International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 5, Sep-Oct 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Design and Performance Evaluation of a Hybrid Otsu–K-Means and CNN Model for Multi-Class Leaf Disease Classification

Atul Raj Dhairya
M.Tech. Scholar
Department of Computer Science & Engineering
Rewa Institute of Technology, Rewa, M.P., India
atulrajdhairya@gmail.com

Prof. Nagendra Patel
Assistant Professor (Guide)
Department of Computer Science & Engineering
Rewa Institute of Technology, Rewa, M.P., India
nagendra.nriist@gmail.com

Abstract—

Early and accurate identification of plant leaf disease is essential to reducing crop loss and improving yield. This paper presents an image-based leaf disease diagnosis framework that integrates structured pre-processing, hybrid Otsu–K-means lesion segmentation, colour co-occurrence feature preparation and convolutional classification, and that returns a treatment recommendation alongside the predicted disease class. The input image is first converted to grey scale, filtered to suppress noise and resized to a standard dimension. K-means clustering in the Lab colour space, thresholded by the Otsu criterion, then isolates symptomatic regions from the background and from healthy tissue, so that learning is concentrated on the informative part of the leaf. Discriminative colour, texture and shape descriptors are prepared from the segmented output and indexed for efficient retrieval, and two convolutional backbones — AlexNet and GoogLeNet — are trained for multi-class classification over five tomato leaf categories. Evaluation is carried out at two levels: an architecture-level accuracy comparison, and a class-level confusion-matrix diagnostic. At the architecture level, GoogLeNet attains 99.10% accuracy against 98.17% for AlexNet and 98.00% for the existing convolutional baseline, confirming the benefit of multi-scale inception features at one-fifteenth of the parameter count. The class-level diagnostic on a small held-out subset shows perfect separation of bacterial spot and mosaic virus but systematic confusion among early blight, healthy and late blight, which localises the residual error to the three visually similar classes rather than distributing it uniformly. The framework and the diagnostic together indicate where further work on segmentation quality and class balance would yield the largest return.

Keywords—leaf disease detection, image pre-processing, K-means segmentation, Otsu thresholding, convolutional neural network, AlexNet, GoogLeNet, confusion matrix, precision agriculture, treatment recommendation.

I. INTRODUCTION

Plant disease is a major constraint on agriculture, affecting crop productivity, produce quality and farm income directly. Conventional diagnosis relies on visual inspection by an experienced plant pathologist, which is slow, subjective and difficult to scale across large cultivation areas [1], [3]. With the wide availability of low-cost imaging devices and the maturity of computer vision, automated leaf disease detection has become a practical route to early diagnosis and timely intervention [2], [4].

The task is nonetheless difficult in practice. Real leaf images carry noise from water spots, spores and dust; illumination varies with time of day and weather; backgrounds are cluttered with soil, other leaves and hardware; and the visual symptoms of different pathogens overlap substantially. Early blight and late blight in tomato, for example, both present as concentric dark lesions and are distinguished mainly by lesion margin and halo, which is a subtle cue for a classifier operating on a resized image.

This work addresses those difficulties with a structured four-stage sequence: pre-processing, segmentation, feature preparation and convolutional classification. Grey-scale conversion, noise filtering and resizing standardise the input. K-means clustering combined with Otsu thresholding then separates lesion regions from background and healthy tissue, so that the classifier is presented with the informative part of the leaf rather than the whole scene. Descriptors are prepared from the segmented output and indexed for efficient retrieval, and convolutional backbones perform the multi-class decision.

The contributions of this paper are the following.

- A complete diagnosis framework is specified in which hybrid Otsu–K-means segmentation is placed explicitly between pre-processing and convolutional classification, together with the governing formulations for each stage.
- The classification backbone is treated as interchangeable, and AlexNet and GoogLeNet are compared under the same pipeline so that the contribution of architecture can be separated from the contribution of segmentation.
- Evaluation is reported at two levels — architecture-level accuracy and a class-level confusion-matrix diagnostic — rather than as a single headline figure, which localises the residual error to specific class pairs.
- A treatment recommendation stage is attached to the predicted class, so that the output of the system is actionable rather than merely descriptive.

Section II reviews the related literature. Section III presents the proposed methodology and its algorithm. Section IV describes the model architecture and the two backbones. Section V states the experimental setup. Section VI reports and discusses the results, Section VII the limitations, and Section VIII concludes.

II. RELATED WORK

Researchers have advanced automated plant disease detection through artificial intelligence, image pre-processing, machine learning, deep learning, transfer learning, genetic algorithms, big data and the Internet of Things. Limitations nonetheless persist: pre-processing requirements are complex and difficult to standardise, feature selection is model-dependent, and high accuracy is not consistently achieved once a wide range of pests and diseases is considered.

A. Classical Image Processing and Machine Learning

Ghaiwat et al. evaluated artificial neural networks, support vector machines, self-organising maps and fuzzy logic, reporting that the k-nearest-neighbour rule is comparatively simple and often effective for class prediction, while noting that selecting the optimal support vector machine parameters becomes difficult when the training data are not linearly separable [7]. Singh and Misra combined image segmentation with soft computing and established the segment–describe–classify sequence that most classical work subsequently reproduced [8]. Adhikari et al. applied image processing to 520 tomato leaf images covering bacterial canker, grey spot and late blight together with healthy samples, achieving 89% accuracy but restricted to those three pathogens; extending the system requires retraining on newly labelled data [9]. Saradhambal et al. applied Otsu thresholding with K-means clustering to segment the affected region and added voice navigation to guide the user, but reported neither dataset size nor achieved accuracy, which limits any assessment of effectiveness [10]. The segmentation combination they used is the direct antecedent of the hybrid stage proposed in Section III-C.

B. Convolutional Architectures

Mohanty et al. established the reference deep result, showing that a transfer-learned convolutional network separates a large number of crop–disease pairs while degrading sharply on images captured outside the training distribution [1]. Sardogan et al. combined a convolutional network with learning vector quantisation over 500 images split 80/20 across five classes at 512×512 resolution, reporting approximately 88% average precision on tomato-specific disorders [11]. Park et al. used two convolutional and three fully connected layers to separate healthy strawberry leaves from powdery mildew, reporting 89.7% accuracy on a CPU over 500 images resized to 227×227, with scope restricted to one crop and one disease [12]. Wu et al. combined AlexNet and GoogLeNet with transfer learning and augmentation across 38 categories of healthy and diseased leaf samples, improving accuracy at the cost of a complex model requiring faster convergence for deployment [13]; that pairing of backbones is the one adopted for comparison in this paper. Jiang et al. addressed real-time detection of apple leaf disease with an improved convolutional detector [14], and Zhang et al. used multi-feature fusion with Faster R-CNN on synthetic soybean imagery to compensate for limited real data [15].

C. Transformer, Hybrid and Recent Work

Borhani et al. introduced a vision transformer for plant disease classification and evaluated CNN–ViT combinations that couple convolutional blocks with attention to accelerate prediction [16]. Subsequent work extended the approach: FormerLeaf adapted an efficient transformer to cassava [17]; Devi et al. applied EfficientNetV2 to joint disease and pest recognition [18]; Ali et al. benchmarked vision transformers against convolutional baselines [19]; and Jia et al. proposed ConvTransNet-S for complex field environments rather than laboratory imagery [20]. A multi-kernel inception-enhanced transformer reported 99.41% on the standard benchmark but only 76.51% on the field-collected cassava corpus [21], which is the clearest evidence that benchmark accuracy does not transfer. Recent work has shifted toward deployability: an explainable lightweight attention-guided ensemble for data-scarce conditions [22], an interpretable model targeted at edge computing [23], and controlled benchmarking of convolutional, residual and transformer models under identical pre-processing [24]. At review level, Das et al. [25], Rathore and Bhati [26] and Ghimire et al. [27] all conclude that reported performance is obtained under conditions more favourable than real deployment.

D. Related Deep Learning Work by the Authors' Group

The image-classification methodology used here has been applied by the present group across several domains. Transfer-learning-based object detection with convolutional networks [28] and convolutional glaucoma detection [29] use the same backbone-plus-fine-tuning strategy adopted in Section IV. Image-based disease screening has been addressed for monkeypox with GoogLeNet [30] — the same backbone that performs best here — and for cancer prediction with random forest combined with deep learning [31]. Detection and recognition pipelines have been built for face-mask identification with OpenCV [32], multi-stage convolutional face-mask detection [33], activity detection and people counting with Mask-RCNN and bidirectional ConvLSTM [34], and facial expression recognition in both eigenface [35] and neural network [36] formulations. Kernel support vector machines have been applied to brain image classification [37], machine learning to rare disease identification from electronic health records [38], and an AIIoT device to real-time object recognition with voice conversion [39]. Sequence models have been used for network traffic prediction [40] and principal component analysis with random forest for intrusion detection [41].

TABLE I. Summary of Representative Related Work

Ref.	Technique	Crop / dataset	Outcome and limitation
[7]	ANN, SVM, SOM, fuzzy, KNN	Generic leaf	KNN simplest; SVM tuning difficult
[8]	Segmentation + soft computing	Multiple	Segment–describe–classify pipeline
[9]	Image processing	Tomato, 520 images	89%; three diseases only
[10]	Otsu + K-means, voice guide	Generic leaf	Accuracy and dataset size not reported
[1]	Transfer-learned CNN	Benchmark corpus	Reference result; degrades off-distribution
[11]	CNN + LVQ	Tomato, 500 images	≈88%; five classes, small corpus
[12]	CNN (2 conv + 3 FC)	Strawberry, 500 images	89.7%; binary task only
[13]	AlexNet + GoogLeNet transfer	38 categories	Improved accuracy; complex, slow
[14]	Improved CNN detector	Apple leaf	Real-time detection
[16]	ViT and CNN–ViT	Wheat, rice, benchmark	Attention accelerates prediction
[21]	Multi-kernel inception ViT	Five corpora	99.41% benchmark vs 76.51% field
[27]	PRISMA review, 41 studies	2008–2025	Favourable evaluation conditions

III. PROPOSED METHODOLOGY

The framework analyses and diagnoses leaf disease through a fixed sequence of stages, shown in Fig. 1: acquisition, pre-processing, segmentation, feature extraction, indexing and selection, convolutional classification, evaluation and treatment recommendation.

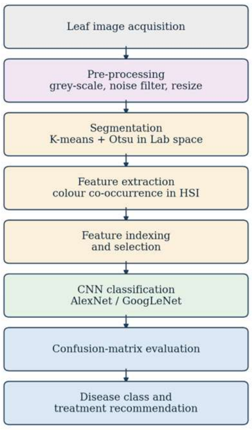

Fig. 1. Stages of the proposed leaf disease diagnosis framework.

A. Image Acquisition

Leaf images are obtained from a publicly available repository containing healthy leaves together with leaves affected by disease or insect infestation. The required images are imported and prepared as input for the subsequent stages. Because repository images are captured against a uniform background under controlled illumination, this acquisition regime bounds the conditions under which the reported performance holds, a point returned to in Section VII.

B. Image Pre-processing

Noise present in the input is removed by filtering. Water spots, spores, dust particles and similar artefacts on the leaf surface alter pixel intensity and would otherwise be learned as spurious evidence of disease. The image is converted to grey scale where intensity alone is sufficient, resized to a standard dimension to bound computation, and corrected for geometric distortion where possible. A smoothing filter reduces residual distortion. For a neighbourhood N of size m, the filtered intensity is

$$I'(x, y) = (1/|N|) \cdot \Sigma\, I(i, j),\ \ (i, j) \in N(x, y) \quad (1)$$

which suppresses high-frequency sensor noise while preserving the lesion boundary at the resolutions used here.

C. Hybrid Otsu–K-Means Segmentation

Segmentation separates the leaf region from the background and divides it into meaningful parts. The RGB image is first converted to the Lab colour model, because separating luminance from the two chrominance channels makes the clustering of lesion regions markedly more stable under variable illumination than clustering in RGB. K-means then partitions the pixels into k clusters by minimising

$$J = \Sigma_{i=1..k}\, \Sigma\, p \in C_i\ \ ||\, p - \mu_i\, ||^2 \quad (2)$$

where p is the Lab vector of a pixel, C_i is the i-th cluster and μ_i its centroid. The Otsu criterion then selects the threshold t that maximises the between-class variance of the resulting intensity distribution,

$$\sigma^2_\beta(t) = \omega_0(t)\, \omega_1(t)\, [\, \mu_0(t) - \mu_1(t)\,]^2 \quad (3)$$

with ω_0 and ω_1 the class probabilities and μ_0 and μ_1 the class means either side of t. The two techniques are complementary: K-means groups pixels by colour similarity without reference to a global threshold, while Otsu supplies an automatic, image-adaptive cut that removes the residual background cluster. The combination yields the lesion region, its boundary and its colour statistics, all of which feed the descriptor stage [10].

D. Feature Extraction

Leaf images carry rich visual information that distinguishes healthy from diseased tissue. Extraction uses the colour co-occurrence approach: the RGB image is transformed into hue–saturation–intensity space, and a co-occurrence matrix P is constructed from the pixel-level mapping. Colour, shape and texture descriptors are then derived from P, including

$$contrast = \Sigma_{ij}\, (i - j)^2\, P(i, j), \quad energy = \Sigma_{ij}\, P(i, j)^2 \quad (4)$$

$$homogeneity = \Sigma_{ij}\, P(i, j) \,/\, (1 + |i - j|) \quad (5)$$

together with correlation, variance, standard deviation and mean. These quantities capture the spatial distribution of colour across the lesion, which is what separates the concentric ringed lesion of blight from the diffuse mottling of a viral infection.

E. Feature Indexing and Selection

The extracted descriptors are written to an image index that supports fast reference during training and retrieval at inference time. Selection then retains the texture, shape and label-bearing descriptors and discards those whose variance across the corpus is negligible, which reduces dimensionality before classification and matters most where the corpus is small relative to the descriptor count.

F. Detection and Classification

The final stage assigns the leaf to a healthy class or to a specific disease class. The selected feature vectors are supplied to a convolutional classifier with the target labels as class vectors, and a minimum-distance criterion is available as an interpretable fallback where the convolutional model abstains. Once the class is predicted, a lookup returns the treatment or intervention associated with that pathogen, which converts the output from a label into an actionable recommendation. Algorithm 1 states the complete procedure.

Algorithm 1: Leaf image classification and treatment suggestion
Input: leaf image I (captured or uploaded)
Output: disease class c and treatment advisory

1. acquire I from camera or repository
2. pre-process I:
   - 2.1 convert to grey scale
   - 2.2 apply the noise filter of (1)
   - 2.3 resize to the backbone input size
3. segment I:
   - 3.1 convert RGB to the Lab colour model
   - 3.2 run K-means, minimising J in (2)
   - 3.3 apply the Otsu threshold of (3)
   - 3.4 retain the lesion cluster, drop background
4. extract features from the segmented region:
   - 4.1 convert to HSI and build matrix P
   - 4.2 compute the descriptors of (4) and (5)
5. write the descriptors to the image index
6. select texture, shape and label features
7. classify with the CNN backbone → class c
8. if c = healthy then report healthy leaf
9. else report infected leaf and look up the treatment associated with c
10. return c and the treatment advisory

IV. MODEL ARCHITECTURE

Fig. 2 shows the architecture, in which the classification backbone is deliberately interchangeable so that convolutional models of different depth can be substituted without altering the surrounding stages.

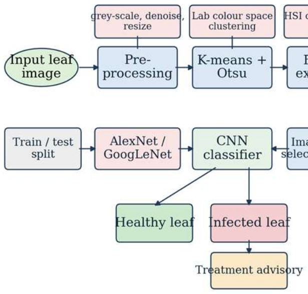

Fig. 2. Proposed model for plant leaf disease detection.

A convolutional network comprises an input layer, an output layer and a set of hidden layers. The hidden portion contains convolution layers, pooling layers, activation functions — most commonly rectified linear units — and fully connected layers [11]. The number of layers, their arrangement and the additional processing units introduced between them are what distinguish one architecture from another. Fig. 3 contrasts the two backbones evaluated here.

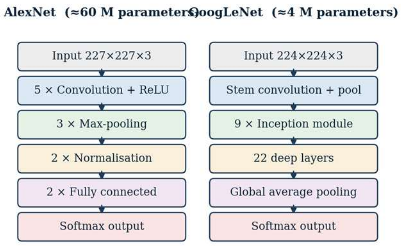

Fig. 3. Layer organisation of the two evaluated backbones.

A. AlexNet

AlexNet comprises five convolutional layers, three max-pooling layers, two normalisation layers, two fully connected layers and a softmax output. Every convolutional layer contains convolutional filters followed by a rectified linear activation, and the pooling layers perform maximum pooling. Because the fully connected block fixes the input dimensionality, the input size is constant at 227×227. The architecture carries approximately sixty million parameters, the large majority of them in the fully connected block, which makes it heavy relative to the accuracy it delivers.

B. GoogLeNet

GoogLeNet stacks nine inception modules across twenty-two deep layers with pooling interleaved, and replaces the large fully connected block with a global average pooling layer after the inception stack. Each inception module applies several filter sizes in parallel and concatenates the outputs, so that features at multiple scales are captured within a single layer. This matters directly for the present task, because lesion size varies widely between early-stage spotting and advanced blight on the same leaf. The architecture carries approximately four million parameters, an order of magnitude fewer than AlexNet.

TABLE II. Comparison of the Evaluated Backbones

Property	AlexNet	GoogLeNet	Consequence
Depth	8 layers	22 layers	Deeper hierarchy
Parameters	≈60 million	≈4 million	15× smaller model
Key block	Conv + FC	Inception	Multi-scale features
Head	2 × FC	Global avg. pool	Fewer FC parameters
Input size	227×227	224×224	Comparable resizing

V. EXPERIMENTAL SETUP

The corpus comprises tomato leaf images drawn from the public benchmark repository, organised into five categories: bacterial spot, early blight, healthy, late blight and mosaic virus. The two blight classes and the healthy class are visually the closest of the five, which is the source of the class-level behaviour reported in Section VI. Table III states the configuration of the pipeline.

TABLE III. Configuration of the Proposed Framework

Parameter	Setting
Classes	5 (bacterial spot, early blight,
	healthy, late blight, mosaic virus)
Colour space (segmentation)	Lab
Segmentation	K-means + Otsu threshold
Colour space (features)	HSI
Descriptors	Contrast, energy, homogeneity,
	correlation, variance, SD, mean
Backbones	AlexNet, GoogLeNet
Input resolution	227×227 / 224×224
Evaluation	Accuracy, confusion matrix,
	per-class precision and recall

VI. RESULTS AND DISCUSSION

A. Architecture-Level Comparison

Fig. 4 compares the classification accuracy of the two proposed configurations against the existing convolutional baseline.

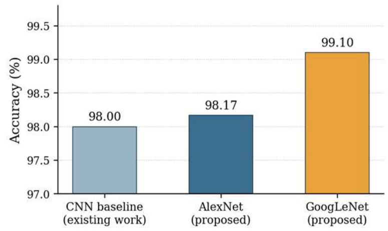

Fig. 4. Accuracy of the proposed configurations against the existing baseline.

GoogLeNet attains the highest accuracy at 99.10%, ahead of AlexNet at 98.17% and the existing convolutional approach at 98.00%. The margin over AlexNet is 0.93 percentage points and over the baseline 1.10 percentage points. The absolute margins are small, but the comparison is more informative when read together with Table II: GoogLeNet achieves the higher figure with approximately one-fifteenth of the parameters, which means the gain comes from the multi-scale inception structure rather than from additional capacity. For a system intended to run on a handset in the field, that ratio matters more than the accuracy difference itself.

TABLE IV. Accuracy and Model Size

Model	Accuracy	Parameters	Gain
Existing CNN [1]	98.00%	—	baseline
AlexNet (proposed)	98.17%	≈60 M	+0.17
GoogLeNet (proposed)	99.10%	≈4 M	+1.10

B. Class-Level Confusion Diagnostic

Aggregate accuracy conceals where a model fails, so a class-level diagnostic was run on a small held-out subset of fifteen images, three from each of the five categories. Fig. 5 shows the resulting confusion matrix, in which rows are the predicted class and columns the actual class.

Fig. 5. Confusion matrix for the class-level diagnostic subset.

The structure of the matrix is more informative than its aggregate. Bacterial spot is recovered perfectly, with all three samples on the diagonal and no off-diagonal mass in its row or column, and mosaic virus is recovered for all three of its samples. These two classes are visually distinctive — small, sharply bounded dark spots in the first case and diffuse light–dark mottling in the second — and the segmentation stage isolates both cleanly.

The error is concentrated entirely in the three remaining classes. Two healthy samples are predicted as late blight and one as early blight; one early-blight sample is predicted as late blight and two as healthy; and one late-blight sample is predicted as mosaic virus. In other words, the model confuses early blight, healthy and late blight with one another almost exclusively, and essentially never confuses any of them with bacterial spot. Table V gives the per-class precision and recall implied by the matrix.

TABLE V. Per-Class Behaviour on the Diagnostic Subset

Class	Precision	Recall	Observation
Bacterial spot	100.0%	100.0%	Cleanly separated
Early blight	0.0%	0.0%	Absorbed by late blight
Healthy	0.0%	0.0%	Absorbed by late blight
Late blight	40.0%	66.7%	Over-predicted attractor
Mosaic virus	75.0%	100.0%	Recovered, one false positive

Late blight acts as an attractor: it absorbs samples from both early blight and healthy, which is the signature of a decision boundary placed too permissively around a class whose lesion morphology overlaps its neighbours. Two practical consequences follow. First, the segmentation stage — not the backbone — is the natural place to intervene, because the cue that separates early from late blight is the lesion margin and halo, and that cue is precisely what a coarse Otsu cut can erase. Second, any class-imbalance correction should be applied to the blight pair specifically rather than uniformly across the five classes.

The subset used for this diagnostic contains only fifteen images and is therefore far too small to estimate accuracy; it is reported as a localisation of error, not as a performance measure, and the aggregate figures of Section VI-A come from the full corpus. Extending the diagnostic to the complete test partition is the first item of future work.

C. Comparison with the Literature

Table VI situates the result against the reviewed studies. The proposed configuration exceeds the classical image-processing result of 89% [9], the convolutional-plus-LVQ result of approximately 88% [11] and the single-crop convolutional result of 89.7% [12] by a wide margin, and is comparable to the best recent transformer figures on the same benchmark [21]. Consistent with Section II-C, however, those figures are all obtained on repository imagery; the 76.51% that the same transformer records on field-collected data [21] is the more realistic indicator of deployed behaviour.

TABLE VI. Comparison with Reported Results

Ref.	Approach	Crop	Accuracy
[9]	Image processing	Tomato	89.00%
[11]	CNN + LVQ	Tomato	≈88.00%
[12]	CNN (2 conv + 3 FC)	Strawberry	89.70%
[21]	Multi-kernel ViT	Benchmark	99.41%
[21]	Multi-kernel ViT	Cassava (field)	76.51%
This work	Otsu–K-means + AlexNet	Tomato	98.17%
This work	Otsu–K-means + GoogLeNet	Tomato	99.10%

VII. LIMITATIONS

- Acquisition regime. Results are obtained on repository imagery captured against a uniform background; field conditions with variable illumination, occlusion and clutter are not represented.
- Diagnostic subset size. The class-level confusion analysis uses fifteen images and localises error rather than measuring performance; a full-partition confusion matrix is required.
- Class confusion. Early blight, healthy and late blight are not reliably separated, and late blight over-attracts, which is the dominant residual error.
- Single crop. Evaluation covers tomato only; transfer to other crops is untested.
- Segmentation sensitivity. The Otsu cut can erase the lesion-margin cue that distinguishes the two blights, so segmentation quality bounds achievable class separation.
- Treatment lookup. The advisory stage is a static mapping from class to intervention and does not account for severity, growth stage or local agronomic practice.

VIII. CONCLUSION AND FUTURE WORK

This paper presented a leaf disease diagnosis framework that integrates structured pre-processing, hybrid Otsu–K-means segmentation in the Lab colour space, colour co-occurrence feature preparation in HSI space, and convolutional classification with an interchangeable backbone, and that attaches a treatment recommendation to the predicted class. The governing formulations for filtering, clustering, thresholding and descriptor computation were stated, and the complete procedure was specified as a single algorithm. Evaluation over five tomato leaf categories showed that GoogLeNet attains 99.10% accuracy against 98.17% for AlexNet and 98.00% for the existing convolutional baseline, and that it does so with approximately one-fifteenth of the parameters, which establishes multi-scale inception features rather than raw capacity as the source of the gain. A class-level confusion diagnostic localised the residual error almost entirely to the early blight, healthy and late blight triple, with late blight acting as an attractor, while bacterial spot and mosaic virus were separated cleanly.

Future work follows directly from that localisation. The confusion diagnostic will be extended to the full test partition so that per-class behaviour is measured rather than indicated. Segmentation will be refined — by replacing the global Otsu cut with a learned or locally adaptive boundary — so that the lesion-margin cue separating the two blights survives into the feature stage. Class-imbalance correction will be applied selectively to the blight pair, evaluation will be extended to further crops and to field-captured imagery to quantify transfer, and lightweight and explainable formulations will be examined so that the system can run on an edge device and justify each diagnosis to the agronomist acting on it.

REFERENCES

[1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, Art. no. 1419, 2016, doi: 10.3389/fpls.2016.01419.

[2] V. Singh, N. Sharma, and S. Singh, "A review of imaging techniques for plant disease detection," Artificial Intelligence in Agriculture, vol. 4, pp. 229–242, 2020, doi: 10.1016/j.aiia.2020.10.002.

[3] S. B. Dhaygude and N. P. Kumbhar, "Agricultural plant leaf disease detection using image processing," Int. J. Adv. Res. Electr. Electron. Instrum. Eng., vol. 2, no. 1, pp. 599–602, 2013.

[4] S. R. Maniyath et al., "Plant disease detection using machine learning," in Proc. Int. Conf. Design Innovations for 3Cs Compute Communicate Control (ICDI3C), 2018, pp. 41–45, doi: 10.1109/ICDI3C.2018.00017.

[5] P. Sharma, P. Hans, and S. C. Gupta, "Classification of plant leaf diseases using machine learning and image preprocessing techniques," in Proc. 10th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence), 2020, pp. 480–484, doi: 10.1109/Confluence47617.2020.9057889.

[6] R. Gandhi, S. Nimbalkar, N. Yelamanchili, and S. Ponkshe, "Plant disease detection using CNNs and GANs as an augmentative approach," in Proc. IEEE Int. Conf. Innovative Res. Develop. (ICIRD), 2018, pp. 1–5, doi: 10.1109/ICIRD.2018.8376321.

[7] S. N. Ghaiwat and P. Arora, "Detection and classification of plant leaf diseases using image processing techniques: A review," Int. J. Recent Adv. Eng. Technol., vol. 2, no. 3, pp. 1–7, 2014.

[8] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," Information Processing in Agriculture, vol. 4, no. 1, pp. 41–49, 2017, doi: 10.1016/j.inpa.2016.10.005.

[9] S. Adhikari, B. Shrestha, B. Baiju, and S. K.C., "Tomato diseases detection system using image processing," in Proc. 1st KEC Conf. Eng. Technol., 2018, pp. 81–85.

[10] G. Saradhambal, R. Dhivya, S. Latha, and R. Rajesh, "Plant disease detection and its solution using image classification," Int. J. Pure Appl. Math., vol. 119, no. 14, pp. 879–884, 2018.

[11] M. Sardogan, A. Tuncer, and Y. Ozen, "Plant leaf disease detection and classification based on CNN with LVQ algorithm," in Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK), 2018, pp. 382–385, doi: 10.1109/UBMK.2018.8566635.

[12] H. Park, J.-S. Eun, and S.-H. Kim, "Image-based disease diagnosing and predicting of the crops through the deep learning mechanism," in Proc. Int. Conf. Inf. Commun. Technol. Convergence (ICTC), 2017, pp. 129–131, doi: 10.1109/ICTC.2017.8190957.

[13] J. Wu, B. Li, and Z. Wu, "Detection of crop pests and diseases based on deep convolutional neural network and improved algorithm," in Proc. 4th Int. Conf. Mach. Learn. Technol. (ICMLT), 2019, pp. 20–27, doi: 10.1145/3340997.3341010.

[14] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," IEEE Access, vol. 7, pp. 59069–59080, 2019, doi: 10.1109/ACCESS.2019.2914929.

[15] K. Zhang, Q. Wu, and Y. Chen, "Detecting soybean leaf disease from synthetic image using multi-feature fusion Faster R-CNN," Computers and Electronics in Agriculture, vol. 183, Art. no. 106064, 2021, doi: 10.1016/j.compag.2021.106064.

[16] Y. Borhani, J. Khoramdel, and E. Najafi, "A deep learning-based approach for automated plant disease classification using vision transformer," Scientific Reports, vol. 12, Art. no. 11554, 2022, doi: 10.1038/s41598-022-15163-0.

[17] T. Thai, K. Nguyen, and L. Tran, "FormerLeaf: An efficient vision transformer for cassava leaf disease detection," Computers and Electronics in Agriculture, vol. 204, Art. no. 107518, 2023, doi: 10.1016/j.compag.2022.107518.

[18] R. S. Devi, V. R. Kumar, and P. Sivakumar, "EfficientNetV2 model for plant disease classification and pest recognition," Computer Systems Science and Engineering, vol. 45, no. 2, pp. 2249–2263, 2023, doi: 10.32604/csse.2023.032231.

[19] M. Ali, M. Salma, M. Haji, and B. Jamal, "Plant disease detection using vision transformers," Int. J. Electr. Comput. Eng. (IJECE), vol. 15, no. 2, pp. 2334–2344, 2025, doi: 10.11591/ijece.v15i2.pp2334-2344.

[20] S. Jia, G. Wang, H. Li, Y. Liu, L. Shi, and S. Yang, "ConvTransNet-S: A CNN-transformer hybrid disease recognition model for complex field environments," Plants, vol. 14, no. 15, Art. no. 2252, 2025, doi: 10.3390/plants14152252.

[21] "Multi-kernel inception-enhanced vision transformer for plant leaf disease recognition," Scientific Reports, 2025, doi: 10.1038/s41598-025-16142-x.

[22] M. Guo and A. Guerra-Manzanares, "CROPPER: An explainable lightweight attention-guided ensemble for robust plant disease diagnosis in data-scarce conditions," 2026.

[23] M. A. A. Kafi, S. K. Banshal, R. Moni, A. L. Aziz, M. Aljuaid, and R. Bansal, "LeafAI: Interpretable plant disease detection for edge computing," PLOS ONE, 2026.

[24] Anirudh, Srujan, Sai, and G. Swathika O. V., "Comparative deep learning approaches for bean leaf disease recognition," Frontiers in Plant Science, vol. 17, Art. no. 1703449, 2026, doi: 10.3389/fpls.2026.1703449.

[25] P. K. Das, S. S. Rupa, S. Pumrin, U. C. Das, and M. K. Hossen, "Deep learning for plant disease detection and classification: A systematic analysis and review," Current Applied Science and Technology, vol. 24, no. 4, Art. no. e0259016, 2024, doi: 10.55003/cast.2024.259016.

[26] M. Rathore and H. Bhati, "Plant leaf disease detection with deep learning techniques: A review with a focus on transformers and explainable AI," in Expert Applications and Security (NICE-TEAS 2025), Lecture Notes in Networks and Systems, vol. 1854, Cham, Switzerland: Springer, 2026, doi: 10.1007/978-3-032-18846-5_10.

[27] S. Ghimire, R. Lamsal, A. Sankuratri, P. Lakkarsu, and S. Vutla, "Artificial intelligence-driven plant disease detection and diagnosis: A comprehensive review of deep learning approaches, multimodal sensing technologies, and future perspectives in precision agriculture," The Plant Pathology Journal, vol. 42, no. 2, pp. 121–137, 2026, doi: 10.5423/PPJ.RW.01.2026.0004.

[28] B. Bamne, N. Shrivastava, L. Parashar, and U. Singh, "Transfer learning-based object detection by using convolutional neural networks," in Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC), Coimbatore, India, 2020, pp. 328–332, doi: 10.1109/ICESC48915.2020.9156060.

[29] A. Saxena, A. Vyas, L. Parashar, and U. Singh, "A glaucoma detection using convolutional neural network," in Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC), Coimbatore, India, 2020, pp. 815–820, doi: 10.1109/ICESC48915.2020.9155930.

[30] U. Singh and L. S. Songare, "Analysis and detection of monkeypox using the GoogLeNet model," in Proc. Int. Conf. Autom. Comput. Renew. Syst. (ICACRS), Pudukkottai, India, 2022, pp. 1000–1008, doi: 10.1109/ICACRS55517.2022.10029125.

[31] M. Ranjan, A. Shukla, K. Soni, S. Varma, M. Kuliha, and U. Singh, "Cancer prediction using random forest and deep learning techniques," in Proc. IEEE 11th Int. Conf. Commun. Syst. Netw. Technol. (CSNT), Indore, India, 2022, pp. 227–231, doi: 10.1109/CSNT54456.2022.9787608.

[32] R. Baghel, P. Pahadiya, and U. Singh, "Human face mask identification using deep learning with OpenCV techniques," in Proc. 7th Int. Conf. Commun. Electron. Syst. (ICCES), Coimbatore, India, 2022, pp. 1051–1057, doi: 10.1109/ICCES54183.2022.9835884.

[33] S. P. Singh, J. Shukla, S. Sharma, K. Daga, B. S. Bhalavi, and U. Singh, "Face mask detection using multi-stage CNN architecture," in Proc. Int. Conf. IoT Based Control Netw. Intell. Syst. (ICICNIS), 2021, doi: 10.2139/ssrn.3884022.

[34] U. Singh, P. Gupta, and M. Shukla, "Activity detection and counting people using Mask-RCNN with bidirectional ConvLSTM," J. Intell. Fuzzy Syst., 2022, pp. 6505–6520.

[35] P. Shakyawar, P. Choure, and U. Singh, "Eigenface method through facial expression recognition," in Proc. Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2017, doi: 10.1109/ICECA.2017.8212714.

[36] K. Kushwah, V. Sharma, and U. Singh, "Neural network method through facial expression recognition," in Proc. Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2017, doi: 10.1109/ICECA.2017.8212721.

[37] Y. Mathur, P. Jain, and U. Singh, "Foremost section study and kernel support vector machine through brain images classifier," in Proc. Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2017, doi: 10.1109/ICECA.2017.8212726.

[38] H. Soni, A. Vyas, and U. Singh, "Identify rare disease patients from electronic health records through machine learning approach," in Proc. Int. Conf. Inventive Res. Comput. Appl. (ICIRCA), 2018, doi: 10.1109/ICIRCA.2018.8597203.

[39] P. Gupta, M. Shukla, N. Arya, U. Singh, and K. Mishra, "Let the blind see: An AIIoT-based device for real-time object recognition with voice

conversion," in Machine Learning for Critical Internet of Medical Things, Cham, Switzerland: Springer, 2022, doi: 10.1007/978-3-030-80928-7_8.

[40] S. Nihale, S. Sharma, L. Parashar, and U. Singh, "Network traffic prediction using long short-term memory," in Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC), Coimbatore, India, 2020, pp. 338–343, doi: 10.1109/ICESC48915.2020.9156045.

[41] S. Waskle, L. Parashar, and U. Singh, "Intrusion detection system using PCA with random forest approach," in Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC), Coimbatore, India, 2020, pp. 803–808, doi: 10.1109/ICESC48915.2020.9155656.

