IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
IJARSCT
Impact Factor: 6.252
Volume 2, Issue 4, May 2022

Leaves Disease Detection using Deep Learning

Akshay Tike[1], Jambuvant Kadam[2], Dhairysheel Jadhav[3], Rushikesh Nalbalwar[4], A. P. Kale[5]
Students, Department of Computer Engineering[1,2,3,4]
Professor, Department of Computer Engineering[5]
MES College of Engineering, Pune, Maharashtra, India

Abstract: *Plant diseases are important factors in determining plant yield and quality. Plant disease identification can be accomplished through digital image processing. Deep learning has made significant advances in digital image processing in recent years, far outperforming traditional methods. One of the primary factors determining crop yield loss in crop production and agriculture is the identification and detection of plant diseases. Plant disease research is the study of any visible points in any part of the plant that aids in the differentiation of two plants, technically any spots or colour shades. It is extremely difficult to correctly identify plant diseases. Identification of the disease necessitates a lot of work and expertise, as well as a lot of knowledge in the field of plants and disease detection studies. As a result, image processing is used to detect plant diseases. Disease detection employs image acquisition, image extraction, image segmentation, and image pre-processing techniques.*

Keywords: CNN, Pooling Layer, ReLU, ConvNet

I. Introduction

Identifying plant diseases is the one of the important factors to avoiding loss of yield and quantity of produce. Studying plant diseases means studying the visually observable patterns found in plants. Monitoring plant health and detecting disease are important for sustainable agriculture. Manual monitoring of plant diseases is very difficult. It requires a huge amount of work, plant disease expertise and excessive processing time. Therefore, we use image processing and machine learning techniques to detect plant diseases. Disease detection includes steps such as image acquisition, image pre-processing, image segmentation, feature extraction, and classification. Early detection is the basis for effective prevention and control of plant diseases, which play an important role in the control and decision-making of agricultural production. In recent years, identifying plant diseases has become an important issue. The outbreak of plant diseases adversely affects agricultural production. Food anxiety increases when plant diseases are not detected in time. In most cases, farmers are aware of crop diseases and pests based on their experience. Not only is this method subjective, it is time consuming, tedious and inefficient. Therefore, research on the use of image processing techniques to detect plant diseases can be of great help in addressing these challenges. Here are the steps you need to take to do that: Image acquisition, image pre-processing, feature extraction, and the final step are classification.

1.1 Image Acquisition

The first process is to collect data from the public repository. It takes an image as input for further processing. Since we used the most common image domain, you can use any format as input to the process. .bmp, .jpg, .gif. Images are captured, scanned and converted to manageable units. This process is called image acquisition. In the test phase, a digital scanner is used to capture a series of colour images and a single sheet image. Colour images are digitized to produce RGB digital colour images. Image pre-treatment. The image was taken from a field and may contain dust, spores, and lime as noise. The purpose of data pre-processing is to denoise the image and adjust the pixel values. Image quality is improved. The main purpose of pre-processing is to suppress unwanted image data and enhance some important image functions. Includes RGB to grey conversion, image resizing, and median filtering. Here, the colour image is converted to a grayscale image and the imaging device is independent. The image is then scaled to a size of 256 * 256. The image is then median filtered to remove noise. The digital version of the rotten leaf sample has about 30% of the leaf area of and the remaining 70% of the background. There may be dust, dew droplets, and bug droplets on the plant, so image pre-treatment is required to get better results in the next step. These are counted as noise. Also, the captured image can be distorted by the effects of water droplets and shadows, which can cause segmentation and feature extraction problems. The effects of such distortion can be mitigated

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 4, May 2022

Impact Factor: 6.252

or eliminated using various denoising filters. The contrast of the captured image may be low. You can use a contrast enhancement algorithm for such images. If you need to extract areas of interest, you may also need background removal techniques. The median filter can be used for noise such as salt and pepper. For images taken with a high-resolution camera, the image size can be very large, so you need to reduce the image size. Image reduction also helps reduce computational memory performance.

1.2 Classification

Convolutional neural networks are used in the automatic detection of leaves diseases. CNN is chosen as a classification tool due to its well-known technique as a successful classifier for many real applications. CNN architectures vary with the type of the problem at hand. The proposed model consists of three convolutional layers each followed by a max-pooling layer. The final layer is fully connected MLP. ReLu activation function is applied to the output of every convolutional layer and fully connected layer. The first convolutional layer filters the input image with 32 kernels of size 3x3. After max-pooling is applied, the output is given as an input for the second convolutional layer with 64 kernels of size 4x4. The last convolutional layer has 128 kernels of size 1x1 followed by a fully connected layer of 512 neurons. The output of this layer is given to SoftMax function which produces a probability distribution of the four output classes.

II. Literature Survey

Plant Disease Identification Using a Novel Convolutional Neural Network - In this paper they have proposed a novel CNN model based on the inception and residual connection that can effectively classify the diseases in plants. In addition, they have used depth wise separable convolution in the inception architecture which reduces the computation cost by reducing the number of parameters by a margin of 70%. Therefore, training the network requires much less time as compared to the standard CNN. The experimental result shows that the proposed model achieves higher performance accuracy. To evaluate the robustness of the model, we have used three different plant datasets [1].

Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection- They proposed a multi-class mango leaf disease classification using deep neural networks. At first, they used a wrapper-based feature selection approach using an Adaptive Particle-Grey Wolf metaheuristic (APGWO) which was performed to select 81 features out of the originally proposed 114 features. These features are selected as inputs for the MLP for the classification task. Their approach developed outperformed deep learning models such as VGG, AlexNet, ResNet-50, which are already enhanced with transfer learning (89.41% vs 78.64%, 79.92%, and 84.88%, respectively).
For the ANN with FS approach, the model attains 91.32% training accuracy and 85.45% testing accuracy for recognizing type of infected blobs [2].

Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture- They proposed a deep-learning-based method for texture classification with performance compatible and real-time processing scenarios. They used CNN as feature descriptor rather than end-to-end classifiers and combined them with SVMs. And for obtaining a relevant classification performance even for small datasets, they based their work on the transfer learning concept and adapted to the task popular CNN models (AlexNet, Vgg16, ResNet) pre-trained on the very large ImageNet object-based dataset. In their experimental section, they considered two datasets: a public one with generic RGB textures (for initial validation of the proposed approach) and a dataset from the applied field of precision agriculture consisting of images with leaves from several plant species and affected by several diseases. They analysed the classification results obtained by extracting features from several different layers of the different CNN pre-trained models and using them for describing the textures in the proposed datasets. Experimental results from the extraction of features from early convolutional layers is relevant for texture classification as the generated characteristics are more general [3].

III. Convolutional Neural Network

A convolutional neural network, abbreviated as CNN, has a complex network structure and can perform convolutional operations. A convolutional neural network model consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. In the model, the convolutional layer and the pooling layer alternate several times, and if the convolutional layer neurons are connected to the pooling layer neurons, a complete connection is not required. CNN is a popular deep learning model. The reason is the vast amount of model capacity and complex information provided by

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
IJARSCT
Impact Factor: 6.252
Volume 2, Issue 4, May 2022

the basic structural characteristics of the CNN. This gives CNN an advantage in image recognition. At the same time, CNN's achievements in computer vision tasks have spurred the growing popularity of deep learning.

3.1 Modelling One Neuron

A neuron is the basic computational unit of the human brain. The human nervous system has about 86 billion neurons, which are connected to about 1014-1015 synapses. The figure below shows a cartoon drawing of a biological neuron (left) and a general mathematical model (right). Each neuron receives an input signal from the dendrite and produces an output signal along its (single) axon. Axons eventually branch and connect to the dendrites of other neurons via synapses. In the computational model of one neuron, signals traveling along an axon (such as x0) interact multiplicatively with the dendrites of other neurons based on the synaptic intensity at that synapse (such as w0). (W0x0, etc.). Synaptic strength (weight w) is learnable and controls the strength of the effect of one neuron on another (and its direction: excitatory (positive weight) or inhibitory (negative weight)). Is the idea. In the basic model, the dendrites carry the signal to the cell body, where everything is summed. When the final sum exceeds a certain threshold, the neuron fires, and spikes the axons.

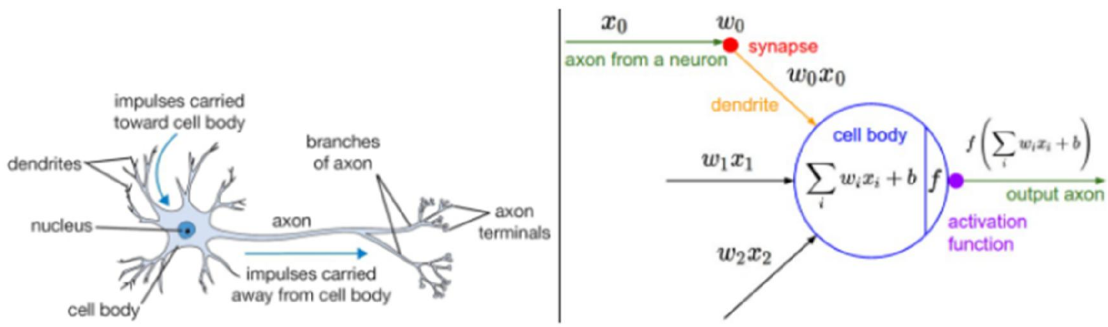

Figure 1: Analogy between brain and mathematical neuron

The computational model assumes that the exact timing of the peaks is irrelevant and that only the frequency of firing provides information. Based on this rate code interpretation, we model the firing rate of neurons using the activation function f, which represents the frequency of spikes along axons. Historically, the sigmoid function is a common choice for activation functions because it takes a real-valued input (the signal strength after summing) and narrows it down to the range 0 to 1.

In a nutshell, A ConvNet usually has 3 types of layers:

- Convolutional Layer
- Pooling Layer
- Fully Connected Layer

3.2 Convolutional Layer

The convolution layer is the core building block of the CNN. It carries the main portion of the network's computational load. The main purpose of the convolution step is to extract features from the input image.

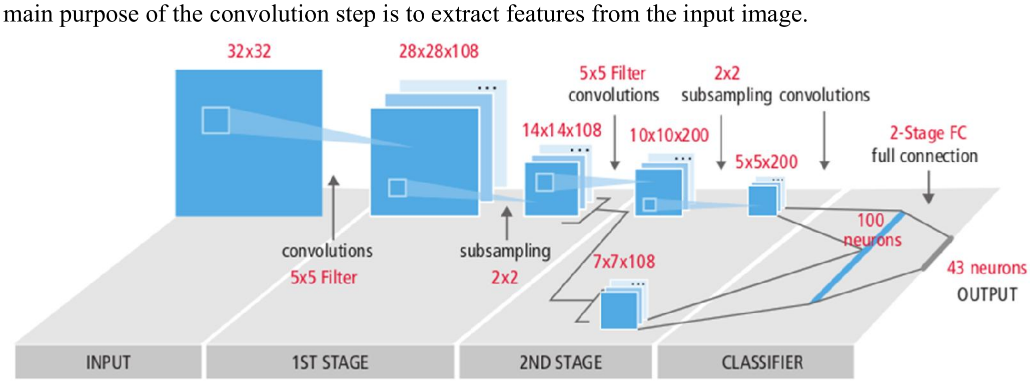

Figure 2: Typical block diagram of a CNN

The convolution layer is always the first step in CNN. There are input images, feature detectors, and feature maps. This layer performs the dot product between the two matrices. One matrix is a set of learnable parameters, also known as the kernel, and the other matrix is the constrained part of the receptive field. The kernel is spatially smaller than the image, but more detailed. That is, if the image consists of three (RGB) channels, the core height and width will be spatially smaller, but the depth will extend to all three channels. The convolution operation extracts various characteristics of the input. The first level of convolution extracts low-level features such as edges, lines, and corners. High-level layers extract high-level features. The input size is NxN x D, which is collapsed in the H kernel, and each size is k x kxD. Convolving inputs in one kernel produces one output function, and the H kernel independently produces H functions. Starting from the upper left corner of the input, each kernel is shifted element by element from left to right. When you reach the upper right corner, the core moves one element down, and the core again moves one element from left to right at a time. This process repeats until the kernel reaches the lower right corner. For N = 32 and k = 5, there are 28 unique positions from left to right that the kernel can occupy, and 28 unique positions from top to bottom. Corresponding to these positions, each feature of the output contains 28 x 28 (that is, (Nk + 1) x (Nk + 1)) elements. For each position of the kernel in a sliding window process, k x k x D elements of input and k x k x D elements of kernel are element by element multiplied and accumulated. So, to create one element of one output feature, k x k x D multiply accumulate operations are required.

The ReLU layer (normalized linear unit) is another step towards the convolution layer. Apply an activation function to the feature map to increase the non-linearity of the network. This is because the image itself is very non-linear. Setting the activation map to zero removes negative values from the activation map. Convolution is a linear operation that involves element-by-element matrix multiplication and addition. The actual data you want the CNN to train will be non-linear.

3.3 Pooling Layer

The figure below explains the pooling process more. The input size is of 4x4. For 2x2 subsampling, a 4x4 image is divided into four non-overlapping matrices of size 2x2. In the case of max pooling, the maximum value of the four values in the 2x2 matrix is the output. If there is average pooling case, the average of the four values is the output. And for the output with index (2,2), the result of averaging is a fraction that has been rounded to nearest integer.

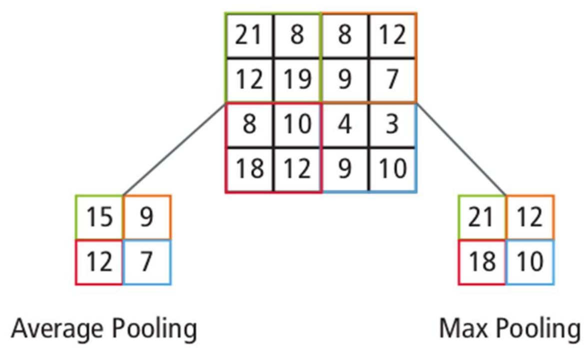

Figure 3: Representation of max and average pooling

3.4 Fully Connected Layer

In a convolutional neural network, the fully connected layer is frequently used as the final layer. These layers mathematically sum the weights of the previous feature layers to indicate the accurate combination of "ingredients" to determine a specifically required target output result. For fully connected layers, all elements of all features in the previous layer are used to calculate each element of each output feature. A fully Connected Layer is simply a feed forward neural network. The following figure shows a fully connected layer L. Layer L1 has two functions, each of which is 2x2 i.e., there are four elements. Layer L has two functions, each with one element.

ISSN (Online) 2581-9429

IJARSCT

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 4, May 2022

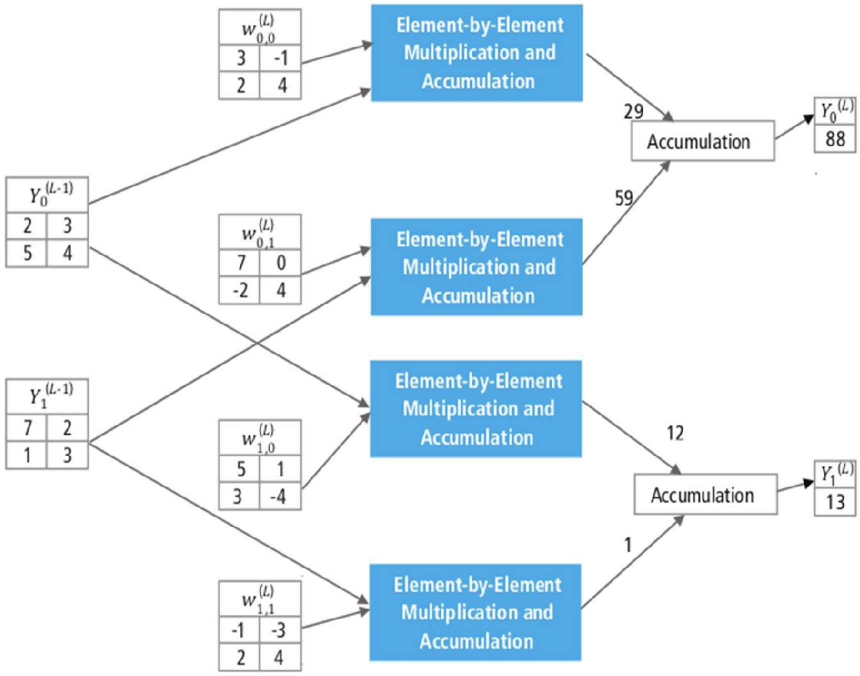

Figure 4: Fully connected layer processing

3.5 Why CNN

The Neural networks and other pattern recognition methods have existed for about 50 years, but recent significant developments have been made in the field of a convolutional neural networks. This section describes the benefits of using a CNNs for image recognition.

3.6 Ruggedness to Shifts and Distortion in the Image

Detection using CNN is rugged to distortions such as change in shape due to camera lens, different lighting conditions, different poses, presence of partial occlusions, horizontal and vertical shifts, etc. However, CNNs are shift invariant since the same weight configuration is used across space. In theory, we also can achieve shift invariant-ness using fully connected layers. But the outcome of training in this case is multiple units with identical weight patterns at different locations of the input. To learn these weight configurations, a large number of training instances would be required to cover the space of possible variations.

3.7 Less Memory Requirements

Extracting features using fully connected layers. In the same hypothetical case, an input image of size 32x32 and a hidden layer with 1000 features would require an order of 106 coefficients, which has huge memory requirements. In the convolutional layer, the same coefficients are used at different locations in space, significantly reducing memory requirements.

3.8 Easier and Better Training

Again, with a standard neural network that supports CNNs, the number of parameters is much larger, so training time increases proportionally. CNN significantly reduces the number of parameters, resulting in a proportional reduction in training time. You can also design a standard neural network that acts like a CNN, given perfect training. However, in actual training, standard CNN-compliant neural networks have more parameters and increase the noise penalty during the training process. Therefore, the performance of standard neural networks equivalent to CNNs will always be degraded.

IV. Implementation of System

While implementing the system we used the TensorFlow, NumPy, matplotlib library and Jupyter notebook to build and train the model. Finally, to make the system easy to use for the farmer a website is developed where a farmer can upload the image of the leaf of the plant, he wants to diagnose. The website will put that image into the model for analysing and the model will predict the result in the form of labels which is the class of the disease from belong to that leaf and the confidence of how much the plant is affected by the disease. If the plant is healthy, it will show the class as healthy and the and confidence of the healthiness.

Table 1: Performance Comparison with Different Deep Learning Models

Author	**Deep Learning Model**	**Dataset**	**Results**
Mohanthy , etc.(2015)	AlexNet,GoogleNet	Plant Village	85% to 99%(Projected)
Sladojevic , etcl(2016)	Fine tune CNN	Own	96.3%(Precision)
Fuentes etc(2017)	Faster R-CNN	Own	83%(Testing)
Geetharamani ,etc.(2019)	Multiple CNN	Plant Village	96%(Testing)
Zeng ,etc(2020)	Self Attention CNN	MK-D2 AES-CD9214	95% 98%
Yan Li ,etc(2020)	Shallow CNN	Plant Village	94%
Oyewola, etc(2021)	Deep Residual CNN	Cassava	96.75%
Sk Mahmudul Hassan etc(2022)	Novel CNN	Plant Village	99%(Projected)
Stefania Barburiceanu, etc (2021)	ResNet50	Plant Village	82%
Tan Nhat Pham,etc(2020)	TL with ResNet-50	Plant Village	84%

V. Conclusion

There are several ways we recognize plant diseases and suggest relief measures for you. Everyone has some experts and restrictions. On the other hand, visual analysis is the most expensive and easy way, which is not efficient and reliable. Image processing is a technique that is most spoken and is most talked and the most consumable advantage is provided. The experimental results showed that it is a valuable approach that can significantly support the exact detection of leaf diseases in low computational efforts. Using this approach, we achieved about 97% accuracy.

Acknowledgment

We would like to take this opportunity to thank our guide A. P. Kale for giving us all the help and guidance we needed and for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for this project.

References

[1]. Stefania Barburiceanu, Serban Meza, Bogdan Orza, Raul Malutan, Romulus Terebes "Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture" IEEE Access (Volume: 9) Page(s): 160085 – 160103.

[2]. V. Singh, N. Sharma, and S. Singh, ‘‘A review of imaging techniques for plant disease detection,’’ Artif. Intell. Agricult., vol. 4, pp. 229–242, Oct. 2020

[3]. M. H. Saleem, J. Potgieter, and K. M. Arif, ‘‘Plant disease detection and classification by deep learning,’’ Plants, vol. 8, no. 11, pp. 468–489, Oct. 2019

[4]. B. Liu, C. Tan, S. Li, J. He, and H. Wang, ‘‘A data augmentation method based on generative adversarial networks for grape leaf disease identification,’’ IEEE Access, vol. 8, pp. 102188–102198, 2020

[5]. S. P. Mohanty, D. P. Hughes and M. Salathé, "Using deep learning for image-based plant disease detection", Frontiers Plant Sci., vol. 7, pp. 1419, Sep. 2016

[6]. F. Alvaro, Y. Sook, K. Sang and P. Dong, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition", Sensors, vol. 17, no. 9, pp. 2022, 2017

[7]. S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic, "Deep neural networks-based recognition of plant diseases by leaf image classification", Comput. Intell. Neurosci., vol. 2016, pp. 1-11, May 2016

[8]. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis", Comput. Electron. Agricult., vol. 145, pp. 311-318, Feb. 2018

[9]. G. Geetharamani and P. J. Arun, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network", Comput. Electr. Eng., vol. 76, pp. 323-338, Jun. 2019.

[10]. E. C. Too, L. Yujian, S. Njuki and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification", Comput. Electron. Agric., vol. 161, pp. 272-279, Jun. 2019.

[11]. P. K. Sethy, N. K. Barpanda, A. K. Rath and S. K. Behera, "Deep feature-based Rice leaf disease identification using support vector machine", Comput. Electron. Agricult., vol. 175, Aug. 2020

[12]. Q. Wu, Y. Chen, and J. Meng, ‘‘DCGAN-based data augmentation for tomato leaf disease identification,’’ IEEE Access, vol. 8, pp. 98716–98728, 2020

[13]. LILI LI, SHUJUAN ZHANG, AND BIN WANG, "Plant Disease Detection and Classification by Deep Learning—A Review", IEEE Access, 10.1109/ACCESS.2021.3069646, April 8, 2021

[14]. PENG JIANG, YUEHAN CHEN, BIN LIU, DONGJIAN HE, AND CHUNQUAN LIANG "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks "IEEE Access, /ACCESS.2019.2914929 May 6, 2019

[15]. A. Johannes et al., ‘‘Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case,’’ Comput. Electron. Agricult., vol. 138, pp. 200–209, Jun. 2017

[16]. TAN NHAT PHAM, LY VAN TRAN, AND SON VU TRUONG DAO "Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection" IEEE ACCESS.2020.3031914, October 19, 2020

[17]. J. G. A. Barbedo, ‘‘Factors influencing the use of deep learning for plant disease recognition,’’ Biosyst. Eng., vol. 172, pp. 84–91, Aug. 2018, doi: 10.1016/j.biosystemseng.2018.05.013

[18]. V. Singh and A. K. Misra, ‘‘Detection of plant leaf diseases using image segmentation and soft computing techniques,’’ Inf. Process. Agricult., vol. 4, no. 1, pp. 41–49, Mar. 2017, doi: 10.1016/j.inpa.2016.10.005