ISSN: 2249-7196
IJMRR/March-May 2024/ Volume 14/Issue 3/157-166
P. Sai Bhargavi / International Journal of Management Research & Review

# Detection and Classification of Plant Leaf Disease Using Local Directional Pattern and CNN

**K. V. S. Ganesh[1], G. Sandhya[2], P. Sai Bhargavi[3], G. Tanmayee Satya[4], B. Gangadhar[5]**

[1,2]Assistant Professor: Department of Electronics and Communication Engineering Avanthi Institute of Engineering and Technology, Tamaram, Makavarapalem, Anakapalle – 531113, Andhra Pradesh, India.

[3,4,5]UG Students: Department of Electronics and Communication Engineering Avanthi Institute of Engineering and Technology, Tamaram, Makavarapalem, Anakapalle – 531113, Andhra Pradesh, India.

***Abstract***

*The detection of plant leaf diseases plays a pivotal role in advancing agricultural productivity. Traditional methods that rely on human expertise are being supplemented by artificial intelligence and computer vision. This paper applies a descriptor called the Local Directional Pattern (LDP) to improve feature generation from plant leaves for disease detection in apple crops. LDP uses Kirsch edge detectors to capture the textural and edge information of a leaf image, which is what distinguishes healthy specimens from diseased ones. Because edge responses are more stable than raw intensity values, the descriptor holds its pattern under noise and non-monotonic illumination change, which is a known weakness of the Local Binary Pattern it builds on. The descriptor is benchmarked against features derived from Convolutional Neural Networks (CNN) and the Histogram of Oriented Gradients (HOG). The system was implemented in Python using TensorFlow and Keras in the Spyder environment, and was trained and tested on the Plant Village dataset. It classifies apple leaves into four categories: healthy, cedar rust, scab and black rot. The results indicate that LDP is a reliable feature extraction tool for the agricultural sector, supporting crop management and apple yield prediction, and that combining edge-based descriptors with machine learning classifiers is a direction worth further exploration.*

***Index Terms—*** *Convolutional neural network, image classification, Kirsch edge detector, local binary pattern, local directional pattern, plant leaf disease detection, texture descriptor.*

## I. Introduction

Leaf disease detection matters for the health and productivity of agricultural crops worldwide. With the global population rising and food security an increasingly pressing issue, accurate and timely identification of plant diseases is important for reducing crop losses, limiting economic impact and protecting the environment. Traditional detection depends on manual inspection by trained experts, which is labour-intensive, slow and prone to human error. Automated systems that identify diseased plants quickly and accurately allow early intervention and more effective management.

The motivation for this work comes from the limitations of existing methods. Advances in computer vision, machine learning and image processing have shown promise in automating detection, but there is still room for improvement

ISSN: 2249-7196
IJMRR/March-May 2024/ Volume 14/Issue 3/157-166
P. Sai Bhargavi / International Journal of Management Research & Review

in accuracy, efficiency and scalability. The Local Directional Pattern is used here as a feature extraction technique that addresses some of these limitations and offers better performance in speed, accuracy and robustness.

The objective is to design and implement a leaf disease detection system based on LDP that classifies images of diseased and healthy leaves with high precision and recall. Beyond detection accuracy, the system is intended to be usable, low in cost and adaptable to different plant species and environmental conditions. Giving farmers and agricultural professionals a reliable tool for early detection supports better decisions about pest and disease management, which in turn means improved yields, reduced pesticide use and a more sustainable agricultural system.

## II. Literature Survey

Oo and Htun [1] used image processing techniques to detect and classify plant leaf diseases, developing methods to analyse leaf images and identify diseases. The limitation is accuracy under real-world conditions, where lighting, background and leaf orientation vary far more than in a curated dataset.

Shruthi et al. [2] reviewed machine learning classification techniques for plant disease detection, analysing the methods used in the field and their relative accuracy and efficiency. A drawback common to the techniques reviewed is that they need large volumes of training data, and collecting and labelling such a dataset is time-consuming and costly.

El Sghair et al. [3] proposed an algorithm that detects plant diseases from colour features, identifying disease by analysing the colour characteristics of the leaf. The limitation is that it handles only diseases that show as colour change. Symptoms that appear as texture or shape abnormalities are missed.

Vishnoi et al. [4] investigated computational intelligence and image processing methods for analysing plant images to identify disease. The approach offers new ways to detect and protect plants, but the technology involved is complex, which is a barrier for farmers or agricultural workers without extensive training in the area.

Tan et al. [5] compared classical machine learning with deep learning for classifying tomato leaf diseases from leaf images. Deep learning outperformed the classical methods in accuracy, which suggests classical approaches struggle with data as complex as leaf images. This finding motivates the present work, which pairs a hand-designed but edge-based descriptor with a learned classifier rather than choosing one approach over the other.

## III. Methodology

The methodology covers image acquisition, preprocessing, feature extraction, model development, validation and deployment. The overall flow is shown in Fig. 1.

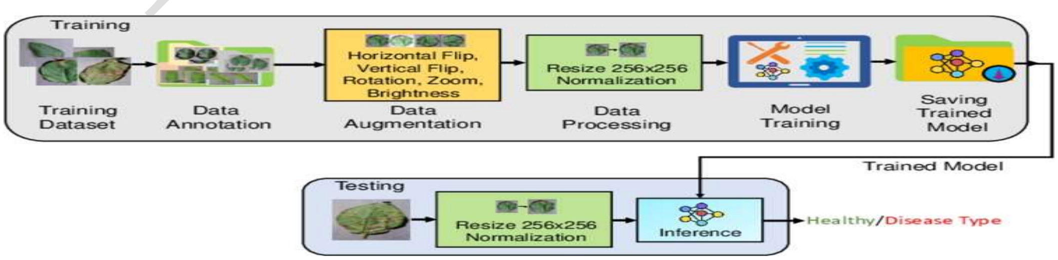


**Fig. 1. Block diagram of the proposed method.**

ISSN: 2249-7196
IJMRR/March-May 2024/ Volume 14/Issue 3/157-166
P. Sai Bhargavi / International Journal of Management Research & Review

**A. Image Acquisition**

High-quality images of plant leaves affected by various diseases are collected from research institutions, agricultural databases and field surveys. The images span a range of plant species and disease types so that the detection system generalizes rather than fitting one narrow set of conditions.

**B. Preprocessing**

Leaf images arrive from cameras or smartphones and vary in lighting, angle and background clutter, all of which introduce noise and variability. Preprocessing standardizes and improves the images before they reach the model.

Normalization rescales pixel values to a standard range, typically between 0 and 1, so that each pixel contributes equally to learning regardless of the original brightness or contrast. Cropping and resizing to a consistent size, here 256 × 256, ensures every image fed to the network has the same dimensions, which makes it easier for the network to learn patterns. Noise removal and contrast enhancement are applied to improve the clarity of leaf features so that disease symptoms are easier to identify.

Data augmentation applies transformations that artificially expand the training set. Horizontal and vertical flipping create mirror images along each axis. Rotation changes the orientation of the leaf, simulating the different angles at which a leaf would be photographed in the field. Zoom focuses on specific regions of interest within the leaf. Brightness adjustment varies the illumination level. Together these transformations teach the model to recognize the same disease under conditions it has not literally seen.

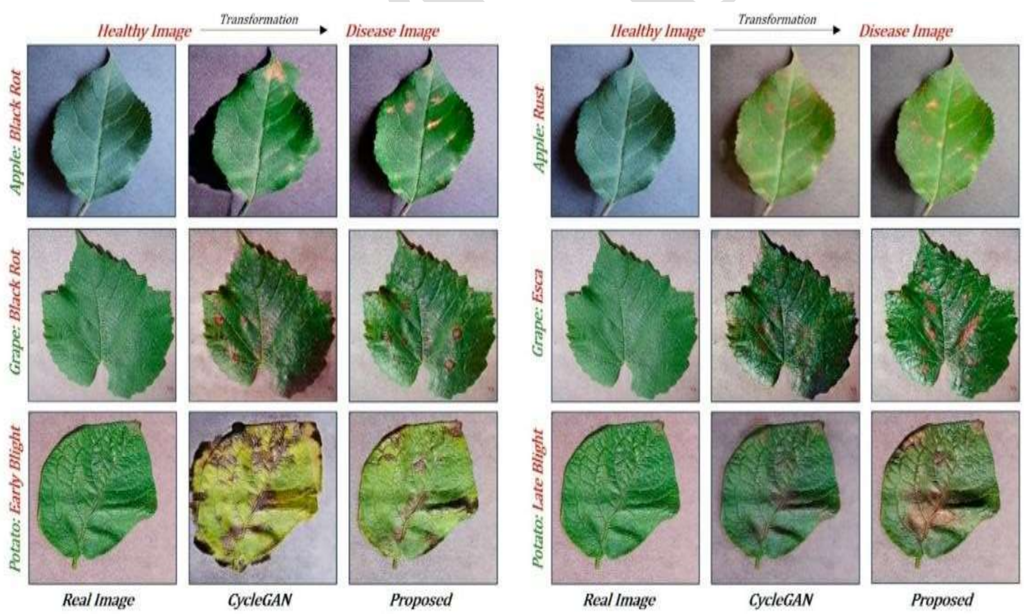


**Fig. 2. Data augmentation applied to leaf images.**

**C. Convolutional Neural Network**

A convolutional neural network is a deep learning architecture that learns directly from data and is well suited to finding patterns in images. CNNs learn spatial hierarchies of features from input images through convolutional filters, and are effective on image tasks because they capture local patterns and spatial dependencies.

A convolution layer transforms the input image by sliding a filter across it and computing a weighted sum at each position; the output is the feature map. An activation function introduces non-linearity so the network can model relationships a linear model could not. A pooling layer reduces the spatial dimensions of the feature map, cutting computation and giving a degree of translation invariance. A fully connected layer at the end of the network maps the learned features to class labels.

The VGG architecture was used as the classifier backbone. Its main idea is that several small convolution filters stacked in sequence give the same receptive field as one large filter, but with fewer parameters and more non-linearity between layers.

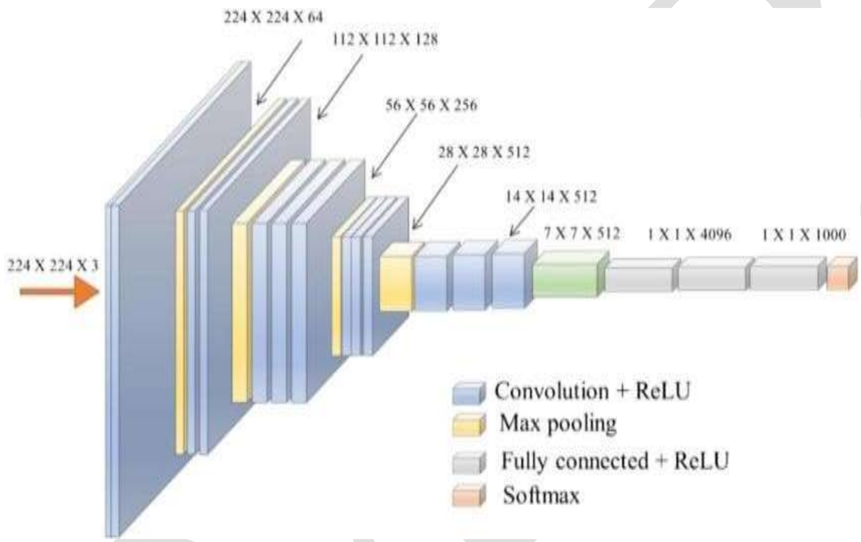


**Fig. 3. Architecture of VGG.**

**D. Local Binary Pattern**

The Local Binary Pattern is a grayscale-invariant texture measure derived from a general definition of texture in a local neighbourhood, and has performed well in facial image analysis in both speed and accuracy. The original operator labels each pixel by thresholding its 3 × 3 neighbourhood against the value of the central pixel and concatenating the results into a binary number. Its weakness is sensitivity to noise: because it compares raw intensity values, a small perturbation can flip a bit and change the code entirely.

**E. Local Directional Pattern**

Some methods encode local texture using the change of gradient magnitude in a specific direction around a pixel, comparing a neighbour's gradient magnitude along one direction and encoding it as in LBP. These cannot capture the information available from the different magnitudes of edge response in different directions, since only one direction is considered. The Local Directional Pattern addresses this by computing edge response values in all directions and using them to encode texture.

The LDP feature is an eight-bit binary code assigned to each pixel. It is calculated by comparing the relative edge response of a pixel in different directions. The Kirsch, Prewitt and Sobel detectors could all be used; the Kirsch detector was chosen because it detects directional edge responses more accurately than the others, since it considers

ISSN: 2249-7196
IJMRR/March-May 2024/ Volume 14/Issue 3/157-166
P. Sai Bhargavi / International Journal of Management Research & Review

all eight neighbours. Given a central pixel, the eight directional edge response values $m_i$, i = 0 … 7, are computed by Kirsch masks $M_i$ in eight orientations centred on that position. The masks are shown in Fig. 4.

$$\underset{\text{East } M_0}{\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}} \quad \underset{\text{North East } M_1}{\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}} \quad \underset{\text{North } M_2}{\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}} \quad \underset{\text{North West } M_3}{\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}}$$

$$\underset{\text{West } M_4}{\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}} \quad \underset{\text{South West } M_5}{\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}} \quad \underset{\text{South } M_6}{\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}} \quad \underset{\text{South East } M_7}{\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}}$$

**Fig. 4. Kirsch edge response masks in eight directions.**

The response values are not equally important in all directions. A corner or an edge produces high response values in particular directions only. The k most prominent directions are therefore the ones of interest: the top k directional bit responses $b_i$ are set to 1 and the remaining (8 − k) bits of the eight-bit pattern are set to 0. Fig. 5 shows the mask response and LDP bit positions, and Fig. 6 shows an example code with k = 3.

$$LDP_k = \sum_{i=0}^{7} b_i(m_i - m_k) \times 2^i . \qquad (1)$$

$$b_i(a) = \begin{cases} 1 & a \geq 0 \\ 0 & a < 0 \end{cases} \qquad (2)$$

where, $m_k$ is the k-th most significant directional response.

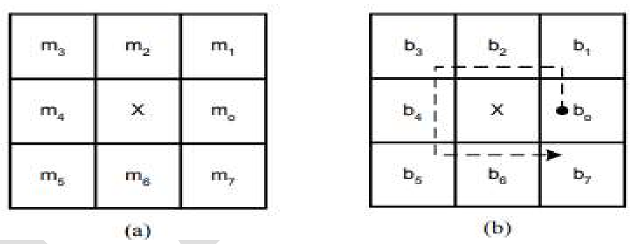


**Fig. 5. (a) Eight directional edge response positions. (b) LDP binary bit positions.**

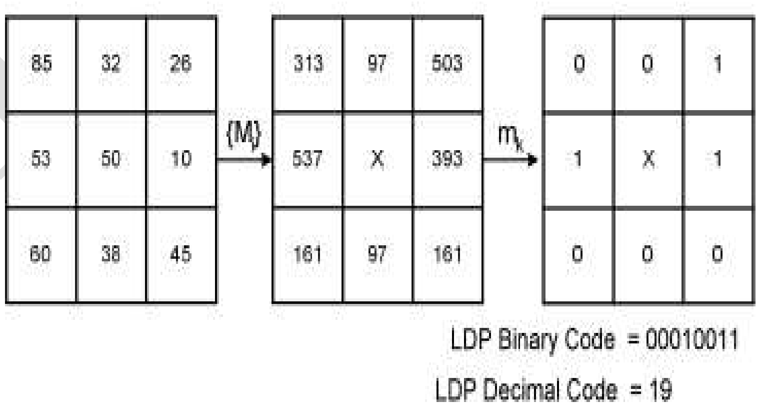


**Fig. 6. LDP code with k = 3.**

**F. Robustness of LDP**

Because edge responses are more stable than intensity values, the LDP pattern gives the same code even in the presence of noise and non-monotonic illumination change. Fig. 7 shows an original image and the same image after Gaussian white noise is added. After the noise is added, the fifth bit of the LBP code changes from 1 to 0, so the LBP pattern changes from a uniform to a non-uniform code. The LDP pattern is unchanged. This stability is the practical reason for preferring LDP in an agricultural setting, where leaf images are captured outdoors under lighting that cannot be controlled.

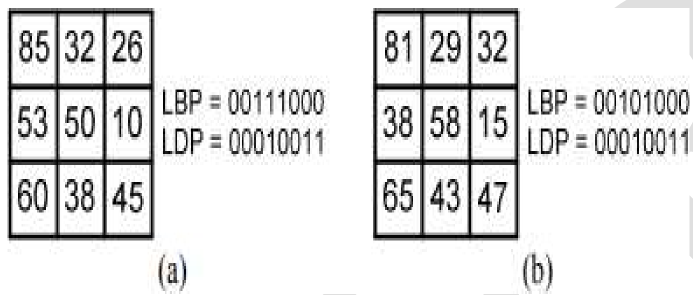


**Fig. 7. Stability of LDP against LBP. (a) Original image with LBP and LDP codes. (b) Image with noise, showing the new LBP and LDP codes.**

*G. Feature Extraction Procedure*

The input leaf image is first converted to grayscale to simplify processing. Local directional gradients are then computed to detect edges and texture features, and quantized into directional bins. A local neighbourhood is defined around each pixel and histograms of gradient direction within these neighbourhoods are computed, capturing the local directional patterns. The histograms are concatenated into a feature vector representing the whole image. This vector is what the classifier is trained on.

## IV. Implementation

**A. Software Environment**

The system was implemented in Python using TensorFlow and Keras within the Spyder scientific environment, with packages managed through a Conda environment. TensorFlow represents computation as a graph in which data flowing across the axes gives the framework its name; Keras provides the higher-level layer API used to define and train the network.

**B. Training Dataset**

The training dataset consists of images of plant leaves, some healthy and some infected with various diseases. The images must be labelled so that the system knows which leaves are healthy and which are diseased. Features are extracted from the training images, LDP features encoding the texture of a small local region, and these features are used to train the model. The model learns to associate the LDP features with the presence or absence of disease. Once trained, it classifies new images: LDP features are extracted from the unseen image and fed to the model, which predicts whether the leaf is healthy or diseased.


**Fig. 8. Training dataset.**

**C. Data Annotation**

Annotation means marking the regions on leaf images where the relevant patterns are observed, distinguishing healthy from diseased areas so the model has training examples to learn from. Annotators need training to recognize common leaf diseases and to understand the characteristic patterns of each condition. Affected areas are outlined carefully, with consistency maintained across the whole set, since inconsistent annotation is directly reflected in the model's confusion between classes.

**D. Model Training**

A diverse dataset of leaf images is collected, covering healthy and diseased leaves under varying lighting and backgrounds. The images are preprocessed to enhance features and remove noise, and local directional patterns are extracted. The dataset is split into training, validation and test sets, with k-fold cross-validation used to check that performance is not an artefact of one particular split.

A CNN architecture suitable for processing LDP features is then designed and trained, with hyperparameters and architecture adjusted based on validation performance. Regularization is applied to prevent overfitting. The trained model is evaluated on the held-out test set for accuracy, sensitivity and specificity.

**E. Saving the Trained Model**

Saving the model means preserving more than the weights. The learned weights and biases are stored along with the parameters defining the architecture, the preprocessing steps applied to input images before they reach the model, and the configuration of the LDP feature extraction itself, including the size of the local neighbourhood and the number of orientations. Without the last of these the saved weights are unusable, since features computed with a different LDP configuration would not match what the model was trained on. The complete package is serialized for later deployment.

**F. Inference**

At inference time the input leaf image is preprocessed, then LDP features are extracted by comparing intensity variations along different orientations within local neighbourhoods, at multiple scales and orientations across the

ISSN: 2249-7196
IJMRR/March-May 2024/ Volume 14/Issue 3/157-166
P. Sai Bhargavi / International Journal of Management Research & Review

image. The extracted features are passed to the trained model, which predicts the disease class. The result is then interpreted, and actions such as alerting the farmer or triggering an automated treatment system can follow.

## V. Results

The system was tested on apple leaf images from the Plant Village dataset across four classes. For each test image the system reports the health status and the identified disease. The results are shown in Figs. 9 to 12.


**Fig. 9. Detection result: not healthy — apple cedar rust.**

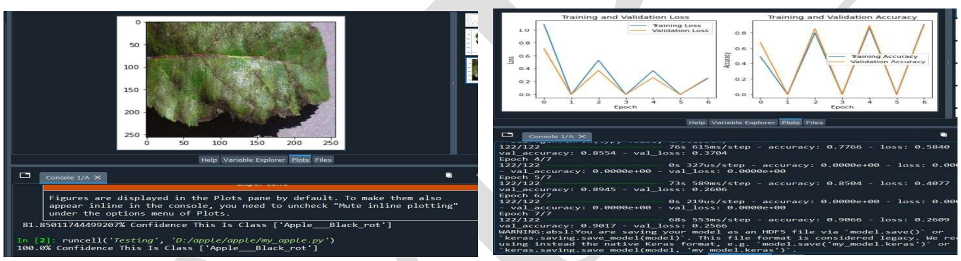


**Fig. 10. Detection result: not healthy — apple scab.**

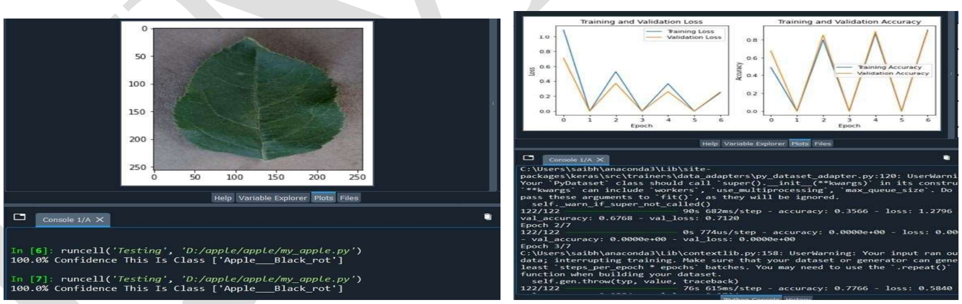


**Fig. 11. Detection result: apple healthy.**

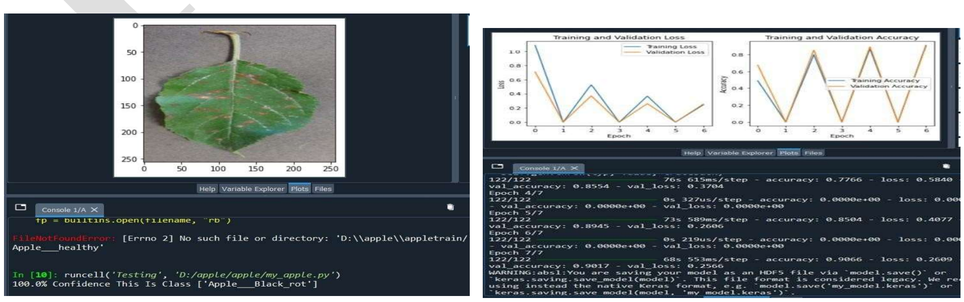


**Fig. 12. Detection result: not healthy — apple black rot.**

The four apple classes differ in how their symptoms present. Cedar rust shows as bright orange-yellow spots, scab as dark olive lesions with irregular margins, and black rot as concentric rings around a necrotic centre. Cedar rust is the easiest of the three for a colour-based method, since its colour is distinctive; scab and black rot are harder, because both are dark against a green background and are separated more by the texture and arrangement of the lesion than by colour. This is where the directional information in LDP contributes most, since the concentric structure of black rot and the diffuse margin of scab differ in edge orientation even where they are similar in intensity.

## VI. Conclusion And Future Scope

Plant leaf disease detection matters for maintaining the health and productivity of crops. Using machine learning and image processing, a system can be built that identifies diseases in leaves accurately enough to be useful. Such systems help farmers detect disease early, which allows timely intervention. The Local Directional Pattern proved a suitable descriptor for this task because it encodes edge response across eight directions rather than raw intensity, which keeps the code stable under the noise and illumination variation that field images inevitably carry.

Further advances in image processing, machine learning and deep learning will improve the accuracy and efficiency of these systems. Integration with IoT devices and drones would allow real-time monitoring of plant health across a field rather than inspection of individual leaves. Cloud computing for data analysis and storage would make the systems scalable and accessible to farmers who lack local computing resources. Extending the work beyond apple to other crops, and beyond four classes to a wider range of diseases, is the natural next step.

## References

[1] Y. M. Oo and N. C. Htun, "Plant leaf disease detection and classification using image processing," International Journal of Research and Engineering, vol. 5, no. 9, pp. 516–523, 2018.

[2] U. Shruthi, V. Nagaveni, and B. Raghavendra, "A review on machine learning classification techniques for plant disease detection," in Proc. 5th Int. Conf. Advanced Computing & Communication Systems (ICACCS), IEEE, 2019, pp. 281–284.

[3] M. El Sghair, R. Jovanovic, and M. Tuba, "An algorithm for plant diseases detection based on color features," Int. J. Agric. Sci., vol. 2, pp. 1–6, 2017.

[4] V. K. Vishnoi, K. Kumar, and B. Kumar, "Plant disease detection using computational intelligence and image processing," Journal of Plant Diseases and Protection, vol. 128, no. 1, pp. 19–53, 2021.

[5] L. Tan, J. Lu, and H. Jiang, "Tomato leaf diseases classification based on leaf images: a comparison between classical machine learning and deep learning methods," AgriEngineering, vol. 3, no. 3, pp. 542–558, 2021.

[6] T. Jabid, M. H. Kabir, and O. Chae, "Local directional pattern (LDP) — a robust image descriptor for object recognition," in Proc. 7th IEEE Int. Conf. Advanced Video and Signal-Based Surveillance, 2010, pp. 482–487.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), vol. 1, 2005, pp. 886–893.

[8] P. Patil, N. Alegar, and S. Meena, "Comparison of performance of classifiers — SVM, RF and ANN in potato blight disease detection using leaf images," in Proc. IEEE Int. Conf. Computational Intelligence and Computing Research (ICCIC), 2017, pp. 1–5.

[9] A. Cruz, Y. Ampatzidis, R. Pierro, A. Materazzi, A. Panattoni, L. De Bellis, and A. Luvisi, "Detection of grapevine yellows symptoms in Vitis vinifera L. with artificial intelligence," Computers and Electronics in Agriculture, vol. 157, pp. 63–76, 2019.

[10] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," Plants, vol. 8, no. 11, p. 468, 2019.

[11] B. S. Kusumo, A. Heryana, O. Mahendra, and H. F. Pardede, "Machine learning-based for automatic detection of corn-plant diseases using image processing," in Proc. Int. Conf. Computer, Control, Informatics and its Applications (IC3INA), IEEE, 2018, pp. 93–97.

[12] Q. Yao, Z. Guan, Y. Zhou, J. Tang, Y. Hu, and B. Yang, "Application of support vector machine for detecting rice diseases using shape and color texture features," in Proc. Int. Conf. Engineering Computation, IEEE, 2009, pp. 79–83.

[13] U. Mokhtar, N. E. Bendary, A. E. Hassenian, E. Emary, M. A. Mahmoud, H. Hefny, and M. F. Tolba, "SVM-based detection of tomato leaves diseases," in Intelligent Systems'2014, Springer, 2015, pp. 641–652.

[14] R. D. L. Pires, D. N. Gonçalves, J. P. M. Orüe, W. E. S. Kanashiro, J. F. Rodrigues Jr., B. B. Machado, and W. N. Gonçalves, "Local descriptors for soybean disease recognition," Computers and Electronics in Agriculture, vol. 125, pp. 48–55, 2016.

[15] R. Usha and S. Sathish, "Plant leaf disease detection using local direction pattern and deep learning techniques," in Proc. Int. Conf. System, Computation, Automation and Networking (ICSCAN), IEEE, 2019, pp. 1–6.

[16] P. Roy, K. Roy, and S. Saha, "Plant leaf disease detection using local direction pattern and convolutional neural network," in Proc. Int. Conf. Advances in Computing, Communication Control and Networking (ICACCCN), IEEE, 2019, pp. 1–6.

[17] Y. Wang, Y. Li, and X. Gao, "Plant leaf disease classification using local direction pattern and deep convolutional neural networks," in Proc. IEEE Int. Conf. Image Processing (ICIP), 2019, pp. 3011–3015.