and that the oscillation of the accuracy and loss values decreases after the 50th epoch.

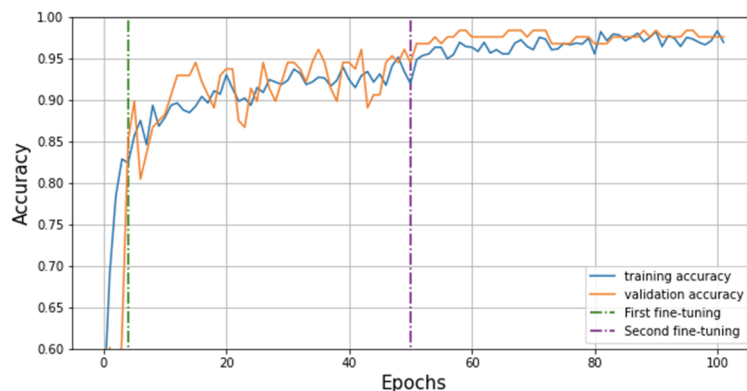


Figure 12. Accuracy results of neural network with feature fusion.

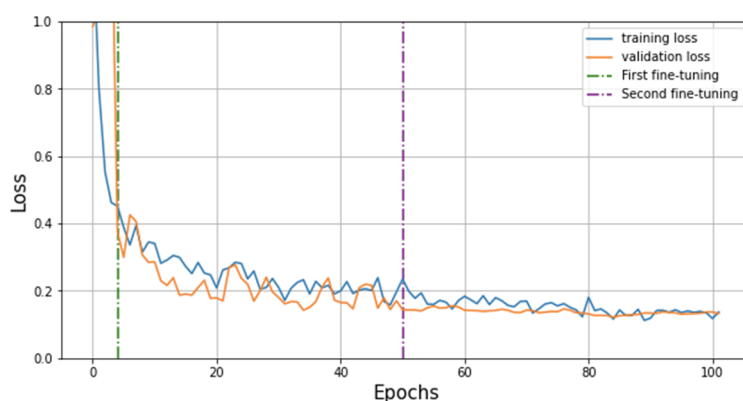


Figure 13. Loss results graphic of artificial neural network with feature fusion.

We gave the accuracy values of all three models on the training, validation and test datasets in Table 3. When the model using feature fusion is compared to the model using transfer learning feature extraction method, it has been seen that the accuracy value is 1.65, 5.53 and 1.53% higher in the training, validation and test datasets, respectively. These results show us that the feature fusion model increases classification accuracy compared to the other two models. When we compared the accuracy values in the training set of the feature fusion model with the accuracy values in the validation and test datasets, we found a difference of -0.07 and -0.9%, respectively.

Table 3. Accuracy results of three models on training, validation and test datasets.

	HOG feature extraction model	Transfer learning feature extraction model	Feature fusion model
Training accuracy	46.51%	96.71%	98.33%
Validation accuracy	50.78%	92.96%	98.40%
Test accuracy	44.36%	97.74%	99.24%

Confusion matrix of the HOG feature extraction model on the test dataset is shown in Table 4. It

is seen that this model predicts the majority of the samples given in the test dataset as healthy class. Confusion matrix of the transfer learning feature extraction model on the test dataset is given in Table 5. In this model, the most error was estimating the images of the healthy class as belonging to the bean rust class. Confusion matrix of the feature fusion model on the test dataset is given in Table 6.

Table 4. Confusion matrix of HOG feature extraction model.

	Predicted Healthy	Predicted Bean Rust	Predicted Angular Leaf Spot
Healthy	34	1	9
Bean Rust	36	2	7
Angular Leaf Spot	21	0	23

Table 5. Confusion matrix of transfer learning feature extraction model.

	Predicted Healthy	Predicted Bean Rust	Predicted Angular Leaf Spot
Healthy	41	3	0
Bean Rust	0	45	0
Angular Leaf Spot	0	0	44

Table 6. Confusion matrix of feature fusion model.

	Predicted Healthy	Predicted Bean Rust	Predicted Angular Leaf Spot
Healthy	44	0	0
Bean Rust	0	44	1
Angular Leaf Spot	0	0	44

Table 7. Classification report of models.

	Precision			Recall			F1-Score		
Healthy	1.00 ^a	1.00 ^b	0.37 ^c	1.00 ^a	0.93 ^b	0.77 ^c	1.00 ^a	0.96 ^b	0.50 ^c
Bean Rust	1.00 ^a	0.94 ^b	0.67 ^c	0.98 ^a	1.00 ^b	0.04 ^c	0.99 ^a	0.97 ^b	0.08 ^c
Angular Leaf Spot	0.98 ^a	1.00 ^b	0.59 ^c	1.00 ^a	1.00 ^b	0.52 ^c	0.99 ^a	1.00 ^b	0.55 ^c

*note: ^a: Feature fusion model, ^b: Transfer learning feature extraction model, ^c: HOG feature extraction model

In Table 4, we examined the confusion matrix obtained using the test dataset with the HOG feature extraction model. 77.27% of the leaves belonging to the healthy class were classified as healthy, 2.27% as bean rust, and 20.45% as angular leaf spot. 80% of the leaves belonging to the bean rust class were classified as healthy, 4.44% as bean rust, and 15.56% as angular leaf spot. 47.72% of the leaves belonging to the angular leaf spot class were classified as healthy and 52.28% classified correctly.

In Table 5, we examined the confusion matrix obtained using the test dataset with the Transfer learning feature extraction model. 93.18% of the leaves belonging to the healthy class were classified as healthy and 6.82% as bean rust. All of the leaves belonging to the bean rust class and angular leaf spot classes were classified correctly.

In Table 6, we examined the confusion matrix obtained by using the test dataset with the feature fusion model. All of the leaves belonging to the healthy and angular leaf spot classes were classified correctly. On the other hand, 97.78% of the leaves belonging to the bean rust class were classified as

bean rust and 2.2% as angular leaf spot.

In Table 7, we examined the classification report of three different models on the test dataset. Feature fusion model has best values on precision, recall and F1-scores.

4. Discussion

In the HOG feature extraction model, HOG images have compressed information. Because of that the HOG feature extraction model was kept shallow to prevent overfitting. Also HOG feature extraction model has been trained from scratch with a shallower structure, so the loss and accuracy values have remained low compared to other methods (Figures 8 and 9). Fluctuation in validation losses can be about the small validation dataset. However, the accuracy and loss values obtained in the training and validation sets are not far from each other. In this sense, we can say that HOG images are successful in preventing overfitting.

In the transfer learning feature extraction model, the training and validation accuracy values, which converged during the first 35 epochs, then started to diverge from each other. This indicates overfitting. In addition, after the 80th epoch, both validation and training dataset accuracy values remained approximately constant, reaching a plateau in the 92 and 96% bands, respectively (Figure 10).

When we examine the accuracy and loss results of the feature fusion model, in which HOG and transfer learning feature extraction methods are used simultaneously, it is seen that they training and validation results converge after the 20th epoch, and that the oscillation of accuracy and loss values decreases after the 50th epoch (Figures 12 and 13). The results indicate that this artificial neural network with feature fusion shortens the converge time, increases the accuracy, and a more generalizable model can be obtained [24,25].

The results of the accuracy performances of all three models on training, validation and test datasets are given in Table 3. When the feature fusion model is compared with the transfer learning feature extraction model, it is seen that it provides 1.65, 5.53 and 1.53% performance increase in accuracy on training, validation and test datasets, respectively. These results show us that the feature fusion model increases classification accuracy compared to the other two models.

In Table 4, we examined the confusion matrix obtained using the test dataset with the HOG feature extraction model. 77.27% of the leaves belonging to the healthy class were classified as healthy, 2.27% as bean rust, and 20.45% as angular leaf spot. 80% of the leaves belonging to the bean rust class were classified as healthy, 4.44% as bean rust, and 15.56% as angular leaf spot. 47.72% of the leaves belonging to the angular leaf spot class were classified as healthy and 52.28% classified correctly. These results show that the HOG feature extraction model is not very successful in classification and especially fails to distinguish diseased leaves from healthy ones. The shallow nature of the HOG feature extraction model and the use of only HOG images that already have compressed information are the most important reasons for this underfitting.

In Table 5, we examined the confusion matrix obtained using the test dataset with the Transfer learning feature extraction model. 93.18% of the leaves belonging to the Healthy class were classified as healthy and 6.82% as bean rust. All of the leaves belonging to the Bean rust class and Angular leaf spot classes were classified correctly. These results show that while the diseased leaves are correctly detected with the transfer learning feature extraction model, some healthy leaves are misclassified as bean rust.

In Table 6, we examined the confusion matrix obtained by using the test dataset with the feature

fusion model. All of the leaves belonging to the healthy and angular leaf spot classes were classified correctly. On the other hand, 97.78% of the leaves belonging to the bean rust class were classified as bean rust and 2.2% as angular leaf spot. These results show that while healthy and angular leaf spot leaves were detected correctly with the feature fusion model, some healthy leaves were misclassified as angular leaf spot.

There are two main reasons for the misclassifications in Tables 5 and 6. The first and most important is that the dataset we use is not large enough and the second there are background scenes in most of the images. The fact that the background scenes have similar color and texture to the diseases may be the reason for the error [26,28].

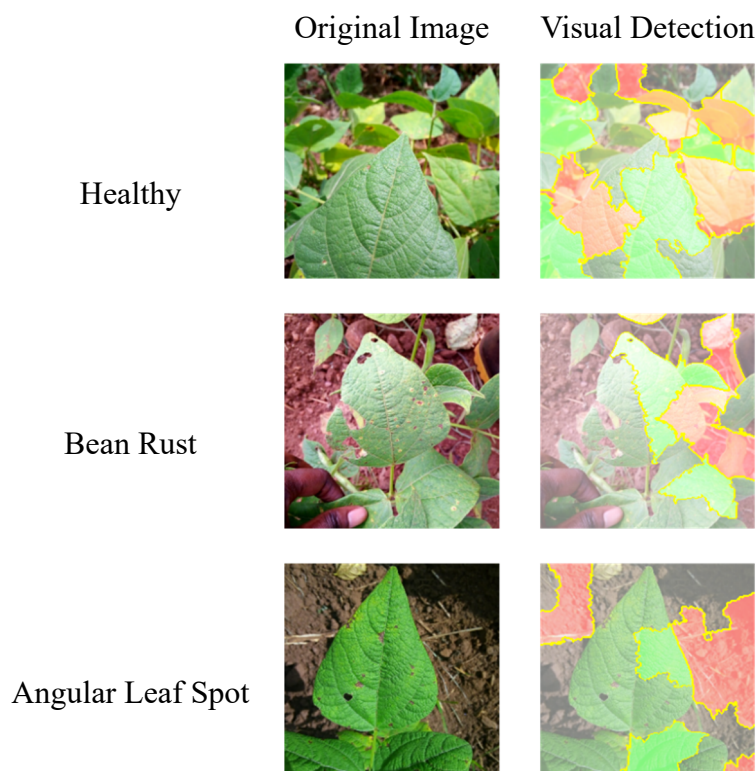


Figure 14. Visual detection results.

Figure 14 shows some images and visual detection results in the test dataset by using feature fusion model. Lime v0.2.0.1 library was used to visualize the results [37]. As seen in visual detection column, the most important parts of the image used by the model to make a decision for classification, are outlined in green, and the unimportant parts for the model are outlined in red.

It is seen that the proposed feature fusion model has shown state of the art accuracy when compared with other studies using ibean dataset and transfer learning method [25,38].

In recent years, image classification by transformer-based methods has been reported to give more successful results in terms of accuracy than CNN-based methods [39,40]. However, training transformer models from scratch requires much more data than CNN models, but agricultural production has a periodic structure, and it is often not possible to collect more images of the problem in question [41]. Also, transformer methods usually result in larger model size. Therefore, the operation of models using the transformer method may be slower than CNN models in computer

vision [42]. If the models developed to detect leaf diseases are used in smart spraying machines (only spraying the areas where the disease is present) machines, they should be able to work in mobile and edge devices with low computational resources [43]. In this respect, it is important that the models developed are not only accurate but also suitable for working on mobile and edge devices. Yu et al. [44] developed the Inception Convolutional Vision Transformer (ICVT) model by mixing CNN and transformer architecture and tested it on the ibean dataset. The number of parameters, which was 49M in the ViT/16 model [45], has decreased to 25M. The model we developed that is based on CNN feature fusion achieved slightly better accuracy compared to the ICVT model with 34% less (16M) parameters.

Furthermore, inductive bias plays an important role in the ability of machine learning models to generalize to unseen data, and CNN models have a higher inductive bias than transformer models [43].

5. Conclusions

In this study, we made feature fusion of descriptive vectors obtained from two different machine learning models, which are using HOG feature extraction and transfer learning feature extraction, to classify healthy, bean rust and angular leaf spot classes on bean leaf images. The model developed in the study is unique in terms of its network structure, pre-processing processes used, input types and the use of this method for disease detection in bean leaves. This model is composed of two different branches that pass-through HOG and RGB images. After combining the descriptive characteristic vectors that contain 128 neurons obtained from each of these branches with concatenation layer, they reach the output layer through the dense, batch normalization and dropout layers. The softmax activation function in the output layer is used to estimate the class of input image. The model using feature fusion has higher accuracy on training, validation and test datasets than models using only HOG feature extraction and only Transfer Learning feature extraction. In addition, this new method provides a faster converge to the solution during the training of the artificial neural network. The present study has only examined bean leaf dataset Therefore, we will apply this model on various datasets to obtain better generalizability. Also, we will focus on model size optimization for future mobile applications.

Conflicts of interest

The authors declare that there are no conflicts of interest.

References

1. G. S. Malhi, M. Kaur, P. Kaushik, Impact of climate change on agriculture and its mitigation strategies: A review, *Sustainability*, **13** (2021), 1318. <https://doi.org/10.3390/su13031318>
2. K. Yin, J. L. Qiu, Genome editing for plant disease resistance: applications and perspectives, *Phil. Trans. R. Soc. B*, **374** (2019), 20180322. <https://doi.org/10.1098/rstb.2018.0322>
3. Z. Hu, What socio-economic and political factors lead to global pesticide dependence? A critical review from a social science perspective, *Int. J. Environ. Res. Public Health*, **17** (2020), 8119. <https://doi.org/10.3390/ijerph17218119>

4. S. Roy, J. Halder, N. Singh, A. B. Rai, R. N. Prasad, B. Singh, Do vegetable growers really follow the scientific plant protection measures? An empirical study from eastern Uttar Pradesh and Bihar, *Ind. J. Agric. Sci.*, **87** (2017), 1668–1672.
5. M. Ş. Şengül Demirak, E. Canpolat, Plant-based bioinsecticides for mosquito control: impact on insecticide resistance and disease transmission, *Insects*, **13** (2022), 162. <https://doi.org/10.3390/insects13020162>
6. W. Cramer, J. Guiot, M. Fader, J. Garrabou, J. P. Gattuso, A. Iglesias, et al., Climate change and interconnected risks to sustainable development in the Mediterranean, *Nat. Clim. Change*, **8** (2018), 972–980. <https://doi.org/10.1038/s41558-018-0299-2>
7. H. N. Fones, D. P. Bebber, T. M. Chaloner, W. T. Kay, G. Steinberg, S. J. Gurr, Threats to global food security from emerging fungal and oomycete crop pathogens, *Nat. Food*, **1** (2020), 332–342. <https://doi.org/10.1038/s43016-020-0075-0>
8. M. Tudi, H. D. Ruan, L. Wang, J. Lyu, R. Sadler, D. Connell, et al., Agriculture development, pesticide application and its impact on the environment, *Int. J. Environ. Res. Public Health*, **18** (2021), 1112. <https://doi.org/10.3390/ijerph18031112>
9. A. S. Tulshan, N. Raul, Plant leaf disease detection using machine learning, in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019. <https://doi.org/10.1109/ICCCNT45670.2019.8944556>
10. A. Kumar, J. P. Singh, A. K. Singh, Randomized convolutional neural network architecture for eyewitness tweet identification during disaster, *J. Grid Comput.*, **20** (2022). <https://doi.org/10.1007/s10723-022-09609-y>
11. L. Xu, J. Xie, F. Cai, J. Wu, Spectral classification based on deep learning algorithms, *Electronics*, **10** (2021), 1892. <https://doi.org/10.3390/electronics10161892>
12. Ü. Atila, M. Uçar, K. Akyol, E. Uçar, Plant leaf disease classification using Efficient Net deep learning model, *Ecol. Inf.*, **61** (2021), 101182. <https://doi.org/10.1016/j.ecoinf.2020.101182>
13. S. Zhang, S. Zhang, C. Zhang, X. Wang, Y. Shi, Cucumber leaf disease identification with global pooling dilated convolutional neural network, *Comput. Electron. Agric.*, **162** (2019), 422–430. <https://doi.org/10.1016/j.compag.2019.03.012>
14. D. Jakobovitz, R. Giryes, M. R. Rodrigues, Generalization error in deep learning, in *Compressed Sensing and Its Applications: Third International MATHEON Conference 2017*, Birkhäuser, Cham, (2019), 153–193. <https://doi.org/10.48550/arXiv.1808.01174>
15. A. Al-Saffar, A. Bialkowski, M. Baktashmotlagh, A. Trakic, L. Guo, A. Abbosh, Closing the gap of simulation to reality in electromagnetic imaging of brain strokes via deep neural networks, *IEEE Trans. Comput. Imaging*, **7** (2020), 13–21. <https://doi.org/10.1109/tci.2020.3041092>
16. G. Algan, I. Ulusoy, Image classification with deep learning in the presence of noisy labels: A survey, *Knowl.-Based Syst.*, **215** (2021), 106771. <https://doi.org/10.1016/j.knosys.2021.106771>
17. C. Wu, S. Guo, Y. Hong, B. Xiao, Y. Wu, Q. Zhang, Discrimination and conversion prediction of mild cognitive impairment using convolutional neural networks, *Quant. Imaging Med. Surg.*, **8** (2018), 992. <https://doi.org/10.21037/qims.2018.10.17>
18. K. Aderghal, A. Khvostikov, A. Krylov, J. Benois-Pineau, K. Afdel, G. Catheline, Classification of Alzheimer disease on imaging modalities with deep CNNs using cross-modal transfer learning, in *2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)*, IEEE, (2018), 345–350. <https://doi.org/10.1109/cbms.2018.00067>

19. D. Chen, Y. Lu, Z. Li, S. Young, Performance evaluation of deep transfer learning on multi-class identification of common weed species in cotton production systems, *Comput. Electron. Agric.*, **198** (2022), 107091. <https://doi.org/10.1016/j.compag.2022.107091>
20. M. Ahsan, M. A. Based, J. Haider, M. Kowalski, COVID-19 detection from chest X-ray images using feature fusion and deep learning, *Sensors*, **21** (2021), 1480. <https://doi.org/10.3390/s21041480>
21. L. Wei, K. Wang, Q. Lu, Y. Liang, H. Li, Z. Wang, et al., Crops fine classification in airborne hyperspectral imagery based on multi-feature fusion and deep learning, *Remote Sens.*, **13** (2021), 2917. <https://doi.org/10.3390/rs13152917>
22. C. Shang, F. Wu, M. Wang, Q. Gao, Cattle behavior recognition based on feature fusion under a dual attention mechanism, *J. Visual Commun. Image Represent.*, **85** (2022), 103524. <https://doi.org/10.1016/j.jvcir.2022.103524>
23. H. C. Chen, A. M. Widodo, A. Wisnujati, M. Rahaman, J. C. W. Lin, L. Chen, et al., AlexNet convolutional neural network for disease detection and classification of tomato leaf, *Electronics*, **11** (2022), 951. <https://doi.org/10.3390/electronics11060951>
24. X. Fan, P. Luo, Y. Mu, R. Zhou, T. Tjahjadi, Y. Ren, Leaf image based plant disease identification using transfer learning and feature fusion, *Comput. Electron. Agric.*, **196** (2022), 106892. <https://doi.org/10.1016/j.compag.2022.106892>
25. E. Elfatimi, R. Eryigit, L. Elfatimi, Beans leaf diseases classification using mobilenet models, *IEEE Access*, **10** (2022), 9471–9482. <https://doi.org/10.1109/ACCESS.2022.3142817>
26. S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, R. Pramodhini, Plant leaf disease detection using computer vision and machine learning algorithms, *Global Transitions Proc.*, **3** (2022), 305–310. <https://doi.org/10.1016/j.gltp.2022.03.016>
27. J. Annrose, N. Rufus, C. R. Rex, D. G. Immanuel, A cloud-based platform for soybean plant disease classification using archimedes optimization based hybrid deep learning model, *Wireless Pers. Commun.*, **122** (2022), 2995–3017. <https://doi.org/10.1007/s11277-021-09038-2>
28. A. K. Singh, S. V. N. Sreenivasu, U. S. B. K. Mahalaxmi, H. Sharma, D. D. Patil, E. Asenso, Hybrid feature-based disease detection in plant leaf using convolutional neural network, Bayesian optimized SVM and random forest classifier, *J. Food Qual.*, **2022** (2022). <https://doi.org/10.1155/2022/2845320>
29. *Makerere AI Lab*, Bean disease dataset, 2020. Available from: <https://github.com/AI-Lab-Makerere/ibean>.
30. A. Mikołajczyk, M. Grochowski, Data augmentation for improving deep learning in image classification problem, in *2018 International interdisciplinary PhD workshop (IIPhDW)*, IEEE, (2018), 117–122. <https://doi.org/10.1109/iiphdw.2018.8388338>
31. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, **1** (2022), 886–893. <https://doi.org/10.1109/cvpr.2005.177>
32. S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, et al., scikit-image: Image processing in Python, *PeerJ*, 2014. <https://doi.org/10.7287/peerj.preprints.336v2>
33. W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, K. R. Müller, Explaining deep neural networks and beyond: A review of methods and applications, *Proc. IEEE*, **109** (2021), 247–278. <https://doi.org/10.1109/jproc.2021.3060483>

34. Tensorflow Keras: Layers, Retrieved October 6, 2022. Available from: https://www.tensorflow.org/api_docs/python/tf/keras/layers.
35. D. P. Kingma, J. A. Ba, J. Adam, A method for stochastic optimization, preprint, arXiv:1412.6980. <https://doi.org/10.48550/arXiv.1412.6980>
36. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 4510–4520. <https://doi.org/10.1109/cvpr.2018.00474>
37. M. T. Riberio, S. Singh, C. Guestrin, “Why should i trust you?” Explaining the predictions of any classifier, in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, (2016), 1135–1144. <https://doi.org/10.1145/2939672.2939778>
38. P. Bedi, P. Gole, PlantGhostNet: An efficient novel convolutional neural network model to identify plant diseases automatically, in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, IEEE, (2021), 1–6. <https://doi.org/10.1109/ICRITO51393.2021.9596543>
39. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, Z. Xiaohua, T. Unterthiner, et al., An image is worth 16x16 words: Transformers for image recognition at scale, preprint, arXiv:2010.11929. <https://doi.org/10.48550/arXiv.2010.11929>
40. Y. Borhani, J. Khoramdel, E. Najafi, A deep learning based approach for automated plant disease classification using vision transformer, *Sci. Rep.*, **12** (2022), 1–10. <https://doi.org/10.1038/s41598-022-15163-0>
41. Y. Lu, S. Young, A survey of public datasets for computer vision tasks in precision agriculture, *Comput. Electron. Agric.*, **178**, (2020), 105760. <https://doi.org/10.1016/j.compag.2020.105760>
42. X. Zhai, A. Kolesnikov, N. Houlsby, L. Beyer, Scaling vision transformers, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2022), 12104–12113.
43. J. M. P. Czarnecki, S. Samiappan, M. Zhou, C. D. McCraigne, L. L. Wasson, Real-time automated classification of sky conditions using deep learning and edge computing, *Remote Sens.*, **13** (2021), 3859. <https://doi.org/10.3390/rs13193859>
44. S. Yu, L. Xie, Q. Huang, Inception convolutional vision transformers for plant disease identification, *Internet Things*, **21** (2023), 100650. <https://doi.org/10.1016/j.iot.2022.100650>
45. H. Xu, X. Su, D. Wang, CNN-based local vision transformer for covid-19 diagnosis, preprint, arXiv:2207.02027. <https://doi.org/10.48550/arXiv.2207.02027>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>).