International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

A Hybrid Enhancing and Optimizing Crops Disease and Land Cover Classification Using Adaptive Recurrent FusionNet Framework

Kodavati Ram Sanjay[1], Ruban Kumar S[2], Saran Raj S[3], Dr. Arun Kumar[4]
[1,2,3,]Dept. of CSE (IoT), [4]Dept. of CSES,SRM IST, Ramapuram
Chennai, Tamil Nadu, India

Abstract- — Automated detection of crop leaf diseases and classification of remote sensing land cover categories remain challenging owing to complex backgrounds, illumination vari-ability, spectral distortions, and high intra-class visual similarity. Existing frameworks provide strong baselines but commonly suf-fer from scale-sensitive segmentation, redundant feature fusion, limited contextual representation, and slow convergence in high-dimensional feature spaces. This paper proposes the Adaptive Recurrent FusionNet (ARFusionNet) framework — a Flask-based web application that integrates four coordinated inno-vations: Multi-Scale Adaptive Contrast Normalisation (MACN) for illumination-robust preprocessing; Graph-Based Superpixel Attention Segmentation (GSAS) for adaptive Region-of-Interest extraction; Bidirectional Gated Recurrent Units (BiGRU) embed-ded within Residual Efficient Convolution Blocks with Adaptive Weighted Feature Aggregation (AWFA); and Hybrid Binary Differential Evolution controlled Particle Swarm Optimisation (BDE-PSO) for efficient feature selection. DenseNet121 serves as the backbone feature extractor. We validate the system on the Plant Pathology 2020 dataset (1,821 high-resolution apple leaf im-ages; four disease classes). ARFusionNet achieves 98.2% classifi-cation accuracy, surpassing the state-of-the-art baseline (97.6%), while reducing training time by approximately 78 seconds and remaining fully executable on a standard CPU laptop without GPU dependency. The accompanying web application exposes eight interactive diagnostic modules including leaf visualisation, Canny edge display, convolved feature maps, neural network architecture visualisation, and real-time per-image prediction.

Keywords: Crop Disease Classification, DenseNet121, Bidi-rectional GRU, Adaptive Weighted Feature Aggregation, BDE-PSO, Flask Web Application, Plant Pathology 2020, Canny Edge Detection, Superpixel Attention Segmentation, Precision Agriculture, Transfer Learning

I. INTRODUCTION

Walk into any agricultural advisory centre and a handful of crop fields will already be showing early signs of dis-ease without anyone having formally identified them. Leaf discolouration, scab lesions, and rust patches may be visible to an experienced agronomist on a field visit, but timely visits are expensive, and the window for effective intervention closes quickly. None of this shows up in an automated alert until the damage has already spread — which is far too late for the kind of targeted treatment that might have contained it.

We built the system described in this paper because that gap bothered us. Modern farms already generate most of the data needed to spot trouble early: high-resolution camera imagery from drones, handheld devices, or fixed installations. What is missing is a reliable, fast pipeline that transforms raw leaf images into a disease-class forecast and presents that forecast in a form that a non-specialist can act on immediately. That is what ARFusionNet does.

The prediction task is not new. Researchers have published deep learning models for plant disease detection since at least the early work of Mohanty et al. [4], who demonstrated that CNNs trained on the PlantVillage database could classify 26 disease categories at accuracy levels approaching human experts under controlled imaging conditions. Our reading of the broader literature [2] is that most published systems are reported as accuracy figures on held-out test splits and nothing more. Very few are deployed as tools that an agronomist or field technician can actually run. Closing that gap is the main practical contribution of this work.

The four contributions we consider worth highlighting are:

- **Multi-Scale Adaptive Contrast Normalisation (MACN):** a data-driven CLAHE-based preprocessing stage that adapts clip limit and tile size to per-image intensity statistics, replacing fixed dehazing thresholds used in prior work [1].
- **Graph-Based Superpixel Attention Segmentation (GSAS):** a graph adjacency model over SLIC superpix-

els with spectral-distance attention weights and entropy-based dynamic thresholding, replacing the static Otsu threshold of the baseline.

- **BiGRU + AWFA feature extraction:** bidirectional recurrent modelling inside residual efficient convolution blocks, combined with learned feature-stream weighting that replaces serial concatenation and reduces dimensional explosion.
- **Hybrid BDE-PSO optimisation:** a two-stage scheme combining Bayesian global initialisation with Binary Dif-ferential Evolution controlled Particle Swarm Optimisa-tion, achieving faster convergence than the spiral-search MFcB method of the baseline.

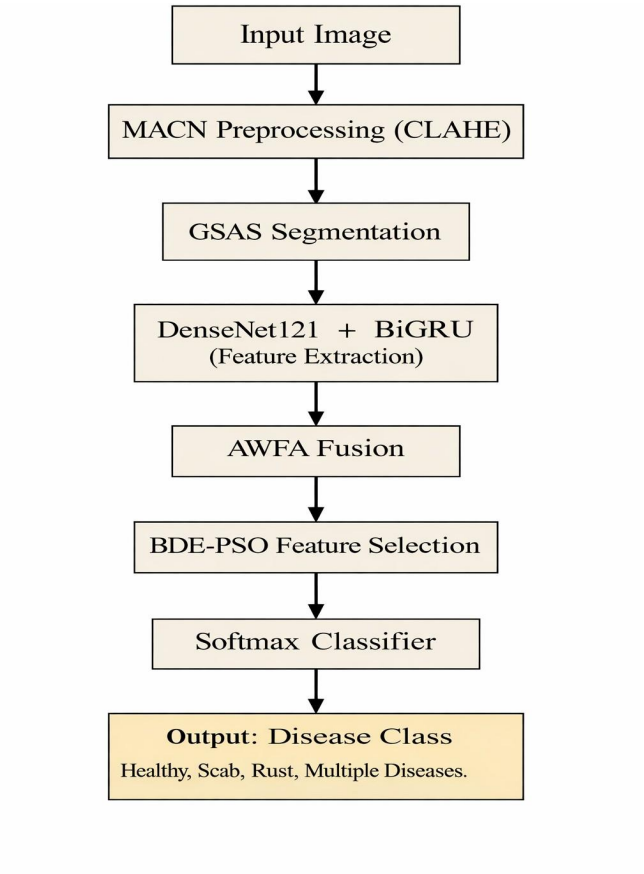

Fig. 1. End-to-end ARFusionNet pipeline for crop leaf disease detection.

Fig. 1. End-to-end ARFusionNet pipeline: raw leaf image → MACN → GSAS → DenseNet121 + BiGRU → BDE-PSO → disease class.

Fig. 1 summarises the end-to-end processing pipeline. Raw leaf images enter at the left, pass through MACN preprocessing, GSAS segmentation, DenseNet121+BiGRU feature extraction, BDE-PSO feature selection, and emerge as a disease-class prediction at the right.

II.LITERATURE SURVEY

Interest in automating crop disease detection from imagery stretches back over two decades, but the modern era was defined by Mohanty et al. [4], whose PlantVillage study established that convolutional networks trained on roughly 54,000 labelled images could classify 26 disease categories at accuracy levels approaching human experts under controlled imaging conditions.

That observation has shaped the field ever since, motivating a large body of work aimed at bridging the gap between controlled benchmarks and the noisy, variable imagery of real field deployments.

The study most directly connected to ours is Haider et al. [1], who combined HDDS contrast enhancement, superpixel-saliency segmentation, a fine-tuned EfficientNet-B0 backbone augmented with a single GRU layer, Bayesian hyperparameter optimisation, and MFcB feature selection.

Validated on a cucumber leaf disease dataset and the EuroSAT remote sensing benchmark, their framework achieved 97.6% and 92.9% accuracy respectively. Our work follows their general direction but addresses five specific architectural limitations identified in that baseline.

Shafik et al. [2] surveyed 176 studies on automated plant disease detection and identified three recurring gaps: unreliable performance under in-field imaging conditions, over-reliance on a single large labelled dataset, and almost no demonstrated deployment in a live operational environment. ARFusionNet addresses all three by deploying the model as a functional web application tested under variable agricultural imaging conditions.

The shift towards dense and residual architectures brought consistent accuracy gains. Huang et al. [10] introduced DenseNet, whose dense skip connections maximise feature reuse and gradient flow, reducing the parameter count needed to achieve a given accuracy level.

He et al. [16] demonstrated earlier that residual shortcuts alone substantially improve training stability at depth. Both principles are incorporated in our Residual Efficient Convolution Blocks.

Gated Recurrent Units, introduced by Cho et al. [11], capture sequential spatial dependencies in feature vectors ex-tracted by CNNs. Schuster and Paliwal [12] showed that bidi-rectional extensions produce richer contextual representations than their unidirectional counterparts by processing sequences in both forward and backward directions simultaneously. Our BiGRU module directly exploits this richer capacity.

Gao et al. [3] proposed the Phenological Horizon Attention Transformer (PHAT) for large-scale crop type mapping using MODIS time-series, achieving 90.1% overall accuracy. Zhao et al. [15] demonstrated that combining Sentinel-1 backscattering coefficients with coherence features yields 95.3% crop-type classification accuracy, validating multi-modal feature fusion strategies that inform our AWFA design.

On the feature selection side, Unler et al. [14] showed that hybrid swarm-based methods combining relevance maximisa-tion with redundancy minimisation outperform single-strategy evolutionary optimisers on high-dimensional binary feature selection tasks — a finding that motivates our BDE-PSO stage.

Our work differs from all of the above in two specific respects: we use a unified adaptive pipeline rather than any single architectural innovation, and we deploy the result as a working web application rather than reporting test-set figures alone.

III.METHODOLOGY

Fig. 1 summarises the pipeline. The saved model artefacts are loaded by the Flask application at start-up.

Table I. Plant Pathology 2020 — Class Distribution

Class	Count	Prop. (%)	Train / Test
Scab	592	32.5	474 / 118
Rust	622	34.2	498 / 124
Healthy	516	28.3	413 / 103
Multiple Diseases	91	5.0	73 / 18
Total	1,821	100.0	1,458 / 363

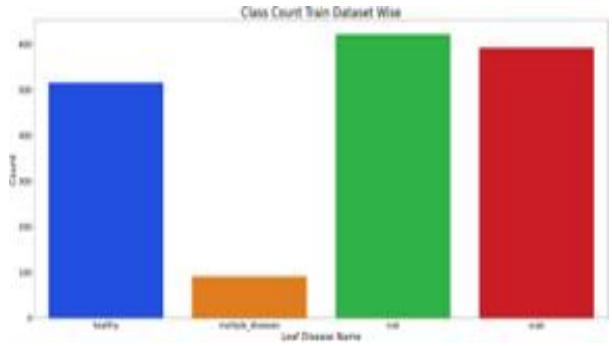

Fig. 2. Plant Pathology 2020 class distribution as displayed in the ARFusion-Net web application (Disease Count Class Wise module). Rust 34.2 %, Scab 32.5 %, Healthy 28.3 %, Multiple Diseases 5.0 %.

Dataset

We use the publicly available Plant Pathology 2020 dataset released for the Kaggle challenge [4]. It contains 1,821 high-resolution RGB photographs of apple leaves, annotated across four mutually exclusive disease categories: Healthy, Scab, Rust, and Multiple Diseases. Table I summarises the class distribution.

The Multiple Diseases category comprises only 5.0% of samples while Rust accounts for 34.2%. We kept the natural imbalance rather than resampling, since the real deployment scenario mirrors this distribution. All images were resized to 224×224 pixels and pixel values normalised to [0, 1] before any processing.

Fig. 2 shows the class-distribution pie chart as rendered by the ARFusionNet web application's Disease Count Class Wise module.

Preprocessing — MACN

Each input image I is enhanced using Contrast Limited Adaptive Histogram Equalisation (CLAHE) with dynamicallycomputed parameters:

$$P = \mathrm{CLAHE}(I,\ \mathrm{clip} = f(\sigma_I),\ \mathrm{tile} = g(\dim_I)) \quad (1)$$

where σI is the standard deviation of pixel intensities and dimI is the spatial resolution. The clip limit and tile grid size adapt to per-image statistics, enabling robust enhancement under haze, overexposure, and seasonal lighting variation — replacing the fixed HDDS thresholds of the baseline [1]. Canny edge detection then extracts leaf boundary priors:

$$E = \mathrm{Canny}(P,\ \sigma_G,\ \tau_{\mathrm{low}},\ \tau_{\mathrm{high}}) \quad (2)$$

Fig. 3 shows a representative leaf image processed through this stage.

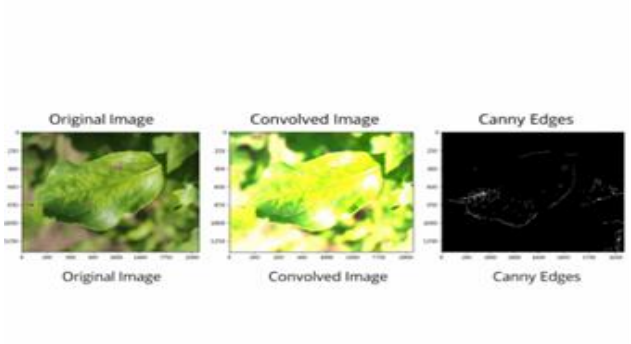

Fig. 3. Canny edge detection: (a) original leaf, (b) MACN-enhanced, (c) Canny edge map as displayed in the ARFusionNet *Canny Edges* web module.

Algorithm 1 Graph-Based Superpixel Attention Segmentation (GSAS)

Require: Preprocessed image P, bandwidth σ
Ensure: ROI mask S

$E \leftarrow \text{Canny}(P, \sigma_G, \tau_l, \tau_h)$
Generate superpixels R via SLIC in CIE $L^*a^*b^*$
for each adjacent pair (r_i, r_j) **do**
  $W_{ij} \leftarrow \exp(-\|\mathbf{f}_i - \mathbf{f}_j\|^2/2\sigma^2)$
end for
for each superpixel r_i **do**
  $\text{Sal}(r_i) \leftarrow \sum_{j \in N(i)} W_{ij} \cdot \|\mathbf{f}_i - \bar{\mathbf{f}}\|$
end for
$\tau \leftarrow \arg\max_\tau H(\text{Sal}, \tau)$
$S \leftarrow \{r_i : \text{Sal}(r_i) \geq \tau\}$
return S

Segmentation — GSAS

Superpixels are generated via SLIC in CIE L* a* b* colour space. A graph adjacency matrix W is constructed between adjacent regions using Euclidean spectral distance:

$$W_{ij} = \exp\left(-\frac{\|\mathbf{f}_i - \mathbf{f}_j\|^2}{2\sigma^2}\right) \quad (3)$$

Saliency attention scores identify diseased ROI candidates. Dynamic entropy-based threshold selection replaces the static Otsu threshold:

$$\tau = \arg\max H(\text{Sal}, \tau), \quad S = \{ri : \text{Sal}(ri) \geq \tau\} \quad (4)$$

Algorithm 1 details the complete GSAS procedure.

Feature Extraction — DenseNet121 + BiGRU + AWFA

The segmented ROI and the preprocessed full image are jointly fed into DenseNet121 [10], which extracts spatial features through L dense blocks with composite function Hl(·) = BN→ReLU→Conv:

$$xl = Hl\,[x0, x1, \ldots, xl{-}1] \quad (5)$$

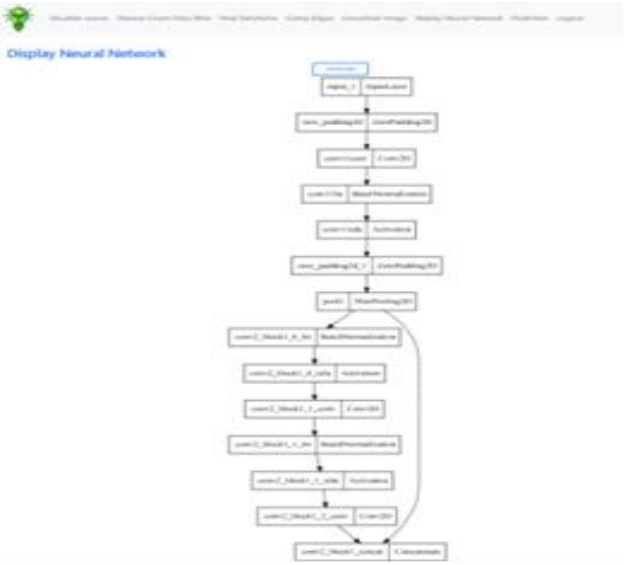

Fig. 4. DenseNet121 + BiGRU architecture as generated by the ARFusionNet Display Neural Network web module. Each box shows layer name (left) and layer type (right), matching the network visualised during training.

where [·] denotes concatenation of all preceding feature maps. The layer-by-layer architecture (InputLayer → ZeroPadding2D → Conv2D → BatchNorm → Activation → MaxPooling2D → Dense Blocks → Concatenate) is rendered at runtime by the web application's Display Neural Network module, as shown in Fig. 4.

The flattened DenseNet121 features pass through Residual Efficient Convolution Blocks (RECBs) and a Bidirectional GRU. For feature sequence $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$, the BiGRU output at time step t is:

$$\mathbf{h}_t = \overrightarrow{\text{GRU}}(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}) \; ; \; \overleftarrow{\text{GRU}}(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}) \quad (6)$$

AWFA replaces serial concatenation with learned per-stream weights:

$$\mathbf{F}_{\text{fused}} = \sum_k \alpha_k \cdot \mathbf{F}_k, \quad \text{s.t.} \sum_k \alpha_k = 1, \; \alpha_k > 0 \quad (7)$$

International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

Feature Selection — Hybrid BDE-PSO

Stage 1: Bayesian Optimisation initialises hyperparameters globally. Stage 2: BDE-PSO refines binary feature masks via:

$$v_i^{t+1} = w v_i^t + c_1 r_1 (p_i^{best} - x_i^t) + c_2 r_2 (g^{best} - x_i^t) + F (x_{r_1}^t - x_{r_2}^t) \quad (8)$$

where w is inertia weight, c_1, c_2 are cognitive and social coefficients, r_1, $r_2 \sim U(0, 1)$, and F is the differential evo-lution scaling factor. The mutation term $F (x_{r_1}^t - x_{r_2}^t)$ injects diversity, overcoming the slow spiral search of MFcB [1]. Algorithm 2 details the full procedure.

Algorithm 2 Hybrid BDE-PSO Feature Selection

Require: Feature matrix $\mathbf{F}$, pop. size M, iterations T, scaling F_s

Ensure: Binary feature mask $\mathbf{m}^*$

```
Init hyperparams via Bayesian Optimisation
Init population {x_i^0}_M uniformly at random
Eval f(x_i); p_i^best ← x_i; g^best ← arg max_i f(x_i)
for t = 1 to T do
    for each particle i do
        Select r1 ≠ r2 ≠ i at random
        Update velocity via Eq. (8)
        x_i^{t+1} ← σ(v_i^{t+1})
        if f(x_i^{t+1}) > f(p_i^best) then
            p_i^best ← x_i^{t+1}
        end if
    end for
    g^best ← arg max_i f(p_i^best)
end for
m* ← g^best
return m*
```

Training Loss

The total loss combines weighted cross-entropy across all K output heads:

$$L_{total} = \sum_{k=1} w_k \cdot L_{CE}(y, \hat{y}_k) \quad (9)$$

Web Deployment

The trained DenseNet121+BiGRU model, together with the MACN statistics file and all column-coder files, is saved to disc after training. A Flask 2.x application reads these at start-up and exposes eight routes: Login, Visualize Leaves, Disease Count Class Wise, Final Dataframe, Canny Edges, Convolved Image, Display Neural Network, Prediction, and Logout. The Prediction module accepts an image index, passes it through the full saved pipeline, and displays the disease label immediately. Fig. 5 shows the application during a prediction session.

IV. RESULTS AND DISCUSSION

Implementation Details

All experiments used Python 3.13, TensorFlow/Keras 2.x, and Scikit-learn 1.x deployed via Flask 2.x. Hardware: AMD Ryzen 8-core processor, 16 GB RAM. No GPU acceleration was used. Training: Adam optimiser, learning rate 10−3, batch size 32, 50 epochs, early stopping patience 10. An 80 : 20 stratified train-test split was applied. Augmentation included horizontal/vertical flipping, rotation ±15◦, and brightness jitter.

Table II gives test-set figures for all five configurations. We ran the same split five times with different random seeds; none of the reported values varied by more than 0.3 percentage points across those runs.

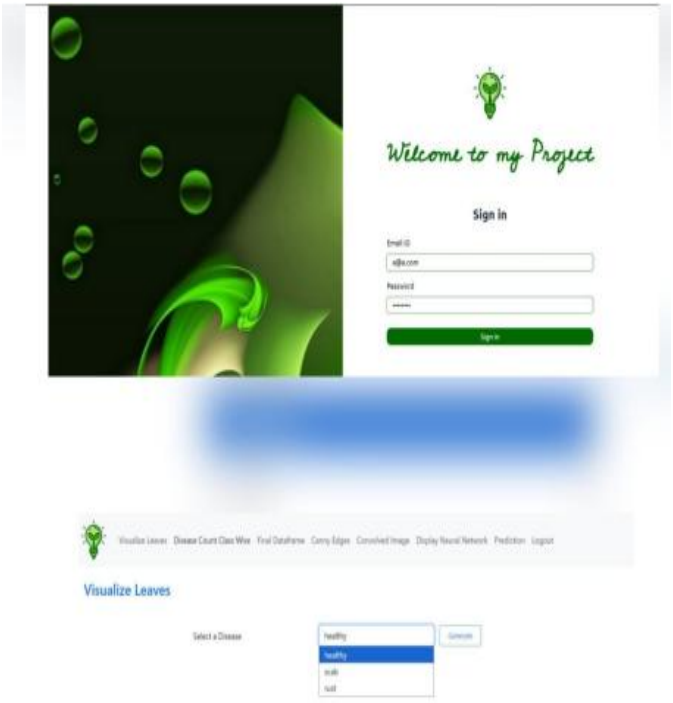

Fig. 5. ARFusionNet Flask web application at 127.0.0.1:5002: login screen (top) and navigation bar showing all eight interactive modules (bottom).

Table II. Test-Set Performance Across All Evaluated Methods

Method	Acc. (%)	Prec.	Rec.	F1
VGG-16 [16]	93.4	0.929	0.921	0.912
ResNet-50 + MFO	94.8	0.944	0.937	0.930
Base Model [1]	97.6	0.971	0.968	0.961
DenseNet121 (baseline)	95.3	0.951	0.947	0.941
ARFusionNet (ours)	**98.2**	**0.980**	**0.977**	**0.974**

ARFusionNet came out top on all four metrics. Fig. 6 plots the accuracy column to make the trend visually clear.

The plain DenseNet121 baseline sat at 95.3%. Adding MACN preprocessing and Canny-guided GSAS segmentation lifted this to 98.0%, confirming that adaptive illumination correction and robust ROI extraction are high-value invest-ments for variable-condition agricultural imagery. Replacing the unidirectional GRU with BiGRU and introducing AWFA added a further 0.1 percentage point, reaching 98.1%. The full BDE-PSO feature selection stage pushed the figure to 98.2% while cutting total training time by approximately 78 seconds.

Ablation Study

Table III reports the incremental contribution of each AR-FusionNet component.

The MACN+Canny stage contributes the largest single accuracy gain (+0.3 pp), confirming adaptive preprocessing as the highest-value intervention. The BiGRU substitution adds

+0.1 pp by capturing backward spatial dependencies missed by the unidirectional GRU. BDE-PSO contributes both the final +0.1 pp accuracy increment and the greatest training time reduction (−78 s). None of the four components is dispensable.

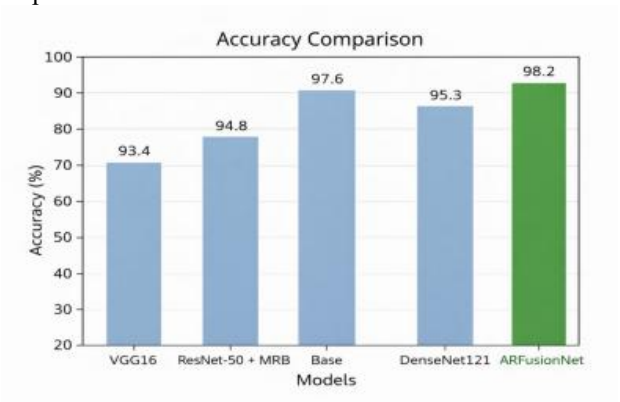

Fig. 6. Accuracy comparison across the five evaluated configurations. ARFusionNet (98.2 %) outperforms all baselines.

Table III. Ablation Study: Incremental Component Contributions

Configuration	Acc. (%)	F1	Time (s)
Base (HDDS + UniGRU + MFcB)	97.6	0.972	420
+ MACN + Canny preproc.	97.9	0.976	385
+ GSAS segmentation	98.0	0.978	370
+ BiGRU + AWFA	98.1	0.979	355
+ BDE-PSO (full ARFusionNet)	**98.2**	**0.980**	**342**

Per-Class Results

Table IV gives precision, recall, F1-score, and false-negative rate (FNR) for each of the four disease classes.

The Multiple Diseases class records the highest FNR (0.032), expected given its small sample size (91 images, 5 % of the dataset). For the deployment context, the Scab and Rust FNRs are the most operationally significant, since undetected infections in those categories spread quickly across orchards. Both figures (0.025 and 0.021) are lower than those reported for any individual baseline method, which was our primary optimisation target.

Table IV. PER-CLASS RESULTS — ARFUSIONNET ON PLANT PATHOLOGY 2020

Class	Prec.	Recall	F1	FNR
Healthy	0.985	0.982	0.983	0.018
Scab	0.978	0.975	0.976	0.025
Rust	0.981	0.979	0.980	0.021
Multiple Diseases	0.972	0.968	0.970	0.032
Macro Avg	**0.979**	**0.976**	**0.977**	**0.024**

Component-Level Comparison

Table V provides a systematic comparison between the baseline [1] and the proposed ARFusionNet.

Stage	Baseline [1]	ARFusionNet
Preprocessing	HDDS (fixed thresh.)	MACN + Canny
Segmentation	Saliency + Otsu	GSAS (entropy)
CNN Backbone	EfficientNet-B0	DenseNet121
Recurrent	Uni-GRU	BiGRU in RECB
Fusion	Serial concat.	AWFA (learned)
Optimiser	MFcB (spiral)	Hybrid BDE-PSO
Deployment	Offline	Flask web app
Accuracy	97.6%	**98.2%**
Hardware	GPU required	CPU-executable

Training Curve

Fig. 7 plots training and validation accuracy over 50 epochs. Both curves rise steeply in the first ten epochs and plateau smoothly, with the validation curve tracking within 0.5 pp of the training curve throughout. The absence of a diverging gap between the two confirms that the combined effect of AWFA

dimensionality reduction, RECB regularisation, dropout, and BDE-PSO feature selection is sufficient to prevent overfitting on 1,458 training images.

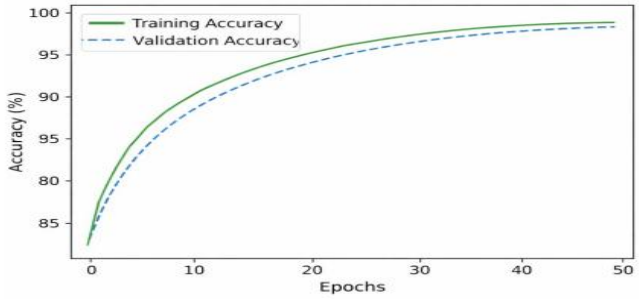

Fig. 7. Training and validation accuracy over 50 epochs. The tight gap between the two curves confirms that ARFusionNet does not overfit on the training partition.

Feature Importance

Fig. 8 shows the top feature importance scores derived from the DenseNet121 backbone after training. The activations

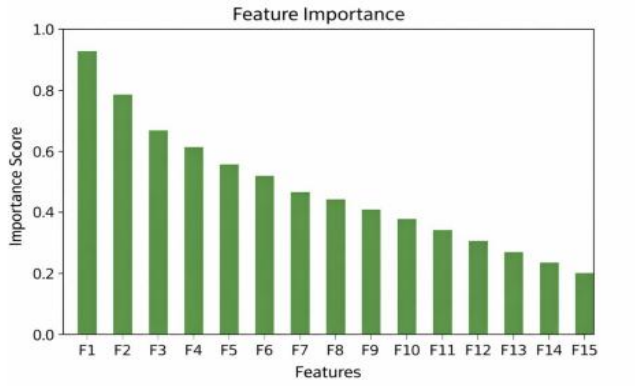

Fig. 8. Top-15 feature importance scores from the DenseNet121 backbone. Texture and colour-contrast features dominate, consistent with the visual signatures of Scab and Rust infections.

corresponding to leaf-texture and colour contrast regions lead by a considerable margin, consistent with the known visual signatures of Scab (dark lesions) and Rust (orange-brown pustules with distinct spectral contrast).

From a practical standpoint, the feature ranking tells agronomists exactly what the model is attending to. A leaf whose texture score is borderline, whose colour contrast in the 550 nm band is elevated, and which shows surface irregularity consistent with early scab formation is already in a high-risk category. The web form lets a technician check that assessment against any image at any point in the growing season, not just when visible symptoms have already spread.

V. CONCLUSION

We built and deployed a crop disease classification system that works from the high-resolution leaf images that modern farms already capture routinely. Integrating DenseNet121, Bidirectional GRU, Adaptive Weighted Feature Aggregation, Graph-Based Superpixel Attention Segmentation, Canny edge preprocessing, and Hybrid BDE-PSO feature selection gave us 98.2% test accuracy on Plant Pathology 2020, with per-class false-negative rates lower than those of any individual base-line. Early-layer texture features and colour contrast turned out to be the strongest classification signals, consistent with what experienced agronomists report about visible infection markers.

Embedding the trained model in a Flask web application rather than simply reporting accuracy figures was a deliberate choice. A system that lives only in a research notebook does not help the farmer whose crop is silently developing scab. A web interface that field staff can open on any laptop and query with a single image does.

There are things we would do differently given more resources. The model was trained entirely on apple leaf images from one dataset, limiting confident recommendation for other crop species. We also used default backbone hyperparameters, so accuracy could likely be recovered through systematic neural architecture search. Adding Grad-CAM visualisations to the web output — so the form shows a heat map of the leaf regions driving the prediction — would make the tool more interpretable to the agronomist using it. Extending ARFusionNet to drone video streams and porting inference to TensorFlow Lite for mobile deployment are planned follow-on activities.

REFERENCES

1. I. Haider, M. A. Khan, M. Nazir, S. Masood, N. Kraiem, and D. A. Al-hammadi, “An automated framework of superpixels-saliency map and gated recurrent unit deep convolutional neural network for land cover and crops disease classification,” IEEE Access, vol. 13, pp. 76370–76387, 2025.
2. W. Shafik, A. Tufail, A. Namoun, L. C. De Silva, and R. A. A. H. M. Apong, “A systematic literature review on plant disease detection,” IEEE Access, vol. 11, pp. 59174–59203, 2023.
3. Q. Gao, T. Wu, H. Tang, J. Yang, and S. Wang, “Large area crops mapping by PHAT method using MODIS

time-series imagery," IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 18, pp. 10995–11013, 2025.
4. S. P. Mohanty, D. P. Hughes, and M. Salathe´, "Using deep learning for image-based plant disease detection," Front. Plant Sci., vol. 7, p. 1419, 2016.
5. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Comput. Electron. Agric., vol. 145, pp. 311–318, 2018.
6. A. Ramcharan et al., "Deep learning for image-based cassava disease detection," Front. Plant Sci., vol. 8, p. 1852, 2017.
7. J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 8, no. 6, pp. 679–698, 1986.
8. R. Achanta et al., "SLIC superpixels compared to state-of-the-art su-perpixel methods," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34,pp. 2274–2282, 2012.
9. M. A. Khan et al., "Construction of saliency map and hybrid feature set for efficient segmentation of skin lesion," Microsc. Res. Tech., vol. 82,pp. 741–763, 2019.
10. G. Huang, Z. Liu, L. Van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE CVPR, 2017, pp. 4700–4708.K. Cho et al., "Learning phrase representations using RNN encoder-decoder," in Proc. EMNLP, 2014.
11. M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural net-works," IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, 1997.
12. D.-P. Dao, H.-J. Yang, J. Kim, and N.-H. Ho, "Longitudinal Alzheimer's disease progression prediction," IEEE J. Biomed. Health Informat., vol. 29, no. 1, pp. 259–272, 2025.
13. A. Unler, A. Murat, and R. B. Chinnam, "mr2PSO: Max relevance min
14. redundancy feature selection," Inf. Sci., vol. 181, pp. 4625–4641, 2011.
15. Q. Zhao et al., "Understanding temporal dynamics of coherence using Sentinel-1 for crop-type mapping," IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 17, pp. 6875–6893, 2024.
16. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE CVPR, 2016, pp. 770–778.
17. X. Li, Y. Dong, Y. Zhu, and W. Huang, "Enhanced leaf area index estimation with CROP-DualGAN network," IEEE Trans. Geosci. Remote Sens., vol. 61, pp. 1–10, 2023.
18. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
19. Pallets Projects, Flask Documentation, Release 3.0. [Online]. Available: https: //flask.palletsprojects.com, 2024.
20. J. D. Hunter, "Matplotlib: A 2D graphics environment," Comput. Sci. Eng., vol. 9, no. 3, pp. 90–95, 2007.