

A deep learning-based approach for early detection of disease in sugarcane plants: an explainable artificial intelligence model

Rubini Pudupet Ethiraj, Kavitha Paranjothi

Department of Computer Science and Engineering, CMR Institute of Technology, Visvesvaraya Technological University, Bengaluru, India

Article Info

Article history:

Received May 25, 2023

Revised Sep 13, 2023

Accepted Sep 25, 2023

Keywords:

Agnostic explanation

agriculture

DenseNet

Disease detection

Local interpretable model

Support vector machine

ABSTRACT

In many regions of the nation, agriculture serves as the primary industry. The farming environment now faces a number of challenges to farmers. One of the major concerns, and the focus of this research, is disease prediction. A methodology is suggested to automate a process for identifying disease in plant growth and warning farmers in advance so they can take appropriate action. Disease in crop plants has an impact on agricultural production. In this work, a novel DenseNet-support vector machine: explainable artificial intelligence (DNet-SVM: XAI) interpretation that combines a DenseNet with support vector machine (SVM) and local interpretable model-agnostic explanation (LIME) interpretation has been proposed. DNet-SVM: XAI was created by a series of modifications to DenseNet201, including the addition of a support vector machine (SVM) classifier. Prior to using SVM to identify if an image is healthy or un-healthy, images are first feature extracted using a convolution network called DenseNet. In addition to offering a likely explanation for the prediction, the reasoning is carried out utilizing the visual cue produced by the LIME. In light of this, the proposed approach, when paired with its determined interpretability and precision, may successfully assist farmers in the detection of infected plants and recommendation of pesticide for the identified disease.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Rubini Pudupet Ethiraj

Department of Computer Science and Engineering, CMR Institute of Technology

Visvesvaraya Technological University

Sri Nivasa Reddy Layout, AECS Layout, Marathahalli, Bengaluru, Karnataka 560037, India

Email: rubinipe@gmail.com

1. INTRODUCTION

The primary sector that needs to be prioritized for economic expansion is agriculture. Farmers in agricultural areas deal with a variety of problems, including erratic weather patterns, the detection of pests and disease, improper harvesting times, and a lack of labor. Disease prediction is one of the major themes that needs study attention since it is a difficult activity that, if not effectively managed, reduces yield [1]. The sugarcane crop was chosen for this effort because it is the second most important manufacturing crop in the country, with approximately five million hectares under cultivation. The crop is raised in India under a variety of climatic circumstances, which have a considerable impact on the production. Red rot, smut, wilt, and yellow leaf disease, which is pervasive throughout the nation and poses a serious threat to the crop, must be dealt with quickly and properly. Years of sugarcane research have built a solid technological foundation that makes these accomplishments possible.

Traditional disease identification involves farmers spotting the disease with their unaided eyes, however diagnosing the disease takes a lot of expertise and work. Most importantly, disease detection should

occur early to prevent harm and loss of productivity. However, it might be time-consuming to manually detect disease in large farms. Sugarcane is an expensive crop, and consulting experts is awfully expensive.

Therefore, to address the issue of disease detection, a deep learning model is employed. This model enables us to identify the diseases quickly and reliably in sugarcane leaf and stem by identifying patterns in the images. As far as we are aware, the pretrained models of convolutional neural network (CNN) such as ResNet-50, visual geometry group (VGG)-16, AlexNet, and inception V4 state-of-the-art deep learning architectures displaying the most effective results in plant disease detection. The vanishing gradient issue affects AlexNet, Inception, and VGG-16. The derivative's value decreases virtually exponentially as we deepen the CNN and back-propagate to the base layers. The ResNet architecture, on the other hand, uses short-cut connections to address the vanishing gradient issue however it is computationally expensive. Therefore, there is a need for a structure that equalizes the disparities between the two i.e., the model should have a simple and practical architecture, be computationally cheap, have a smaller number of layers and specifications, and attain equivalent performance. This work introduces a DenseNet-support vector machine called "DNet-SVM" which utilizes only up to three dense block to avoid the computational issue and replaces the softmax classifier with SVM for better performance in identifying diseased images. It creates additional feature maps and makes reuse of characteristics easier at each succeeding layer inside a dense block. In order to route residuals during backpropagation, it also addresses the vanishing gradient problem utilizing an obvious route to all prior layers. Also, DenseNet outperformed other cutting-edge CNN architectures in the ImageNet competition.

Additionally, the widespread implementation of a model known as local interpretable model-agnostic explanations (LIME) is the research's most important contribution which provides an explainable solution regardless of neural network's classification, for which the developer could not fully comprehend the reasons for classification as it is completely a black box. LIME provides the justification of why it is classified as a healthy or diseased image. Around 14,000 photos of the leaves and stems of sugarcane are included in the dataset utilized for the proposed research. There are 6 classifications in the dataset (5 types of sugarcane diseases and a healthy class of sugarcane). The dataset is created by photographing the sugarcane field to test the model. Each of the group include entirely unique photos, for epitome, the validation dataset will not contain any of the images from the training dataset. Transfer learning is used to compare the outcomes after the DNet-SVM model has been trained. Transfer learning places an emphasis on remembering the knowledge acquired while addressing one issue and utilizing it to another, similar problem. VGG-16, VGG-19, inception, and ResNet are the model architectures that were utilized to compare the results. The following is indicated as to the importance of this work:

- For precise plant disease diagnosis, a unique model is contrived.
- A description of how to justify the categorization using LIME.
- Support for farmers to diagnose the disease in the early stages and increases yield.
- Comparison of several transfer learning models based on accuracy, specificity, and sensitivity.
- A suitable pesticide recommendation for the noted sugarcane disease.

Following are definitions for the remaining sections of the paper: the second section focuses on the relevant work carried out by various researchers in the related subject. The proposed architecture and LIME model, which explains an understandable solution, are described in section 3. Section 4 provides the detailed results and discussion and section 5 confers the conclusion.

2. RELATED WORK

In this section, we will describe the current common transfer learning algorithm and techniques which are used by researchers in their research work. Knowledge transfer between tasks is the fundamental principle of transfer learning. The skills learned while mastering a specific task can be used for the one at hand [2]. The techniques of transfer learning are the representation of inductive learning, which is to infer a mapping from a series of training instances where the model learns a mapping between input feature and class labels. These techniques can be utilized in the context of deep learning. Using a network that has already been trained on alternate subject matter for a different source task and adapting it to the suitable field and target task saves time compared to having to train a deep network from scratch. In the domain of deep learning, there are numerous pre-trained models available, including VGG-16 [3], VGG-19, InceptionNet [4], ResNet [5], and DenseNet [6]. In this research, a few of pre-trained models were put into practice, and their accuracies and other metrics were assessed for comparison with the proposed architecture model.

To assist with disease prediction in agriculture, Pallagani *et al.* [7] integrate deep learning methods and computer vision technologies. A CNN model is used on a dataset of 54,306 publically accessible photos. The trained model provides a precision of 99.24%. As a tangible output of this research, dCrop, a smartphone application created utilizing the trained disease prediction model, is made available. The farmer can demonstrate the viability of the solution by using the app to capture crop photos and assess the presence or

absence of disease. Militante *et al.* [8] suggested a method for locating diseases in sugarcane plants. A convolution neural network model is trained using 13,842 pictures of healthy and infected sugarcane leaves. The model demonstrated 95% accuracy in identifying and categorizing pathogens of sugarcane [9]. A method to identify disease in coconut trees was suggested by Piyush Singh and Colleagues. Image processing and deep learning technologies are utilized to forecast diseases and pest infections. The manually constructed CNN model's validation accuracy was 96.44%, with a Kappa value of 0.91. To automatically detect the coconut tree disease, the flask micro-web framework was used to install the MobileNet model and a modified 2D-CNN model. The paper [10] with the aid of an NVIDIA tegra system on chip (SoC), a camera-equipped drone, and a deep learning algorithm, Chandy proposed a precision agricultural technique to detect numerous pests in coconut trees and identify the sick and pest-affected trees. Wi-Fi is used to transport the data immediately to the farmer's smartphone.

This aids in the prompt treatment of pest-infested trees and increases tree production. Ribeiro *et al.* [11] explains the interpretability of the model and proposed a method to demonstrate the predictions of the model and its justifications. The researchers have used random forest and neural network model to showcase the predictions and its explanations by implementation of LIME model. Al-Amin *et al.* [12] put forth a CNN model for the prediction of disease from the leaves of potato plants. They have used a plant village dataset to identify 2 categories of disease and a healthy plant. Hasan *et al.* [13] devised a CNN model for the recognition of jute disease. The authors have selected two types of diseases chlorosis and yellow mosaic. They used a dataset of 600 images to classify two common jute leaf diseases and used the evaluation metrics of accuracy, sensitivity, specificity, precision, and GMean. Tenorio *et al.* [14] presented three computer vision techniques. For the detection of pests and maintenance of pest-free plants, deep CNN (DCNN), hierarchical deep CNN (HD-CNN), and pixel-wise segmentation networks are used. The algorithm has been put into practice for the plant's cotton and soybean production. The comparison results of the three algorithms proved SegNet is outperforming with an accuracy of 93.80% while DCNN provides 70.14% and HD-CNN provides 74.70%. Brahimi *et al.* [15] implemented the pretrained model of AlexNet and GoogleNet for the classification of disease in tomato plants. The accuracy of these models is compared with the machine learning algorithm of support vector machine and random forest. AlexNet and GoogleNet provides the better performance than the traditional algorithm. Shiji *et al.* [16] proposed VGG-16 architectural model for the classification of disease in tomato plants. The pretrained model of VGG-16 fully connected layer is used for feature extraction and the extracted features are passed to support vector machine for the classification. The hybrid model of VGG-16 with SVM provides the accuracy of 88%. Amara *et al.* [17] and the co author of this research suggested a deep learning-based method for automating the illness classification of banana leaves. In order to categorize sets of visual data, the authors employed the LeNet architecture as a convolutional neural network. Durmus *et al.* [18] to identify various diseases on tomato plant leaves, the authors suggested using deep learning techniques like Alex Net and Squeeze Net. For this investigation, the robot was designed to run the deep learning algorithm in real time. So, the robot will be able to identify plant diseases while it is travelling manually or automatically on the field or in the greenhouse [19]. The authors talked about the method for identifying strawberry disease. The farmer uploads a mobile device image of a strawberry leaf or fruit to a data-driven engine system with many convolutional and fully connected networks in order to identify the infected strawberry fruits. Up to 92% of classifications made by the model are accurate. To recognize crop leaf diseases, Khamparia *et al.* [20] presented a hybrid strategy known as a convolutional encoder network. For their implementation, they have considered three types of crops and five different agricultural diseases.

The suggested method performs effectively when given a leaf image as the input to diagnose crop disease. In this article [21], the authors conducted a survey of 40 research projects using deep learning techniques to address diverse problems in agriculture and food production. The variations in classification or regression performance between deep learning and other popular existing techniques were also examined. This study also made the claim that deep learning produces outcomes with higher accuracy than currently used image processing methods. The authors in the article [22] implemented pre-trained convolutional neural networks models like DenseNet-121, ResNet-50, VGG-16, and Inception V4 for accurate disease classification using PlantVillage dataset. The tests showed that DenseNet-121 outperformed cutting-edge models by 99.81% in terms of classification accuracy. The convolutional neural network EfficientNet was applied in the article [23] to 18,161 plain and segmented tomato leaf pictures by the authors in order to detect tomato diseases. U-net and modified U-net, two segmentation models, were employed. The article [24] suggests a deep learning-based model named plant disease detector. The model can recognize several plant diseases from photos of leaves.

3. METHOD

Data preparation, feature extraction, classification, and LIME interpretation are the four sections that make up the proposed methodology. The suggested strategy is shown in Figure 1. The initial step is the data collection. The data was gathered from the plant village database and sugarcane farms. Images of sugarcane's leaves and stems were the focus of the proposed research. Images from both healthy and diseased plants are divided into six groups in the dataset.

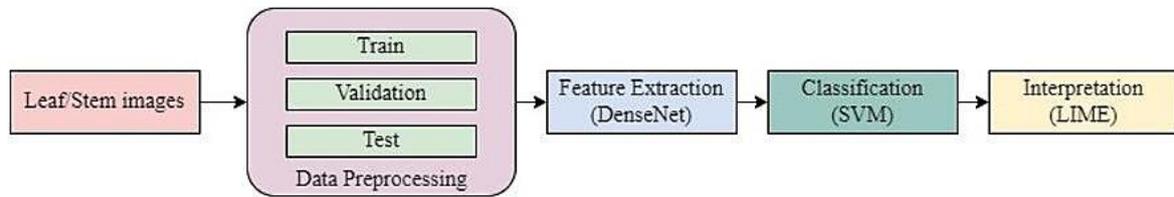


Figure 1. Topology of the proposed approach

Five different categories-bacterial blight/red stripe, wilt, red rot, red rust, and sett rot are used to categorize the diseased images [25]. Prior to being fed into the model, the image data from the database had been pre-processed and to determine a standard size for every image, and get rid of noise, the frames are normalized and rebuilt. Every image is a 3D vector with X, Y, and Z dimensions. Z represents the red, green, and blue (RGB) channel number, whereas X and Y represent the image's width and height. The photos are shrunk to 224×224 pixels so that they can be used with transfer learning models. In the second stage, features from the image are extracted using the DenseNet architecture. The SVM classifier uses the feature vectors to identify whether the image is healthy or diseased. The final step is to justify whether the output is expected to be healthy or unhealthy.

The proposed model uses DenseNet201 as a fundamental building element with significant personalization. Due to some crucial insights, DenseNet is best suited for classification problems since it encourages reusing features at each successive level inside a network and creates more representations of features. Even though it is stated that "deeper networks yield higher accuracy," CNN's neural network is quite deep. However, because of the vanishing gradient, it is very difficult to train the model. A dense network that connects all layers directly to one another can stop this trend. A conventional network with N layers will have N connections, however DenseNet will have $N(N+1)/2$ connections if there are N layers in the network. DenseNet is made up of transition and dense blocks. The layers are closely interwoven in dense blocks. The output feature maps from the preceding layer are used as input for subsequent layers in the dense block.

The output feature maps from earlier layers serve as the input for each layer in the dense block. If there are 10 layers, for instance, and each layer outputs a feature map of a variable size, each layer will receive more supervision from the one before it, which will result in a loss function, causing the DenseNet model to fix the feature map of each layer. The dense block consists of batch normalization, a 3x3 convolution, and a ReLu activation. All the feature maps are concatenated in the transition layer, which also serves as a max pooling layer to minimize the dimensionality.

In the Figure 2, a dense block connects the input layer. Three layers make up the dense block. Initially, the input layer transmits the feature map to conv1. Then conv2 receives the input layer and the convolution layer 1 feature representation. The input layer, conv2, and the feature map from conv1 are likewise sent to the third conv layer, just like the second conv layer. After receiving the dense block's output, the transition layer acts as the max pooling. Finally, the transition layer receives the feature representations from the input layer, conv3, conv2, and conv1, concatenates it, and reduces the feature map's dimensionality.

Loss function is avoided by fixing the results of each layer's feature representation. The conv layer of each dense block has layers for convolution, batch normalization, and rectified linear unit (ReLu). Growth factors are used to describe the number of feature maps in each layer. Prior to sending the input to the dense block, a first convolution layer produces an output with a 112×112 dimension. The output size is then reduced to 56×56 by a pooling layer. DenseNet's main benefit is that:

- It keeps track of the path from the input layer to the output layer.
- In comparison to other CNN models, DenseNet needs less parameters.
- Solves the gradient problem. Each layer has direct access to the gradient from the input image and loss function. The input has k feature mappings in the DenseNet architecture. A feature map is produced by the convolution layer, which is the first layer.

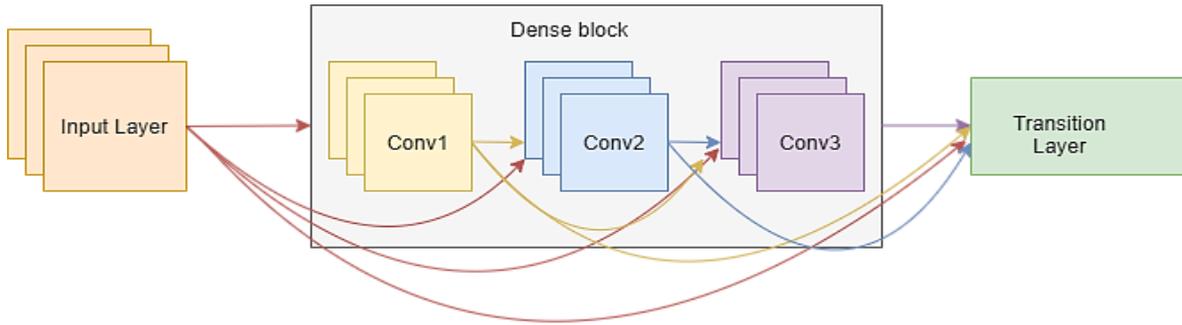


Figure 2. DenseNet architecture

The suggested model's schematic design is shown in Figure 3, with DenseNet acting as a feature generator and SVM functioning as a classifier. A deep CNN model called Dense-Net201 was trained on the ImageNet dataset, which has 4 dense blocks with, respectively 6, 12, 48, and 32 dense layers. Each dense layer provides k feature maps. The model, however, is enormous and has a tonne of layers, making it challenging to train using the images in the dataset. The suggested approach requires up to three blocks for feature extraction. The flatten layer then receives the feature vectors that were extracted from the third block of the global average pooling layer. Two fully connected layers are placed for sampling reduction, and to avoid overfitting, two dropout layers with a 0.2 are added.

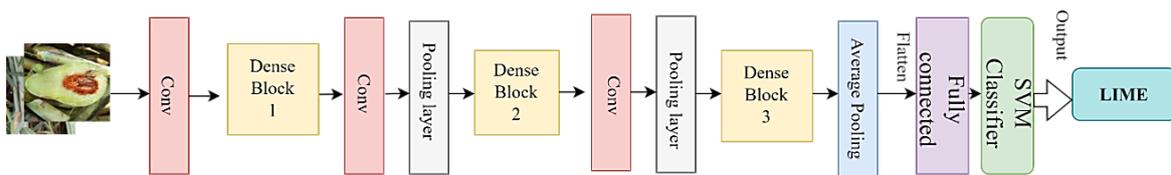


Figure 3. Schematic design of DNet-SVM

The last fully connected layer, which comprises six neurons from six different classes (bacterial blight, healthy, redrot, redrust, sett rot, and wilt), replaces the softmax classifier with an SVM classifier. Nevertheless, SVM classifier performs better in our experiments than softmax. The initial layers in our experiment are kept frozen, and the remaining layers are retrained using the dataset. The learning parameter used for the implementation is shown in Table 1.

Table 1. Learning parameters

S. No	Parameters	Value
1	Optimizer	Adam
2	Loss function	Square hinge
3	Learning rate	0.0001
4	Batch size	32
5	Epoch	50

3.1. Interpretation of the model

While most algorithmic solutions in machine learning or deep learning models are black boxes, they face the difficulty of explain ability. The end-user needs a solution that can be understood by knowing precisely how the model makes predictions. In this research, a model is created to determine whether an image is diseased or healthy. Due to their long history of using farms, it might be difficult to convince farmers of the advantages of technology. The only way to reach the farmer is through outcomes that have been demonstrated, thus it should be justifiable. The researcher makes various efforts to make machine learning models understandable; the tools SHAP, RISE, DEEP LIFT, LIME, and ELI5 are used to interpret the findings. The global surrogate model is illustrated in Figure 4, with its basic operation.

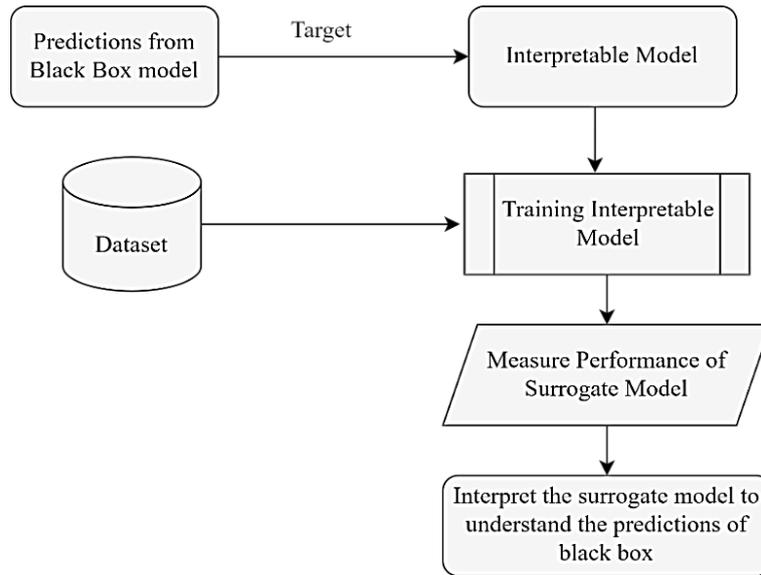


Figure 4. LIME model

3.2. Local interpretable model-agnostic explanation

Finding an understandable representation for image classification is crucial; this involves identifying the collection of pixel values that represent a healthy or diseased image. So, a set representing 0 or 1 makes up the binary vector for image classification. An image's pixels are represented by the coordinates $x \in \{0,1\}$. The explanation obtained by LIME can be expressed as in (1) for a sample case:

$$\xi(x) = \operatorname{argmin} U(p, m, Mx) + \omega(m) \quad (1)$$

Let ' x ' be a sample and the interpretation for the sample (x), is calculated by two terms $\operatorname{argmin} U(p, m, Mx)$ and $\omega(m)$. ' p ' is the probability that x belongs to a particular class, ' m ' is any interpretable model, and Mx provides the proximity measure to define the closer locality around x . Let $U(p, m, Mx)$ is a measure of how deceitful the model ' m ' is in approximating the probability value ' p ' in the proximity defined by Mx . To ensure interpretation and loyalty, the value of $U(p, m, Mx)$ is minimized. $\omega(m)$ defines the complexity of the model.

LIME explainer object (sample instance, black box model) For each of the examples:

- Choose an image (a sample case) to see a breakdown of the black box prediction;
- Create a sample dataset centered on the sample instance using the data sampled from a normal distribution;
- Give the sample that is closest to the sample instance more weight;
- Using the sample dataset and the variations, train an understandable model;
- Explain the prediction using the local model's interpretation; and
- Provide the sample instance's explanation.

3.3. Evaluation metric

The model's goodness must be evaluated after it has been developed. sensitivity, specificity, false negative rate (FNR) and false positive rate (FPR) are the metrics used to validate the model were calculated using (2)-(5). The developer can see how a classifier is performing using the area under the curve (AUC) of the receiver characteristic operator (ROC). Sensitivity and specificity are two concepts that are crucial to understand in order to comprehend the AUC-ROC curve. The confusion matrix, which is often referred to as the error matrix, can be used to derive sensitivity and specificity. It includes details on both actual and predicted classifications. True positive (TP), true negative (TN), false positive (FP), and false negative (FN) are the four outcomes that can occur in the matrix.

$$\text{Sensitivity} = \frac{TP}{(TP+FN)} \quad (2)$$

$$\text{FNR} = \frac{TP}{(TP+FN)} \quad (3)$$

$$FPR = \frac{FP}{(TN+FP)} = 1 - \text{Specificity} \quad (4)$$

$$\text{Specificity} = \frac{TN}{(TN+FP)} \quad (5)$$

In many classification algorithms, the classifier outputs the actual class, but the better part is to portray the probability of predicting the different classes, where the user can set the threshold value to interpret the output. To lower the numbers of false negative or false positive, different value of threshold will be set which inadvertently change specificity and sensitivity. Consider, the predicted probability (x) and the threshold value (y), by default the user sets the threshold value as 0.5.

if $x \geq y$, Class 0 or healthy image
else, Class 1 or diseased image

But setting a default threshold value would not be wise because different FPR, true positive rate (TPR), true negative rate (TNR), and FNR can be obtained by adjusting threshold values. The AUC-ROC curve, which compares TPR and FPR, is designed to demonstrate which threshold value yields the best outcomes. An evaluation of a classifier's ability to differentiate between classes is done using the AUC statistic, which summarizes the ROC curve. The area under the ROC curve had a value of 0.94. Analysis utilizing different thresholds reveals that the value of 0.94 delivers the best result. The AUC value increases with the model's ability to distinguish between classes.

The classifier's estimated AUC is in the range of $0.5 < \text{AUC} < 1$, which indicates that it would successfully classify the images. The point is located between the coordinates (0,1) in the plane and the top left corner of the receiver operating characteristic (ROC) graph, where both specificity and sensitivity are at the highest levels, allowing the classifier to accurately categorize the images. Precision-recall (PR) curve provides the trade-off between precision and recall for various threshold values. In the proposed approach, both PR curve and ROC curve provides the optimal threshold value as 0.94.

3.4. Performance analysis

Using various evaluation measures, the experimental outcomes of various models including VGG-16, VGG-19, ResNet, and inception are compared. DNet-SVM, the proposed model, has demonstrated greater accuracy. So, the architecture DNet-SVM is fixed to proceed with LIME interpretation. The main finding from the experimental research is that SVM outperforms softmax in terms of classification for the proposed DNet-SVM model. Both DenseNet experiments with and without SVM are carried out. DenseNet reports 94% when there is no SVM classifier, but 97% when there is. Considering the aforementioned results, it is evident that the SVM classifier outperforms softmax when given the same features. As a result, we used SVM to classify the diseased and healthy images. The comparative results of different models are shown in Table 2.

Table 2. Performance comparison of various pre-trained model with the proposed approach

Models	Sensitivity	Specificity	FNR	FPR
VGG-16	0.75	0.375	0.25	0.625
VGG-19	0.79	0.44	0.20	0.55
ResNet	0.704	0.31	0.29	0.68
Inception	0.73	0.37	0.26	0.62
DenseNet	0.91	0.75	0.08	0.25
DNet-SVM	0.94	0.86	0.05	0.13

4. RESULTS AND DISCUSSION

To evaluate the various performances of the state-of-the-art model like ResNet-50 [22], VGG-16 [16], AlexNet [15], and inception V4 [22], comparative research was conducted. Additionally, sensitivity, specificity, FPR, and FNR are used as performance metrics for model evaluation. From the findings in Table 2, it can be deduced that the proposed approach performs better than all state-of-the-art models with a minimum increase of 3% accuracy when using DNet-SVM.

The effort of the proposed work is to distinguish between healthy and unhealthy images and inform the farmer of the type of disease. Additionally, the proposed strategy will advise the proper pesticide for the diagnosed disease, providing farmers with a one-stop shop for all their needs. The validation set is used to test trained models using a GPU. 2,800 validation images and 11,200 training images are present. They are all

labelled. Classification accuracy for the DNet-SVM model was 97 percent for the count of 50 Epoch. The best performance is displayed by DNet-SVM. Our primary outcome is that the DNet-SVM is a simpler model with a desirable layer configuration, comprising minimal learning specifications and performing on par with state of the art. The proposed model accuracy and loss is shown in Figure 5. The image shown in Figure 5(a) displays the proposed model accuracy and image in Figure 5(b) shows the model loss. The test image shown in Figure 6(a) was given to the model for prediction is the one that has the disease RedRot. Figure 6 (b) displays the model's prediction findings.

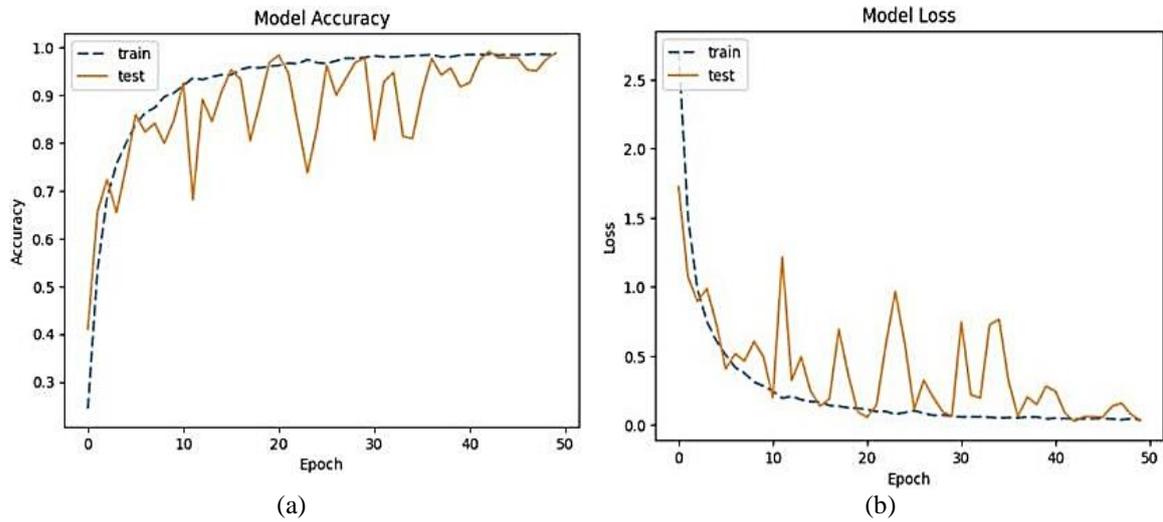


Figure 5. Accuracy and Loss of DNet-SVM (a) accuracy and (b) loss

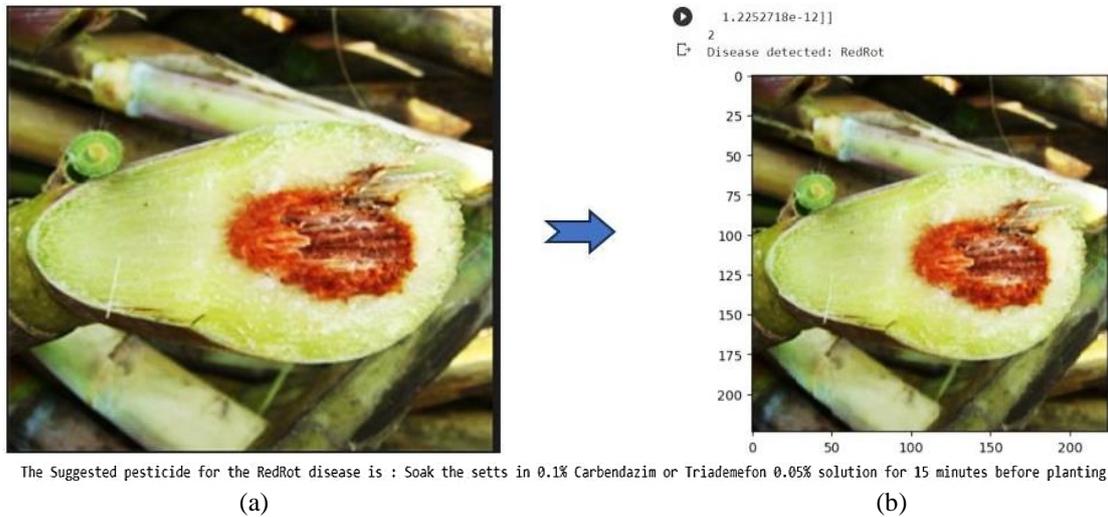


Figure 6. Sample image and output (a) test image passed to model and (b) predicted output along with suggested pesticide

Figure 7 indicates the super-pixel containing areas of the test image; the model classifies the image as diseased. The region of super-pixels colored in grey on the image are those that raise the possibility that the image is part of a diseased class, whereas the area of super-pixels colored in black lower the possibility. The results of LIME provide an explanation, the image is predicted as disease because of the identified portions It is reasonable to conclude that the proposed approach may significantly help farmers in the early diagnosis of disease given its assessed interpretability and accuracy.

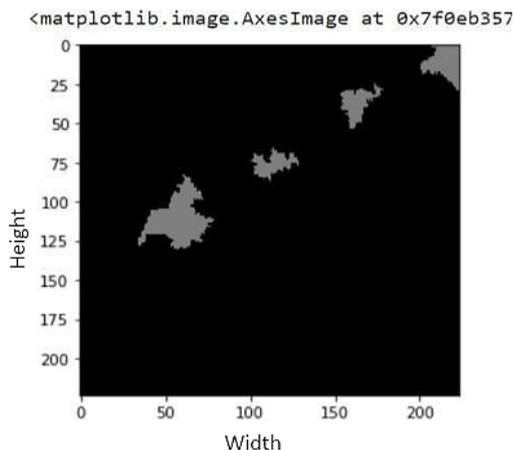


Figure 7. LIME interpretation

5. CONCLUSION

An important problem in the realm of agriculture is disease detection. Early illness diagnosis is crucial in agriculture since it affects crop output. Several methods have been suggested in the literature to deal with this issue. The traditional methods used hand-crafted elements and produced pleasing outcomes. However, the proposed model DNet-SVM has fewer layers and learning parameters which learns more essential features because of the deep connections between layers. Furthermore, for performance evaluation, extensive experiments were carried out on a database using a variety of evaluation methodologies. The qualitative findings show that the suggested model performs as well as or even better in experiments than state-of-the-art models. Additionally, the research makes an effort to develop the interpretation model, which guarantees the accuracy of the prediction made by the black-box model. Last but not least, the proposed methodology advises farmers to use the right pesticide for the identified illness as a bonus.

ACKNOWLEDGEMENTS

This study was partially supported by CMR Institute of Technology (Research Laboratory), Bangalore. In contributing this study, Rubini Pudupet Ethiraj designed the study; collected data; conducted the research; statistically analyzed and interpreted the data; draft and critical revision of the manuscript. Kavitha P supervised the research. We thank all subjects who participated in this study.

REFERENCES

- [1] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.
- [2] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," *Plant Methods*, vol. 17, no. 1, p. 22, Dec. 2021, doi: 10.1186/s13007-021-00722-9.
- [3] S. Srivastava, P. Kumar, N. Mohd, A. Singh, and F. S. Gill, "A novel deep learning framework approach for sugarcane disease detection," *SN Computer Science*, vol. 1, no. 2, p. 87, Mar. 2020, doi: 10.1007/s42979-020-0094-9.
- [4] M. A. Haque *et al.*, "Deep learning-based approach for identification of diseases of maize crop," *Scientific Reports*, vol. 12, no. 1, p. 6334, Apr. 2022, doi: 10.1038/s41598-022-10140-z.
- [5] A. A. Hernandez, J. L. Bombasi, and A. C. Lagman, "Classification of sugarcane leaf disease using deep learning algorithms," in *2022 IEEE 13th Control and System Graduate Research Colloquium (ICSGRC)*, IEEE, Jul. 2022, pp. 47–50. doi: 10.1109/ICSGRC55096.2022.9845137.
- [6] J. Boulent, S. Foucher, J. Théau, and P.-L. St-Charles, "Convolutional neural networks for the automatic identification of plant diseases," *Frontiers in Plant Science*, vol. 10, Jul. 2019, doi: 10.3389/fpls.2019.00941.
- [7] V. Pallagani, V. Khandelwal, B. Chandra, V. Udutalapally, D. Das, and S. P. Mohanty, "dCrop: a deep-learning based framework for accurate prediction of diseases of crops in smart agriculture," in *2019 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, IEEE, Dec. 2019, pp. 29–33. doi: 10.1109/iSES47678.2019.00020.
- [8] S. V. Militante, B. D. Gerardo, and R. P. Medina, "Sugarcane disease recognition using deep learning," in *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, IEEE, Oct. 2019, pp. 575–578. doi: 10.1109/ECICE47484.2019.8942690.
- [9] P. Singh, A. Verma, and J. S. R. Alex, "Disease and pest infection detection in coconut tree through deep learning techniques," *Computers and Electronics in Agriculture*, vol. 182, p. 105986, Mar. 2021, doi: 10.1016/j.compag.2021.105986.
- [10] A. Chandy, "Pest infestation identification in coconut trees using deep learning," *Journal of Artificial Intelligence and Capsule Networks*, vol. 01, no. 01, pp. 10–18, Sep. 2019, doi: 10.36548/jaicn.2019.1.002.

- [11] M. Ribeiro, S. Singh, and C. Guestrin, “‘Why should i trust you?’: explaining the predictions of any classifier,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 97–101. doi: 10.18653/v1/N16-3020.
- [12] M. Al-Amin, T. A. Bushra, and M. Nazmul Hoq, “Prediction of potato disease from leaves using deep convolution neural network towards a digital agricultural system,” in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, IEEE, May 2019, pp. 1–5. doi: 10.1109/ICASERT.2019.8934933.
- [13] M. Z. Hasan, M. S. Ahamed, A. Rakshit, and K. M. Z. Hasan, “Recognition of Jute diseases by leaf image classification using convolutional neural network,” in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, Jul. 2019, pp. 1–5. doi: 10.1109/ICCCNT45670.2019.8944907.
- [14] G. L. Tenorio *et al.*, “Comparative study of computer vision models for insect pest identification in complex backgrounds,” in *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, IEEE, Oct. 2019, pp. 551–556. doi: 10.1109/DeSE.2019.00106.
- [15] M. Brahimi, K. Boukhalfa, and A. Moussaoui, “Deep learning for tomato diseases: classification and symptoms visualization,” *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, Apr. 2017, doi: 10.1080/08839514.2017.1315516.
- [16] J. Shijie, J. Peiyi, H. Siping, and Sl. Haibo, “Automatic detection of tomato diseases and pests based on leaf images,” in *2017 Chinese Automation Congress (CAC)*, IEEE, Oct. 2017, pp. 2537–2510. doi: 10.1109/CAC.2017.8243388.
- [17] J. Amara, B. Bouaziz, and A. Algergawy, “A deep learning-based approach for banana leaf diseases classification,” *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI)*, vol. 266, pp. 79–88, 2017.
- [18] H. Durmus, E. O. Gunes, and M. Kirci, “Disease detection on the leaves of the tomato plants by using deep learning,” in *2017 6th International Conference on Agro-Geoinformatics*, IEEE, Aug. 2017, pp. 1–5. doi: 10.1109/Agro-Geoinformatics.2017.8047016.
- [19] H. Park, E. JeeSook, and S.-H. Kim, “Crops disease diagnosing using image-based deep learning mechanism,” in *2018 International Conference on Computing and Network Communications (CoCoNet)*, IEEE, Aug. 2018, pp. 23–26. doi: 10.1109/CoCoNet.2018.8476914.
- [20] A. Khamparia, G. Saini, D. Gupta, A. Khanna, S. Tiwari, and V. H. C. de Albuquerque, “Seasonal crops disease prediction and classification using deep convolutional encoder network,” *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 818–836, Feb. 2020, doi: 10.1007/s00034-019-01041-0.
- [21] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: a survey,” *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, Apr. 2018, doi: 10.1016/j.compag.2018.02.016.
- [22] A. J., J. Eunice, D. E. Popescu, M. K. Chowdary, and J. Hemanth, “Deep learning-based leaf disease detection in crops using images for agricultural applications,” *Agronomy*, vol. 12, no. 10, p. 2395, Oct. 2022, doi: 10.3390/agronomy12102395.
- [23] M. E. H. Chowdhury *et al.*, “Automatic and reliable leaf disease detection using deep learning techniques,” *AgriEngineering*, vol. 3, no. 2, pp. 294–312, May 2021, doi: 10.3390/agriengineering3020020.
- [24] R. M. J. Al-Akkam and M. S. M. Altaei, “Plants leaf diseases detection using deep learning,” *Iraqi Journal of Science*, pp. 801–816, Feb. 2022, doi: 10.24996/ij.s.2022.63.2.34.
- [25] K. Nithya, B. Parameswari, and R. Viswanathan, “Mixed infection of sugarcane yellow leaf virus and grassy shoot phytoplasma in yellow leaf affected Indian sugarcane cultivars,” *The Plant Pathology Journal*, vol. 36, no. 4, pp. 364–377, Aug. 2020, doi: 10.5423/PPI.OA.06.2020.0092.

BIOGRAPHIES OF AUTHORS



Rubini Pudupet Ethiraj    holds an engineering degree in Computer Science and Engineering in the year 2010 from SKP Engineering College, Anna University, Tamil Nadu, India. Completed Master of Engineering in the year 2012 from Sri Ramakrishna College of Engineering, Anna University, Tamil Nadu, India. Currently pursuing the Ph.D. degree in computer science and engineering from Visvesvaraya Technological University and working a Senior Project Engineer in Wipro Ltd. at Bangalore, India. She has attended many workshops and She has 20 publications which includes both national and international journal conferences with one IEEE. She can be contacted at email: rubinipe@gmail.com.



Kavitha Paranjothi    holds a Bachelor of Engineering in Computer Science and Engineering, Master of Engineering in Computer Science and Engineering, and Ph.D. in Computer Science and Engineering, besides several professional certificates and skills. She is currently working as an Associate Professor in Computer Science Engineering at CMR Institute of Technology, Bangalore, India. Her research interest includes data mining, machine learning, and wireless security. She has guided many doctoral and postgraduate researches. She is a member of IE(I) and ISTE. She can be contacted at email: kavitha.p@cmrit.ac.in or paranjothi.kavitha@gmail.com.