

International Journal of Intelligent Systems Technologies and Applications

ISSN online: 1740-8873 - ISSN print: 1740-8865
<https://www.inderscience.com/ijsta>

Deep learning model for plant-leaf disease detection in precision agriculture

Chandrabhanu Bajpai, Ramnarayan Sahu, K. Jairam Naik

DOI: [10.1504/IJISTA.2023.10055077](https://doi.org/10.1504/IJISTA.2023.10055077)

Article History:

Received:	06 January 2023
Last revised:	07 January 2023
Accepted:	27 February 2023
Published online:	27 April 2023

Deep learning model for plant-leaf disease detection in precision agriculture

Chandrabhanu Bajpai, Ramnarayan Sahu and
K. Jairam Naik*

Department of Computer Science and Engineering,
National Institute of Technology Raipur,
Raipur, Chhattisgarh, India
Email: bhanubajpai75@gmail.com
Email: ramsatya13970@gmail.com
Email: Jnaik.cse@nitrr.ac.in
*Corresponding author

Abstract: Crop disease in the agricultural is the main factor limiting yield and food quality. It requires timely diagnosis of crop illnesses for the better economy developing country. Manual crop illness assessment is limited due to lesser accuracy and restricted accessibility. It is very difficult to accurately identify and classify plant diseases due to corrupt in the data samples, lesser intensity of foreground and background, and the extreme similarity between unhealthy and healthy leaves in terms of colouring and size of crop leaves. Hence the employments of automated and computerised optimisations are needed. To identify plant leaf diseases, a DNSVM classification strategy fusing DenseNet-201 with support vector machine (SVM) is proposed in this work. Plant-Village dataset that provides good-variations, colour-differences, differences in orientation and size-of-leaves. Sugarcane plants-leaves were used for performance analysis of proposed model and obtained 97.78% of classification accuracy over the existing DenseNet-121-based classifier model (94%).

Keywords: plant leaves; disease detection; deep learning models; agriculture; classification; DenseNet-201.

Reference to this paper should be made as follows: Bajpai, C., Sahu, R. and Naik, K.J. (2023) 'Deep learning model for plant-leaf disease detection in precision agriculture', *Int. J. Intelligent Systems Technologies and Applications*, Vol. 21, No. 1, pp.72–91.

Biographical notes: Chandrabhanu Bajpai received his Bachelor of Technology in Computer Science and Engineering from the National Institute of Technology Raipur, India in 2023. Her research interest includes image processing, deep learning, and reinforcement learning.

Ramnarayan Sahu received his Bachelor of Technology in Computer Science and Engineering from the National Institute of Technology Raipur, India in 2023. His research interest includes IoT, and deep learning.

K. Jairam Naik is working as Assistant Professor in the Department of Computer Science and Engineering, National Institute of Technology Raipur, India since 2018. He received his PhD in Computer Science and Engineering from the Jawaharlal Nehru Technological University Hyderabad, India in 2017.

He received his MTech and BTech in Computer Science and Engineering from the JNTU Hyderabad in 2009 and 2003 respectively. His research of interest includes grid and cloud computing, edge/fog, and IoT computing.

1 Introduction

A United Nation-based organisation named as Food and Agriculture Organization (FAO) has predicted that by 2050, there will be 9.1 billion people on the planet (Bruinsma, 2009). Therefore, to fulfil the nutritious requirements for such a highly populated planet, food and crop's growth rate across world should be increased to 70% till the year 2050. Although, many problems, including absence of sizable farmable land and clean supply of water, limit the growth of the crop production industry. Aside from that, agricultural illnesses drastically lower crop yields and quality. Such losses have a big impact on the nation's economy, which eventually reduces farmer income and increases crop cost for consumers. Additionally, it may lead to major grains scarcity following protracted periods of starvation and deprivation, specifically in less evolved nations with limited access to preventive measures. Human resources generally find crop illnesses by physically visiting the farms and inspecting the crops. However, it would take a long duration and inspection of each, and every plant becomes practically impossible by the humans. Crop disease analysis techniques frequently lag due to the shortage of labour. Early disease diagnosis is crucial since it not only enhances grains production but also protects farm workers from costly sprinkling methods. The research community has switched its focus to the introduction and analysis of machine technologies to facilitate this difficult procedure to address the drawbacks of manual plant disease diagnosis. For crop analysis also they have created automated technique.

Methods like immunology and molecular biology were initially used to identify diseases on crop at their early stages (Dinh et al., 2020). However, these strategies needed to be formed, they needed a lot of resources, and they needed human expertise. The bulk of agricultural sites, according to FAO, are small and are managed by low-income residents of developing countries (Ferentinos, 2018). As a result, such pricey solutions are unworkable for them, and researchers must suggest reasonable, workable alternatives for all farmers. To improve decision-making capability, innovative machine learning-based hand-coded algorithms are currently being used in agriculture (Rokach and Maimon, 2005). A significant quantity of data is being captured in real-time thanks to the development of digital techniques, and this data is then used to create ML-based strategies that help people make better decisions.

These techniques, including decision trees (DTs) (Rokach and Maimon, 2005), K-nearest neighbours (KNN) (Liao and Vemuri, 2002), support vector machine (SVM) (Joachims, 1998), Gaussian frameworks (Birgé and Massart, 2001), etc. have also undergone extensive testing for the diagnosis of crop diseases. Methods for computing hand-coded key-points are simpler to use and do not require a large dataset for training, but they take a lot of time and require human experience. Additionally, there is always a choice between computationally heavy and classification reliability in the traditional ML-based features calculation approaches. Using a narrow key-points or features decreases the system's localisation efficiency while computing a big key-points vector

increases the financial burden. Therefore, efficiency improvements are still required, especially for decision-based architectures which help to turn the vast amounts of features into insightful advice.

Since then, we have seen the effectiveness of deep learning (DL) utilising methods like convolutional neural network (Roska and Chua, 1993), recurrent neural networks (Zaremba et al., 2014), and deep belief networks (Hinton, 2009; Salakhutdinov and Hinton, 2009) in a variety of research fields such as data categorisation, segmentation, alteration detection, and agricultural areas. The best features from the input data can be automatically identified by DL-based approaches, such CNN, which does not require any assistance of agricultural specialists. Operation residing in DL architecture mimics how the brain of human processes information. By looking at several different samples of an object, people may visually recognise and locate it. For object and pattern recognition, these methods use the same process. For better decision-making, DL models produce highly accurate performance than bespoke machine learning (ML) oriented techniques. Because of the rapid advancement of computer hardware technologies, DL architecture are being extensively researched to find solutions to difficult problems in a respectably less duration of time. In the sphere of agriculture, DL-based approaches demonstrate cutting-edge precision and generalised well to various jobs.

Deep neural network (DNN) models in various configurations have demonstrated notable performance over hyperspectral analysis. GoogLeNet, AlexNet, VGG, and ResNet are some of the well-known CNN models that did well in classification tasks connected to crops, such as crop harvest prediction, counting fruits, crop disease finding, and categorisation, among other tasks. These frameworks can produce reliable results with the least amount of pre-processing and computing work. Additionally, post-processing vulnerabilities like translation, rotation are there in DL oriented algorithm sand provide higher results for localisation of the object since it utilises topological features from the dataset of input images. With pre-trained frameworks, these methods also produce output of higher quality. There is always potential for improvement even though a significant quantity of research has been conducted on agricultural disease identification and categorisation. In comparison to human brain intelligence, ML techniques are remarkably better at solving complex real-world situations. Since these architectures generate longer length codes which makes it computationally heavy, the main problems with ML oriented approaches on plant disease categorisation and identification are their lower efficiency and lengthy training times. However, at the expense of increased code complexity, DL technique has been created to handle the problem of long codes. Because there are such wide differences in leaf size, shape, colour, and location, it is still difficult to localise and classify crop diseases in an efficient and effective manner. Additionally, the detecting procedure is made more difficult by the brightness changes that occur while taking pictures of leaves. By creating a classification approach called DN-SVM which is developed by the combination of DenseNet-201 framework and SVM, we attempted to look on the problems highlighted above in this paper. We perform training using training dataset and take out leaf key points from the completely connected layer using DenseNet architecture, which has the advantages of a densely connected mechanism and reusability of features. The SVM is then given the vector containing features as an input to recognise plant leaf disease. The technique is resilient to shape of the images, any rotation to the images, different colour, and varied brightness, different contrast, changing lightning conditions, opaqueness, and large

numbers of corrupted input data, according to our findings. The following are the primary contributions of the newly introduced framework:

- To increase the identification and categorisation performance of crop illness while reducing the learning and testing time, we provide a customised model using DenseNet-201 for features computation.
- The resilience of the DenseNet model in our strategy enables precise localisation of the afflicted leaf area.
- The ability of the DenseNet architecture to handle the over-fitting problem in the data allows our strategy to obtain enhanced classification accuracy of plant leaf diseases.

2 Literature review

In this section, we have carried out a critical study of the literature-based research used for classifying and identifying plant diseases. The study for classifying plant diseases is of two types: models on ML or models based on DL.

A technique to detect and classify diseases of crops and weeds was introduced by Le et al. (2020). To reduce the distortion in the images given as input, morphological opening and closing procedures were first carried out. From the sample processing, features were computed from a tailored architecture called the contour mask and coefficient k method for filtered local binary patterns (k-FLBPCM). The SVM categoriser training was done to categorise various crop diseases using the retrieved features. Although the method in Le et al. (2020) exhibits improved plant disease classification accuracy, samples with perspective distortions may not respond favourably to it. Using a framework, Ahmad et al. (2020) suggested finding and categorising plant diseases. Directional local quinary patterns (DLQP) was used to compute the key points in the input samples in the first stage. Results of the plant disease classification were then obtained by training the SVM categoriser on the compute features. Although this method demonstrates higher crop disease identification performance still accuracy can be enhanced by using the input sample's shape and colour-based information.

A model to recognise and categorise tea crop diseases was given forth by Sun et al. (2019). Blocks from the input sample were initially separated using the simple linear iterative cluster (SLIC) technique, after The Harris approach was then applied to extract the most relevant points. The fuzzy salient region contour was obtained utilising convex hull technique in the following phase, and the key points were extracted using the grey level concurrence matrix (GLCM) method. The SVM classifier was then trained to perform the categorisation of tea crop leaf illnesses. Although the technique used by them has higher classification accuracy, it has a high computational cost. An approach to classify and categorise various crop diseases was given by Pantazi et al. (2019). Initially, the GrabCut method was used to segment the input image. After that, the separated data underwent the HSV conversion. The features from the generated ROIs were then created by LBP and utilised to train the class SVM categoriser. Though their given approach done good for categorising crop illness, corrupted data gave poor identification performance. Like Oo and Htun (2018), a hand-coded key point retrieval-oriented method for classifying and identifying crop diseases was provided. To improve the visual

quality of the photos, the input samples were scaled in the first phase and then the histogram equalisation (HE) approach was used over them. The segmentation was then carried out across the processed samples using the K-means clustering method. The segmented regions were then subjected to the use of GLCM and LBP descriptors to calculate the key points. The SVM categoriser received training using the computed features in the last stage to perform the categorisation of plant diseases. Although results are presented for a limited dataset, their approach exhibits superior classification accuracy.

Ramesh et al. (2018) presented a method for classifying plant diseases. The samples were divided into healthy and diseased categories using features from the histogram of oriented gradients (HOGs), which were then utilised to train the random forest (RF) classifier. Although their approach is reliable for classifying plant diseases, performance still has to be enhanced. Kuricheti and Supriya (2019) developed a system for categorising illnesses of turmeric leaves. After pre-processing, the K-means method was used to conduct image segmentation on the input image. Following the application of the GLCM technique for feature extraction, the training of SVM categoriser was done to classify the crop leaves. This technique shows superior results for classifying plant diseases but is not able to perform good on data with significant intensity changes. From the ML oriented techniques that have been discussed above, it can be inferred that this research is straightforward to implement yet they require a lot of learning samples and heavily rely on the human experience. Moreover, existing methods are not resistant to wide differences of dimensions, hues, and forms of leaf crop illness. Further to increase the detection performance of various plant leaf illnesses, a more robust strategy utilising the most recent approaches is therefore required (Liu et al., 2021; Abdollahi and Pradhan, 2021). Due to their high recall rate, DL methods have recently received a lot of attention in a number of automated applications. Few-shot learning (FSL), a DL-based approach to identifying and categorising plant disease, was introduced by Argüeso et al. (2020). Initially, the significant features were calculated utilising the Inception V3 architecture. The next step involves a multi-class support vector.

Due to their high recall rate, DL methods have recently received a lot of attention in several automated applications. FSL, a DL-based approach to identifying and categorising plant disease, was introduced by Argüeso et al. (2020). Initially, the important points were calculated using the Inception V3 framework. Then, using the retrieved features as training data, a multi-class SVM was created. Although results are given on a limited dataset, given technique is reliable to classifying crop diseases. It should thus be analysed using vast and varied collections. To identify and categorise the tomato plant illness, Agarwal et al. (2020) suggested a CNN-based design. Three convolutions and a max-pooling layer were used in this method, to separate the important information from the input samples and categorise it. Although this method (Kuricheti and Supriya, 2019) shows improved accuracy in classifying tomato diseases, it has the drawback of being over-fitted across a limited number of classes. To identify and categorise maize crop disease, Richey et al. (2020) devised a mobile application-oriented method. To calculate the input's deep key points, photos and classifying them into appropriate categories, the ResNet50 over the ImageNet database, a DL oriented method was trained. Their method offers a smart phone-oriented approach to plant illness categorisation, but due to limitations in memory, processing power, and battery life, given technique is computationally challenging but not well suited for smartphone. An

enhanced DL oriented architecture for categorising tomato plant illness was given by Zhang et al. (2020).

A strategy was put forth by Batool et al. (2020) to recognise and classify tomato crop illness at an initial stage. Initially the AlexNet architecture was used to retrieve data from this the input sample's deep key-points that were later employed to program the KNN to identify healthy photos from the input or influenced. KNN is an algorithm that is accurate yet sluggish and time-consuming. A DL-based model was presented by Goncharov et al. (2020). For categorising illnesses of different crops, such as corn and wheat. The computation of the deep Siamese network k attributes that were ultimately utilised to train the image KNN categorisation. Their framework can withstand being planted. Illness detection, however, has the issue of overfitting of a big dataset. A method based on DL was presented by Karthik et al. (2020) to identify illness in tomatoes leaves. A residual network was used in to calculate intricate properties of the supplied samples. Afterward, a convolutional neural network categoriser was taught to calculate important features to distinguish between the good, impacted leaves, etc. Their given strategy displays better leaves. Although this approach accurately classifies diseases, it is not cost-effective. A DL-based approach was presented by Tm et al. (2018) methodology for locating and classifying illnesses in tomato leaves. The input samples were initially scaled before being used for processing later. The samples were then separated into classes of healthy and damaged samples using a DL model called LeNet. Their method offers a lower-costing method for classifying tomato plant diseases; although it is not able to demonstrate reliable accuracy for corrupted samples. The categorisation of tomato plant leaf illnesses was suggested by Sembiring et al. (2021). A four-layer, lightweight CNN framework was utilised to compute the deep key points of suspected data and classify them into ten separate groups. Although this approach is computationally effective, it performs less well in current situations. Turkoglu et al. (2021) developed an ensemble method where the deep key points of various plants were computed using several DL oriented architecture, including AlexNet, GoogleNet, and DenseNet-201, ResNet50, and ResNet101 architectures. The computed characteristics were then used for SVM training to classify various crop illnesses in the following process. However, their method's improved crop leaf disease categorisation accuracy, which comes at the cost of more expensive key point's calculation.

The work in the article by Shaheen et al. (2020) presents the vital role of precision farming that outlines a through systematic literature review. This article also identifies the various opportunities with the adoption of precision farming in India associated challenges. Adopting sustainable farming practices and their effects on smallholders was elaborated by the studies in Ouiminga (2022). The work by Naik and Soni (2020) and Naik et al. 2021a, 2021b) have made various innovations in DL and the literature of that work helped in proceeding with the novelty of the proposed DL model for leaf image processing.

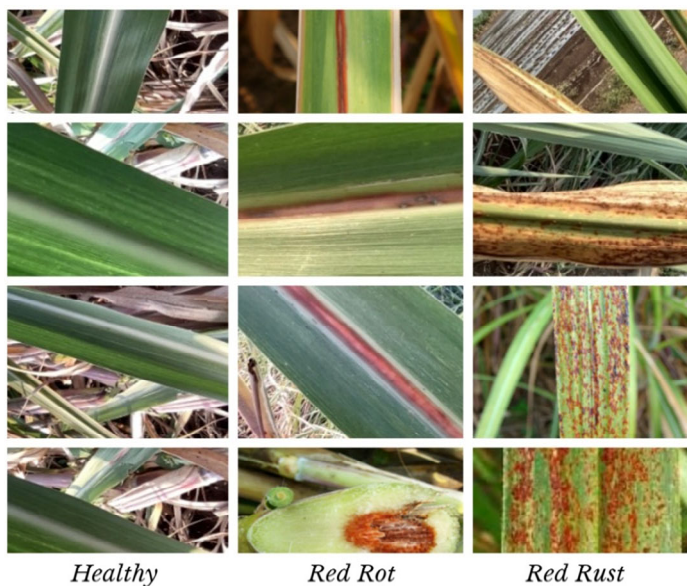
3 Material required

3.1 Image dataset

For the experimental studies of this project, we have used a part of a Plant-Village (<https://drive.google.com/drive/folders/1EbVaGV072ltejuDOEe1qoT11hGtQ89xL?usp=s>

hare_link) dataset. An international general picture dataset called Plant-Village is used to train ML algorithms for identifying plant diseases. It comprises 38 categories which include 12 healthy crops and 26 ill categories for 14 plant types. The 54,306 crop leaf photos in Plant-Village were photographed in both greyscale and colour under carefully regulated conditions. All photos were taken on plain backgrounds with consistent lighting intensities. Keep in mind that there aren't always the same amounts of samples in each category. In addition, several of the photographs were captured from multiple angles on the same plant leaf. We can see the sample images of the Plant-Village dataset of sugarcane plant from the Figure 1 where we can see 3 types of categories of leaf that are healthy, red rot and red rust respectively. Description of different kind of disease found in sugarcane is presented in Table 1 as shown.

Figure 1 Representing samples from Plant-Village dataset for sugarcane (see online version for colours)



4 Proposed methodology

Data preparation, key points (features) extraction, and categorisation are the three primary components of the suggested methodology. To reduce over fitting, we first perform cropping, rotation, colour processing, and other procedures on the photos of the plant leaf dataset. The DenseNet-201 architecture is then used to take out important features from images of crop leaves. The dense connectivity mechanism of the DenseNet architecture is one of its distinguishing features. It exhibits improved accuracy with lower number of parameter and less cost of computation and facilitates reusability of features by joining features on channels. To classify leaf images, the key point vectors are finally used as input by the SVM categoriser.

Table 1 The key features of sugarcane plant diseases

Name of the disease	Disease features		
	Outline	Colour	Location
Red rot	Transverse white spots	Black fruiting bodies	Rind
Wilt	Boat shaped cavities	Yellow	Ground tissue
Grassy shoot	Thin with short internodes	Pale yellow	Leaf
Smut	Whip like structure	Black	Leaf
Leaf scald disease	Pencil lines	White and yellow	Leaf
Red striped disease	Dark stripes	Red	Leaf
Mosaic disease	Mosaic pattern	Green	Leaf
Pokkahboeng	Transverse cuts	Dark brown	Rind
Rust	Spots	Yellow	Leaf
Sugarcane disease	Yellow midrib	Yellow	Under leaf

4.1 Data pre-processing

First light, angle of placement, size of lens, and different variables frequently affect the shot plant leaf in real application settings, which may differ somewhat from the training set. As a result, we increase the model's capacity for generalisation through data improvement, which includes the following five steps:

- a Randomly rotate the image by 0 to 30 degrees with a 50% probability.
- b Randomly crop an image.
- c Change the resolution of the image to 224×224 pixels.
- d Image colour alterations at random: 50% of the time, colour perturbation occurs in the steps of intensity, hue, balanced and, contrast while the other 50% occurs in the steps of intensity, contrast, hue, and balance.
- e Rotate the data horizontally and vertically by a 50% chance.

4.2 Synthetic minority oversampling technique

The samples utilised in this work are classified as an unbalanced sample because each class has a considerably different amount of data than the three other classes. The oversampling method was used in various research to replicate the dataset with the least quantity of data as much as the most dominant dataset so that it can anticipate the issue of offset datasets.

For minority classes, it is one of the types of sample augmentation and it referred to us synthetic minority over sampling technique. One of the issues with imbalanced categorisation is of very few instances of minority category and thus makes difficult for a model to learn the decision boundary. We can solve this issue by doing oversampling of the instances of minority category, which duplicates the instances of minority category in the learning dataset. This solution balances the distribution across different categories but does not give any additional information to the model.

The most used technique for creating new samples is synthetic minority oversampling technique (SMOT). It selects instances on the feature space that are close to one another and draws a line between the instances and then creates a new data at a location along the drawn line. This technique was used in our research.

4.3 Batch normalisation

The act of changing data so that the standard deviation is 1 and the mean is 0 is called as normalisation. Batch input from layer h is there in this stage, hence, mean of the hidden activation must be computed initially as mentioned in equation (1).

$$mean(\mu) = \frac{1}{n} \sum h_i \quad (1)$$

Here n represents the number of neurons at layer h . After the above step is to determine the standard deviation of the hidden activations after we have calculated the mean as per equation (2).

$$\sigma = \left[\frac{1}{n} \sum (h_i - \mu)^2 \right]^{1/2} \quad (2)$$

Furthermore, the mean and standard deviation are available. Using these numbers, we will normalise the hidden activations. To do this, we will take the mean from each input and divide the total by the smoothing factor (ϵ) and the sum of the standard deviation as per equation (3).

$$h_{i(norm)} = \frac{(h_i - \mu)}{\sigma + \epsilon} \quad (3)$$

By preventing division by a zero value, the smoothing term (ϵ) ensures numerical stability inside the operation. Batch normalisation is executed on smaller batches instead of whole data. It lowers the training time by using larger learning rate and thus helps in making the learning process easier. Batch normalisation is used in our DenseNet-201 architecture.

4.4 Feature extraction using DenseNet

All layers are interconnected by the DenseNet design, and each layer provides input to the subsequent layer by being connected to all levels below it on the channel dimension (Huang et al., 2016). There are $L * (L + 1) / 2$ connections in total for the L network (Zou et al., 2015). Equation (4) illustrates how all the earlier levels in the DenseNet design are coupled as an input. The $H_l(\cdot)$ stands for a nonlinear transformation function and concatenation of the feature maps generated by the layers $[0, 1, 2, \dots, L^{\text{th}}]$ is $[i_0, i_1, i_2, \dots, i_{l-1}]$.

$$x_l = H_l([i_0, i_1, i_2, \dots, i_{l-1}]) \quad (4)$$

The architectural structure of DenseNet comprises of transition layer and DenseBlock. Each layer's feature map in the DenseBlock can be joined along the channel dimension

and has the same shape. DenseBlock and the transition layer make up the majority of DenseNet’s network topology.

4.4.1 Max pooling

The phrase ‘max pooling’ refers to a pooling procedure that picks the biggest element from the feature map space that the filter occupies. We are using here max pool (3×3) with stride 2 in our DenseNet-201 architecture.

4.4.2 Avg. pooling

By using the average pooling operation, average value for a particular area of a feature map is derived, to generate a pooled feature map. After a convolutional layer it may be utilised. We are using avg. pooling (2×2) with stride 2 in 3rd transition layer of DenseNet-201.

$$Output = \frac{input - kernel_size + 2 * padding}{stride} + 1 \quad (5)$$

The output size of both the pooling operation (i.e., max and avg. pooling) can be calculated as per equation (5).

4.4.3 DenseNet-201 architecture

We used DenseNet-201 layered architecture in our model. It comprises of three transition layers and four numbers of DenseBlocks where there is a rate of growth of 32. In between the successive DenseBlocks, transition layers have been introduced. These transition layers comprise of 1×1 convolution layer and a max pooling with stride 2. Dropout layer and reduction layer has not been used in our framework. We reduced the dimensions of output key point vector using global average pooling and the key points are finally categorised using SVM categoriser. This layer is utilised to predict the output of every category. The idea for using this architecture is based on the characteristic of categorisation that we want to do. The captured images from the mobile phone cameras contain very minute key points of the images. Categorising these key points are difficult, so we also perform post pre-processing on our sample which changes the geometrical characteristics of the model key points in an unpredictable way. Hence, this technique can detect models’ key point under all difficult and constrained conditions, although training of such deep architecture to identify minute model key points is of greater difficulty. The framework almost every now and then becomes dependent on the sample or some specific corrupt (noise), since all the small statistical related information gets lost when the sample is transmitted subsequent layers.

The above issue can be reduced in the DenseNet using densely connected architecture. For preserving the sample information throughout the architecture, the output from all the network’s previous layers is given as input to all the layers in front of previous layers. Even if there is a loss of key points because of some operations, it can be recreated from previous layer’s output through this densely connected network. This helps in preventing the flow of gradients from vanishing at the time of learning and allows us to take out the key points that are very difficult and cannot be identified by conventional CNN framework.

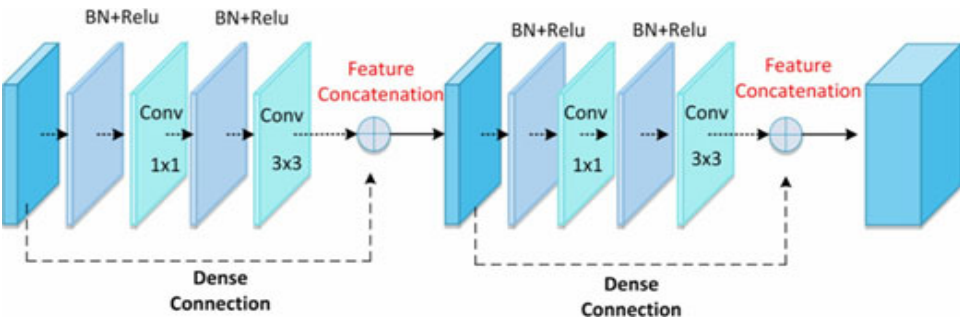
The description of the DenseNet framework which we used for our plant leaf disease detection model can be looked out from Table 2.

Table 2 Specifications DenseNet-201 architecture

Layers	DenseNet-201
Convolution	7×7 convolution with stride 2
Pooling	3×3 max pooling with stride 2
DenseBlock (1st)	6 times \rightarrow 1×1 convolution 3×3 convolution
Transition layer (1st)	1×1 convolution 3×3 max pooling with stride 2
DenseBlock (2nd)	12 times \rightarrow 1×1 convolution 3×3 convolution
Transition layer (2nd)	1×1 convolution 2×2 average pooling with stride 2
DenseBlock (3rd)	48 times \rightarrow 1×1 convolution 3×3 convolution
Transition layer (3rd)	1×1 convolution 2×2 average pooling with stride 2
DenseBlock (4th)	48 times \rightarrow 1×1 convolution 3×3 convolution
Layer	Global average pooling
Classification	Support vector machine (SVM)

From the architecture of our model, we can see that we have used four DenseBlocks with 1×1 convolution and 3×3 convolution. The pictorial representation of the DenseBlock and the dense connection and feature concatenation is represented in Figure 2.

Figure 2 DenseBlock layered architecture (see online version for colours)



The pre-trained models on prior issues are utilised to solve current problems, and DenseNet-201 is one of the DL transfer learning methods that do this. The CNN database contains a variety of pre-treatment models, including AlexNet, VGGNet, LeNet, and

DenseNet. In this paper, we focus on feature extraction from plant leaf photos using a pre-trained DenseNet-201.

Table 3 represents the architecture summary of our model where type of layers in model, out shape of images and number of parameters are respectively mentioned as shown in the table.

Table 3 Summary of proposed model architecture

<i>Layer</i> → <i>type</i>	<i>Output shape</i>	<i>Parameters #</i>
input_2 → input layer	[(None, 224, 224, 3)]	0
DenseNet-201 → functional	(None, 1,920)	18,321,984
dense → dense	(None, 256)	491,776
dropout → dropout	(None, 256)	0
dense_1 → dense	(None, 128)	32,896
dropout_1 → dropout	(None, 128)	0
dense_2 → dense	(None, 3)	387
<i>Total params: 18,847,043</i>		
<i>Trainable params: 525,059</i>		
<i>Non-trainable params: 18,321,984</i>		

4.5 SVM classifier

SVM, a method of data mining was created using the statistical learning theory (SLT) (Xu et al., 2020). It was developed by Boser, Guyon, and Vapnik and was first proposed in 1992. It offers several benefits for tackling regression and classification issues with few data, nonlinear data, and greater dimensions of the data. Support vector categorisation, which uses the learning dataset as the object for the data, analyses the quantitative relationship between the input variable and the categorised output to predict the category value of the output variable of the new data. We create a new coordinate plane, place the leaf features we extracted from the data on it, and use SVM to categorise the sample characteristics in the hyper plane. The SVM, a linear model, is used to solve categorisation and regression issues. It works well for a range of real-world issues and can solve both linear and nonlinear problems.

In our experiment, 80% of the photos serve as the learning set, while 20% serve as the testing set while training. To increase the model’s capacity for generalisation, we first perform data improvement operations on the image. The feature extraction module then uses the DenseNet architecture to extract leaf feature vectors from the whole connection layer. Finally, SVM is used to identify plant leaf illness using the feature vectors that have been processed.

Hinge loss: a loss function called the hinge loss is employed while training classifiers, most notably the SVM. Calculating hinge loss is actually fairly easy. The following equations (6) and (7) give the formula for hinge loss:

$$l = \max(0, 1 - y^i (x^i - b)) \tag{6}$$

The formula can be rewritten as follows:

$$l = \begin{cases} 0 & \text{if } y \cdot (w \cdot x) \geq 1 \\ 1 - y \cdot (w \cdot x) & \text{otherwise} \end{cases} \quad (7)$$

Here, $Y[i]$ and $X[i]$ denote the i^{th} instance in the training set, l is the loss of any given instance, and b denotes the bias term. In this way hinge loss is calculated and used in training of SVM classifier.

5 Result analysis

This part offers a thorough examination of the findings from numerous experiments to gauge the efficacy of the suggested techniques. On a Google Collab platform, the proposed framework's Python implementation was carried out.

5.1 Evaluation metrics

Different assessment criteria, such as recall, accuracy, precision, and mean average precision, were used for evaluation. To calculate this assessment, we need variables like true positive (TP), true negative (TN), false positive (FP) and false negative (FN) and they are defined as follows:

- TP: Output given by the model is positive and is actually positive.
- TN: Output given by the model is negative and is actually negative.
- FP: Output given by the model is positive and is actually negative.
- FN: Output given by the model is negative and is actually positive.

The accuracy was calculated using equations (8):

$$Accuracy = \frac{TP + NP}{TP + FP + TN + FN} \quad (8)$$

In equations (9) and (10) represents the precision and recall (true positive rate or sensitivity) respectively:

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \quad (9)$$

$$Recall\ or\ TPR = \frac{True\ positive}{True\ positive + False\ negative} \quad (10)$$

Specificity or true negative rate can be calculated as given in equation (11).

$$TNR = \frac{True\ negative}{True\ negative + False\ positive} \quad (11)$$

$$F1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

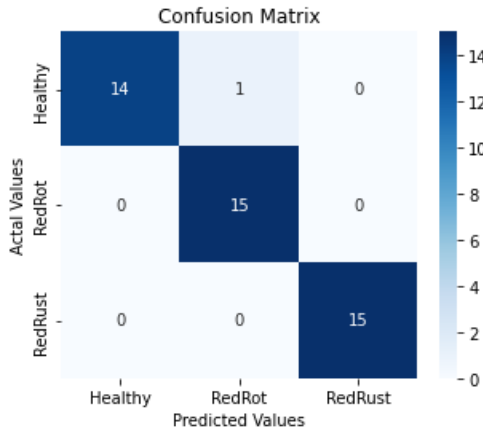
F1-score mentioned in equation (12) represents harmonic mean (HM) of recall and precision and consider giving importance to these two parameters.

5.2 Confusion matrix

A table called a confusion matrix represents how many correct and incorrect classifications a categoriser made. This is used to assess a categorisation model’s correctness. The performance of a DL algorithm is typically our main concern. However, we must also consider the ratio of correctly and categorised data if we are trying to solve a categorisation issue. Therefore, we require a technique that, in addition to performance, helps in determining accurate categorisation. The confusion matrix accomplishes this. Confusion matrix is a $m \times m$ matrix that aids in assessing how well a DL technique performs when applied to a categorisation issue. It can be used to figure out performance indicators like accuracy, precision, recall, and F1-score to estimate how well a categorisation technique is doing. Confusion matrix is widely employed as it gives a correct view of a model’s correctness than categorisation accuracy does.

Confusion matrix for our model categorisation is shown in Figure 3. Y-axis of the confusion matrix contains the actual values of different categories and x-axis contains the values predicted by our model. Dark shaded cells of confusion matrix represent the true positive values of different categories.

Figure 3 Confusion matrix (see online version for colours)



5.3 Performance evaluation

By using the data of confusion matrix and putting those values for the equations of precision, recall, F1-score and support, we created Table 4 as a classification report and incorporated our results for different classes healthy, red rot and red rust mentioned in Table 4.

When training was done for ten numbers of epochs, proposed model achieved the validation accuracy of 97.78% in the 10th epoch which can be seen from Figure 4 as mentioned in Figure 4. As we can see from Figure 4 which represents the accuracy graph and Figure 5 which represents the loss graph, from the accuracy graph we can see that

graph rises very quickly at initial epochs for both training and validation which represents that our model learns quickly. Dense net has dense connections which propagates gradients very quickly across the layers which help in quick learning of our model, which can be seen from loss graph also which shows that both training and validation loss decreases quickly at initial epochs which also suggest quick learning of our model. If we see the comparison graph which is shown in Figure 6, we can see that the slope of proposed model's graph is higher than all other models for initial epochs which suggest that proposed model is learning quickly than all other models mentioned in the graph. Moreover, this model provides computational efficiency as well as parameters efficiency. Then unseen images were fed to our trained model for the detection of sugar cane plant disease. Figure 3 displays the identification result of the proposed procedure. Most identified sugar cane disease type has a high true positive rate and lower rate of false positive. For example, for 15 numbers of healthy leaves, 14 were predicted as correctly which can be called as true positive and 1 was predicted incorrectly by the model which gives 94% as the precision for the healthy category and 100% precision for other two categories that is red hot and red rust. Computing the average, we get as a 98% over the three different categories. These values can be referred from Table 4 which contains the model classification report where, parameters like precision, recall, F1-score, and support is mentioned for all the categories of sugarcane disease.

Table 4 Model classification report

	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Support</i>
Healthy	1.00	0.93	0.97	15
Red rot	0.94	1.00	0.97	15
Red rust	1.00	1.00	1.00	15
Accuracy	-	-	0.98	45
Macro average	0.98	0.98	0.98	45
Weighted average	0.98	0.98	0.98	45

5.4 Comparative analysis

The graph plotted in Figure 6, represents the accuracy comparison between the proposed model which is based on DenseNet-201 architecture and SVM categoriser, and other existing techniques. Red coloured curve in the line graph represents accuracy graph of proposed model, green coloured curve in the line graph represents model based on dense net 121 (Vellaichamy et al., 2021), blue coloured curve in the line graph represents model based on DLQP technique followed by SVM categoriser (Ahmad et al., 2020), yellow coloured curve represents model based on LeNet architecture (Tm et al., 2018), this architecture was created for particular use cases and this architecture does not perform well with the colour images and most of the day to day categorisation of images are done on colour images only that is RGB images. Since this architecture does not have much depth, this architecture finds it difficult to take out all the key points and thus decreases the performance of the model. From the comparison graph, we can see its performance is lower than the proposed model's performance. A comparative study in the field of crop leaf illness identification was conducted and represented in Table 5.

Figure 4 Accuracy vs. epochs (see online version for colours)

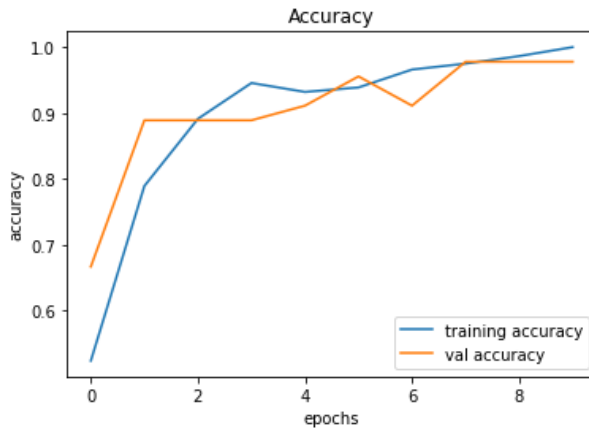


Figure 5 Loss vs. epochs (see online version for colours)

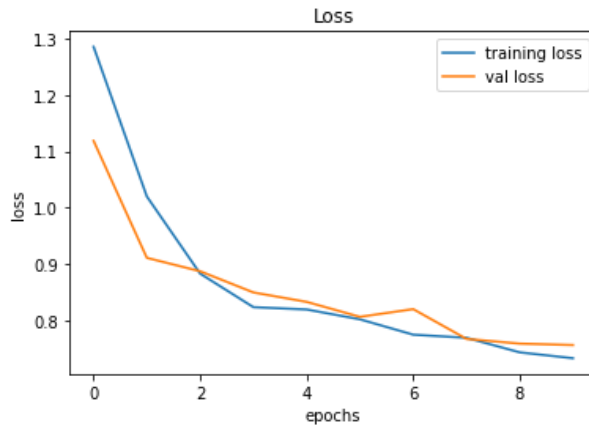


Figure 6 Comparison with existing techniques (see online version for colours)

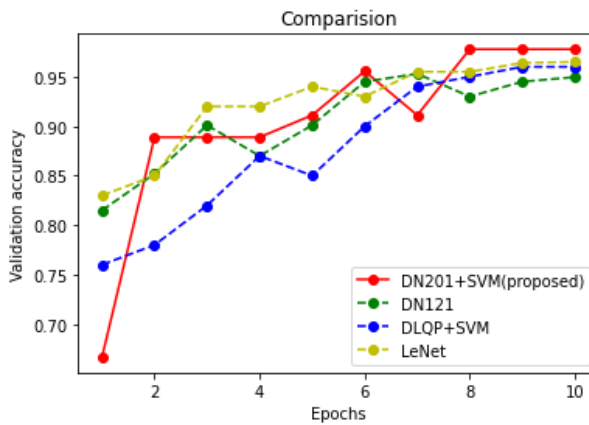


Table 5 shows the comparative study of the proposed models and few of the existing models like LeNet, DenseNet-121, and KNN-based and DLQP with SVM-based models. The relative advantage of all these models and proposed model is represented in Table 5 along with their accuracy. The provided comparative study also guides us in selecting the model based on the different situations and depending upon the dataset used for categorisation.

Table 5 Performance comparison with existing techniques

<i>Method</i>	<i>Accuracy</i>	<i>Advantage</i>
For the purpose of classifying the numerous plant diseases, the DLQP technique with the SVM categoriser was used (Ahmad et al., 2020)	96.53%	This technique is reliable for classifying plant leaf diseases despite significant angle and scale changes in input data
Plant disease categorisation utilised a deep Siamese network and a KNN categoriser (Goncharov et al., 2020)	96%	The method has increased the accuracy of categorisation for samples with complicated backgrounds
LeNet, DL-oriented architecture was used to take out the critical information and divide the data into classes for healthy and ill individuals (Tm et al., 2018)	94.80%	Less training data is required
For classifying crop leaf illness DenseNet-121 architecture was used (Vellaichamy et al., 2021)	94.96%	Provides good flow of gradient
A DenseNet-201 network together with SVM-based classifier for plant leaf disease identification (proposed)	97.78%	Encourages feature reuse and it reduces parameters
Comparison analysis		

6 Conclusions and future scope

This allowed us to create a plant disease detector with a valid accuracy of 97.7% and a training accuracy of 97.78%. Because of its ‘.tflite’ (TensorFlow Lite) variation, we can install this incredibly accurate model on mobile devices. We would suggest natural and eco-friendly treatments to lessen the harm caused by crops, such as water contamination caused using insecticides and pesticides.

These days, it is essential to spot a disease in a crop when it is still budding to boost yield and the quality of the harvest. Since disease detection requires a great deal of skill, it will be worthwhile if this system could be added on smartphone so that farmers could take an image of a leaf and upload it to the server. The server will automatically recognise and categories the kind of illness, and it will then send the reports and any recommended medications back to the user of the system.

References

- Abdollahi, A. and Pradhan, B. (2021) 'Urban vegetation mapping from aerial imagery using explainable AI (XAI)', *Sensors*, Vol. 21, p.4738 [online] <https://doi.org/10.3390/s21144738>.
- Agarwal, M., Singh, A., Arjaria, S., Sinha, A. and Gupta, S. (2020) 'ToLeD: tomato leaf disease detection using convolution neural network', *Procedia Computer Science*, Vol. 167, pp.293–301 [online] <https://doi.org/10.1016/j.procs.2020.03.225>.
- Ahmad, W., Shah, S.M.A. and Irtaza, A. (2020) 'Plants disease phenotyping using quinary patterns as texture descriptor', *KSII Transactions on Internet and Information Systems*, Vol. 14, No. 8, pp.3312–3327, DOI: 10.3837/tiis.2020.08.009.
- Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M.G., Bereciartua, A. and Alvarez-Gila, A. (2020) 'Few-shot learning approach for plant disease classification using images taken in the field', *Computers and Electronics in Agriculture*, Vol. 175, p.105542 [online] <https://doi.org/10.1016/j.compag.2020.105542>.
- Batool, A., Hyder, S.B., Rahim, A., Waheed, N., Asghar, M.A. and Fawad (2020) 'Classification and identification of tomato leaf disease using deep neural network', *2020 International Conference on Engineering and Emerging Technologies (ICEET)*, Lahore, Pakistan, pp.1–6, DOI: 10.1109/ICEET48479.2020.9048207.
- Birgé, L. and Massart, P. (2001) 'Gaussian model selection', *Journal of the European Mathematical Society*, Vol. 3, pp.203–268 [online] <https://doi.org/10.1007/s100970100031>.
- Bruinsma, J. (2009) 'The resource outlook to 2050: by how much do land, water and crop yields need to increase by 2050', in *Expert Meeting on How to Feed the World in 2050* [online] https://www.fsnnetwork.org/sites/default/files/the_resource_outlook_to_2050by_how_much_do_land_water_and_crop_yields_need_to_increase_by_2050_.pdf.
- Dinh, H.X., Singh, D., Periyannan, S., Park, R.F. and Pourkheirandish, M. (2020) 'Molecular genetics of leaf rust resistance in wheat and barley', *Theor. Appl. Genet.*, Vol. 133, pp.2035–2050 [online] <https://doi.org/10.1007/s00122-020-03570-8>.
- Ferentinos, K.P. (2018) 'Deep learning models for plant disease detection and diagnosis', *Computers and Electronics in Agriculture*, Vol. 145, pp.311–318 [online] <https://doi.org/10.1016/j.compag.2018.01.009>.
- Goncharov, P., Uzhinskiy, A., Ososkov, G., Nechaevskiy, A. and Zudikhina, J. (2020) 'Deep siamese networks for plant disease detection', *EPJ Web of Conferences*, DOI: 10.1051/epjconf/202022603010.
- Hinton, G.E.J.S. (2009) 'Deep belief networks', *Scholarpedia*, Vol. 4, No. 5, p.5947.
- Huang, G., Liu, Z., van der Maaten, L. and Weinberger, K.Q. (2016) 'Densely connected convolutional networks' [online] <https://doi.org/10.48550/arXiv.1608.06993>.
- Joachims, T. (1998) *Making Large-Scale SVM Learning Practical*, Technical Reports, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, TU Dortmund [online] <http://hdl.handle.net/10419/77178>.
- Karthik, R., Hariharan, M., Anand, S., Mathikshara, P., Johnson, A. and Menaka, R. (2020) 'Attention embedded residual CNN for disease detection in tomato leaves', *Applied Soft Computing*, Vol. 86 [online] <https://doi.org/10.1016/j.asoc.2019.105933>.
- Kuricheti, G. and Supriya, P. (2019) 'Computer vision based turmeric leaf disease detection and classification: a step to smart agri-culture', in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE.
- Le, V.N.T., Ahderom, S., Apopei, B. and Alameh, K. (2020) 'A novel method for detecting morphologically similar crops and weeds based on the combination of contour masks and filtered local binary pattern operators', *Gigascience*, 1 March, Vol. 9, No. 3, p.giaa017, DOI: 10.1093/gigascience/giaa017.
- Liao, V. and Vemuri, R. (2002) 'Use of K-nearest neighbor classifier for intrusion detection', *11 An Earlier Version of this Paper is to Appear in the Proceedings of the 11th USENIX Security*

- Symposium, Computers & Security*, San Francisco, CA, August, Vol. 21, No. 5, pp.439–448 [online] [https://doi.org/10.1016/S0167-4048\(02\)00514-X](https://doi.org/10.1016/S0167-4048(02)00514-X).
- Liu, M., Fu, B., Xie, S., He, H., Lan, F., Li, Y., Lou, P. and Fan, D. (2021) ‘Comparison of multi-source satellite images for classifying marsh vegetation using DeepLabV3 plus deep learning algorithm’, *Ecological Indicators*, Vol. 125 [online] <https://doi.org/10.1016/j.ecolind.2021.107562>.
- Naik, J.K. and Soni, A. (2020) ‘Video classification using 3D convolutional neural network’, *Advancements in Security and Privacy Initiatives for Multimedia Images*, IGI Global Publishers, Chapter, pp.1–18, DOI: <https://10.4018/978-1-7998-2795-5.ch001>.
- Naik, J.K., Chandra, S. and Agarwal, P. (2021) ‘Dynamic workflow scheduling in the cloud using a neural network-based multi-objective evolutionary algorithm’, *Int. J. Communication Networks and Distributed Systems*, Vol. 27, No. 4, pp.424–451, DOI: 10.1504/IJCND.2021.10040231.
- Naik, J.K., Pedagandham, M. and Mishra, A. (2021) ‘Workflow scheduling optimization for distributed environment using artificial neural networks and reinforcement learning (WiSo_ANRL)’, *International Journal of Computational Science and Engineering (IJCSE)*, Vol. 24, No. 6, pp.653–670, DOI: 10.1504/IJCSE.2021.10041146.
- Oo, Y.M. and Htun, N.C. (2018) ‘Plant leaf disease detection and classification using image processing’, *International Journal of Research and Engineering*, DOI: 10.21276/IJRE.2018.5.9.4.
- Ouiminga, I. (2022) ‘The effects of adopting sustainable farming practices on smallholders’, *International Journal of Agricultural Resources, Governance and Ecology*, Vol. 18, Nos. 1/2, pp.55–76, DOI: 10.1504/IJARGE.2022.10048474.
- Pantazi, X.E., Moshou, D. and Tamouridou, A.A. (2019) ‘Automated leaf disease detection in different crop species through image features analysis and one class classifiers’, *Compute Electron Agric.*, January, Vol. 156, No. 1, pp.96–104, <https://doi.org/10.1016/j.compag.2018.11.005>.
- Plant-Village [online] https://drive.google.com/drive/folders/1EbVaGV072ltejUDOEc1qoTl1hGtQ89xL?usp=share_link (accessed 23 November-2022).
- Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Bhat, N.P., Shashank, N. and Vinod, P.V. (2018) ‘Plant disease detection using machine learning’, *2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C)*, Bangalore, India, pp.41–45, DOI: 10.1109/ICDI3C.2018.00017.
- Richey, B., Majumder, S., Shirvaikar, M. and Kehtarnavaz, N. (2020) ‘Real-time detection of maize crop disease via a deep learning-based smartphone app’, *Proceedings of the SPIE*, Vol. 11401, ID 114010A, 7pp, DOI:10.1117/12.2557317.
- Rokach, L. and Maimon, O. (2005) ‘Decision trees’, in Maimon, O. and Rokach, L. (Eds.): *Data Mining and Knowledge Discovery Handbook*, Springer, Boston, MA, pp.165–192 [online] https://doi.org/10.1007/0-387-25465-X_9.
- Roska, T. and Chua, L.O. (1993) ‘The CNN universal machine: an analogic array computer’, in *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, March, Vol. 40, No. 3, pp.163–173, DOI: 10.1109/82.222815.
- Salakhutdinov, R. and Hinton, G. (2009) ‘Deep Boltzmann machines’, in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Clearwater Beach, Florida, USA, Vol. 5 of JMLR: W&CP 5.
- Sembiring, A. et al. (2021) ‘Development of concise convolutional neural network for tomato plant disease classification based on leaf images’, in *Journal of Physics: Conference Series*, IOP Publishing, DOI: 10.1088/1742-6596/1845/1/012009.
- Shaheen, M., Soma, M.K., Zeba, F. and Aruna, M. (2020) ‘Precision agriculture in India – challenges and opportunities’, *International Journal of Agricultural Resources, Governance and Ecology*, Vol. 16, Nos. 3/4, pp.223–246, DOI: 10.1504/IJARGE.2020.10038089.

- Sun, Y., Jiang, Z., Zhang, L., Dong, W. and Rao, Y. (2019) 'SLIC_SVM based leaf diseases saliency map extraction of tea plant', *Computers and Electronics in Agriculture*, Vol. 157, pp.102–109 [online] <https://doi.org/10.1016/j.compag.2018.05.053>.
- Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N.B. and Koolagudi, S.G. (2018) 'Tomato leaf disease detection using convolutional neural networks', *2018 Eleventh International Conference on Contemporary Computing (IC3)*, Noida, India, pp.1–5, DOI: 10.1109/IC3.2018.8530532.
- Turkoglu, M., Yanikoglu, B.A. and Hanbay, D. (2021) 'PlantDiseaseNet: convolutional neural network ensemble for plant disease and pest detection', *Signal, Image and Video Processing*, Vol. 16, pp.301–309, DOI: 10.1007/s11760-021-01909-2.
- Vellaichamy, A., Swaminathan, A., Varun, C. and Kalaivani, S. (2021) 'Multiple plant leaf disease classification using DenseNet-121 architecture', *International Journal of Electrical Engineering and Technology*, Vol. 12, No. 5, pp.38–57, DOI: 10.34218/IJEET.12.5.2021.005.
- Xu, L., Wang, X., Bai, L., Xiao, J., Liu, Q., Chen, E., Jiang, X. and Luo, B. (2020) 'Probabilistic SVM classifier ensemble selection based on GMDH-type neural network', *Pattern Recognition*, October, Vol. 106, No. 10, p.107373, <https://doi.org/10.1016/j.patcog.2020.107373>.
- Zaremba, W., Sutskever, I. and Vinyals, O. (2014) *Recurrent Neural Network Regularization*, arXiv preprint arXiv [online] <https://doi.org/10.48550/arXiv.1409.2329>.
- Zhang, Y., Song, C. and Zhang, D. (2020) 'Deep learning-based object detection improvement for tomato disease', in *IEEE Access*, Vol. 8, pp.56607–56614, DOI: 10.1109/ACCESS.2020.2982456.
- Zou, Q., Ni, L., Zhang, T. and Wang, Q. (2015) 'Deep learning based feature selection for remote sensing scene classification', *IEEE Geoscience and Remote Sensing Letters*, Vol. 12, No. 11, pp.2321–2325.