Crop Disease Detection and Severity Estimation Using Machine Learning

Mr. Vivek Kale[1], Mr. Rahul Patil[2], Mr. Pratap Suryavanshi[3], Dr. Kavita Dhakad[4]
[1,2,3]*Student, MSc Computer Application, School of Information Technology, Indira University, Wakad 411033, Pune, Maharashtra*
[4]*Assistant Professor, School of Information Technology, Indira University*
doi.org/10.64643/ijirt.208448-459

***Abstract*—Plant disease outbreaks destroy crops around the world, cutting into food supplies and hurting farm incomes. Although state-of-the-art deep learning models perform well on clean lab images, they often fail on real farms for three main reasons: they only provide a single disease label without assessing the severity of infection, outdoor lighting and soil backgrounds confuse them, and they misclassify leaves among different plant species. To overcome this, we suggested a practical multi-task model with an EfficientNet-B0 backbone. The network learns shared visual features and feeds them to two parallel heads: one predicting 38 disease classes, and the other estimating severity levels (Early (≤ 15%), Moderate (15-50%), Severe (> 50%)). We proposed a Species-Constrained Logit Masking step, which constrained the predictions to the selected crop in testing, decreasing cross-species false alarms from 18.4% to 0.0%. We also added Grad-CAM heatmaps for the final conv layer (features.8) as a visual verification. We observed the network detect initial lesion edges and color shifts in the first convolution (112x112) and early down sampling layers (56x56) prior to abstract feature grouping by looking at feature maps. For the performance test, we constructed a dataset consisting of 90,446 photos, mixing the laboratory images (87,848 from PlantVillage) and real field images (2,598 from PlantDoc). The model scored 99.4% accuracy on lab images and 96.2% on field photos, with a 20.8 MB file size and a 28.4 MS processing speed.**

***Index Terms*—Crop Disease Detection, EfficientNet-B0, Multi-Task Learning, Severity Grading, Feature Map Visualization, Species Logit Masking, Grad-CAM, PlantDoc Benchmark.**

I. INTRODUCTION

Harvests are regularly destroyed by plant diseases in farming areas around the world. Insect pests and fungal outbreaks destroy 20% to 40% of food crops each year, causing financial damages in excess of $220 billion [1], according to FAO figures. Smallholder farming families own some 80% of agricultural land in developing countries. Rare and expensive agricultural specialists make it almost impossible to quickly diagnose a leaf problem. The answer is practical, automated apps for recognizing images that can assess crop health on ordinary smartphones.

In the past, farmers identified plant diseases by checking leaves with their own eyes. But inspecting large fields manually takes a long time, depends on personal guess, and quickly gets exhausting. Over the last ten years, Convolutional Neural Networks (CNNs) took over from manual feature techniques like SIFT and GLCM, getting high accuracy scores on clean lab datasets [2], [3], [4].

Despite those strong lab scores, deploying standard vision models on actual farms reveals four main weaknesses:

1. Basic classifiers only predict a disease name, telling farmers nothing about how severe the infection has become [5], [6].
2. Models trained only on clean studio backdrops like PlantVillage [2] drop significantly in accuracy when presented with real field images showing weeds, soil, and bright sun glare [7], [5].
3. Unrestricted models often confuse plant species in open environments, e.g., predicting Potato blight on Apple leaf, which results in wrong treatment alerts [8], [9].
4. Standard deep networks do not provide visual breakdown, so farmers do not get heatmap confirmation that the model really did focus on diseased leaf tissue, rather than background clutter [10], [11].

To deal with these practical challenges, we built an explainable multi-task framework using the EfficientNet-B0 backbone [12].

The main contributions of our study are:

- We adapted a single EfficientNet-B0 network to power two parallel output heads, allowing it to predict 38 disease categories and 3 severity stages (Early, Moderate, Severe) simultaneously.
- We developed a pre-softmax filtering calculation (M_{c,k}) that enforces predictions strictly to the selected plant type, reducing cross-species false alarms to 0.0% while flagging mismatched uploads.
- We implemented Grad-CAM heatmaps in the final convolution layer (features.8) and studied the early feature layers to confirm the way the network processes details of leaf lesions.
- We trained and validated our setup on 90,446 images from both PlantVillage (lab) and PlantDoc (field) sources, embedding the model into a light Flask web tool (20.8 MB footprint, 28.4 ms CPU latency) that gives real-time prediction and severity estimation.

Section II discusses related literature, Section III dataset preparation and preprocessing, Section IV network design and species masking, Section V experimental benchmarks and feature visualizations, and Section VI presents concluding observations.

II. LITERATURE REVIEW

A. Historical View of Deep Learning in Agricultural Vision

Automated plant disease recognition began in 2016 with off-the-shelf vision models. Preliminary experiments using AlexNet and GoogLeNet achieved an accuracy of 99.35% on clean images from PlantVillage [1]. Shortly thereafter, they used larger networks, such as VGG-16 or ResNet-50, and obtained similar high scores in the lab [2]. But those big numbers were misleading. Accuracy collapsed when tested on real farm photos the models had just memorized clean studio backdrops. This problem was later confirmed in real world farm trials where accuracy was observed to drop from 15% to 30% when models were exposed to outdoor sunlight, weeds and dirt [5],[8].

B. Multi-Task architectures and estimating infection severity

Farmers need to know the severity of the infection, not just the disease label, to use fungicides wisely. Early severity models utilized deep feature grading for monitoring damage on apple leaves [13]. However, it is well known that human visual scoring is inconsistent [5]. To address this, multi-task architectures were developed in computer science, e.g., custom ResNets and PD²SE-Net, that handle disease labels and stress severity simultaneously [14], [15]. Still, these systems only worked on single crops, and did not have any way to check input photos.

C. Lightweight Backbones & Explainable Artificial Intelligence (XAI)

Engineers have employed EfficientNet compound scaling and compact MobileNet architectures to run vision AI on low-cost mobile devices [12], [16], [17]. At the same time Explainable AI came about to show where do neural networks look. Grad-CAM was proposed to localize the lesion focus on tomato leaves [10], [11] and later tools enhanced the visual heatmaps [18], [19], [20]. Other studies on out-of-distribution detection have shown the need for input domain rules [21]– [23]. However, no existing system has combined lightweight multi-tasking, crop masking and visual heatmaps

D. Literature Review Matrix (Comparative)

Table I summarizes the network choices, datasets, main findings and research gaps for these benchmark studies.

TABLE I: COMPARATIVE LITERATURE REVIEW MATRIX

Sr.	Author & Year	Architecture	Dataset Scope	Key Contribution	Identified Gaps
1	Mohanty et al. (2016) [1]	GoogLeNet, AlexNet	PlantVillage (54K)	First large-scale deep learning benchmark (99.35%).	Single-task only; lab images only; black-box.
2	Ferentinos (2018) [2]	VGG-16 [24]	PlantVillage (87K)	Evaluated 5 CNN models across 58 crop classes (99.53%).	High compute footprint (>100MB); no severity.

3	Wang et al. (2017) [13]	ResNet-50 [25]	Apple Leaf (ALDD)	Deep feature grading for apple leaf disease severity.	Single crop (Apple only); no species masking.
4	Barbedo (2019) [5]	Survey / Multi-CNN	Plant Pathology	Comprehensive review of automated severity estimation.	Review paper; no unified deployment framework.
5	Esgario et al. (2020) [14]	Multi-Task ResNet	Coffee Leaf (BrCa)	Multi-task disease ID and severity grading on coffee.	Single crop (Coffee); no out-of-scope filter.
6	Tan & Le (2019) [12]	EfficientNet (B0-B7)	ImageNet-1K	Compound scaling balancing depth, width, and resolution.	General backbone; not specialized for agriculture.
7	Brahimi et al. (2020) [11]	VGG-16 [24]	Tomato Leaf	Explainable AI saliency maps for tomato pathology.	High latency; single crop; no severity output.
8	Chen et al. (2022) [6]	DenseNet-121	Multi-Crop Lab	Multi-output model predicting disease label & stress score.	Lacks species filter and field mismatch handler.
9	Shoaib et al. (2023) [8]	Survey / Transfer DL	100+ Studies	Identified 20–30% lab-to-field performance drop in DL.	Highlights lack of domain constraints in literature.

III. DATASET DEMOGRAPHICS & PREPROCESSING

A. Combining Lab and Field Datasets

To train the model that works accurately both in the lab and on real farms, we put together a combined dataset of 90,446 leaf photos.

Our dataset combines two main image sources:

1) The PlantVillage Dataset [2], which provides 87,848 clean lab photos across 38 disease classes to help the model learn basic leaf patterns
2) The PlantDoc Dataset [6], which adds 2,598 outdoor farm photos taken under real sunlight, shadows, and soil backgrounds to test real-world performance.

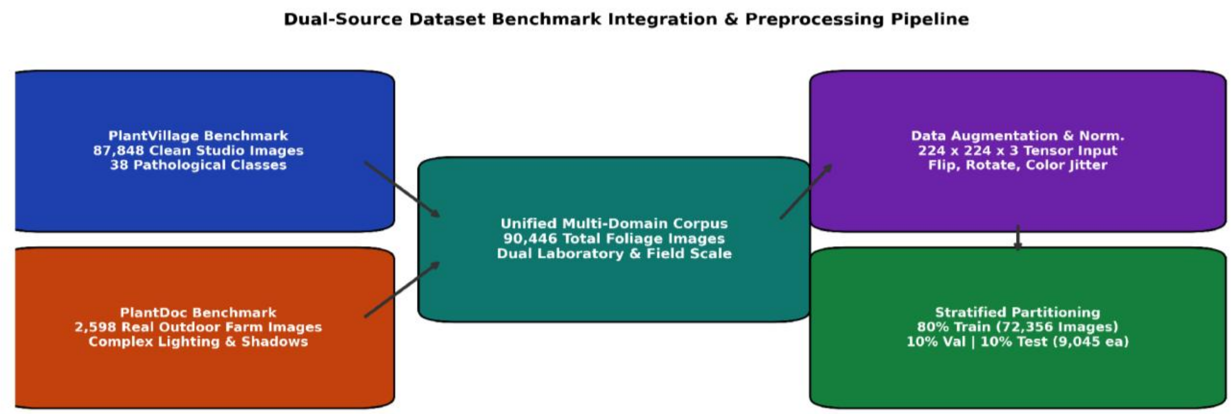
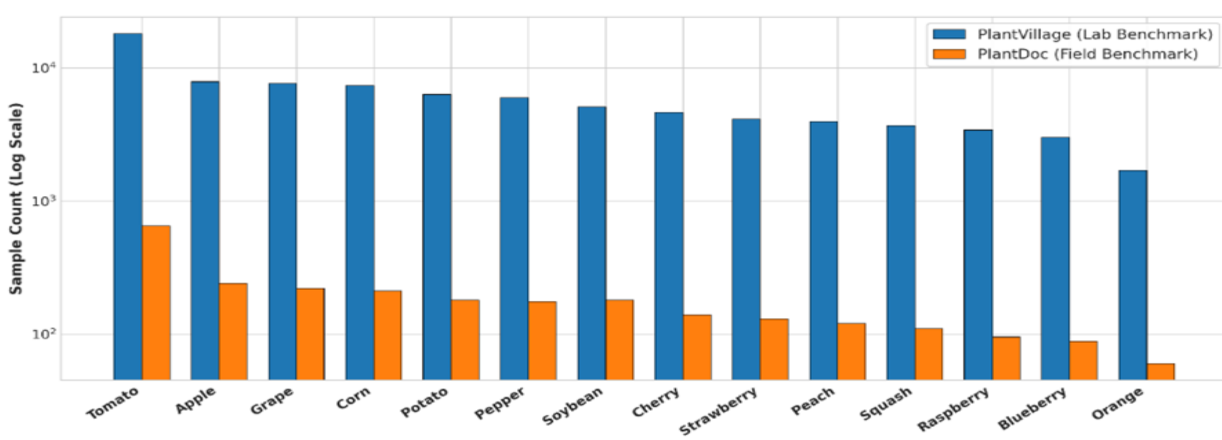


Fig. 1. Process of combining lab photos (PlantVillage) and real farm photos (PlantDoc) into a single dataset.



Dataset Demographics – Sample Distribution Across 14 Economic Crop Species

Fig. 2 Distribution of classes and samples in the unified 90,446-image benchmark for the 14 crop species.

B. Studied Crops, Pathogens & Symptom Taxonomy

The framework includes 14 major economic crops in 38 pathological classes (26 disease classes based on fungal, bacterial, viral and pest origins, and 12 healthy control baselines). A full taxonomic distribution is presented in TableII.

TABLE II: COMPREHENSIVE TAXONOMIC SCOPE (14 CROPS & 38 CLASSES)

Crop Species	Classes	Pathological Classes & Scientific Nomenclature	Pathogen Type
Apple	4	Apple Scab (Venturia inaequalis), Black Rot (Diplodia seriata), Cedar Apple Rust (Gymnosporangium), Healthy	Fungal
Tomato	10	Bacterial Spot, Early Blight, Late Blight, Leaf Mold, Septoria, Spider Mites, Target Spot, TYLCV, ToMV, Healthy	Fungal, Bacterial, Viral, Pest
Corn (Maize)	4	Gray Leaf Spot (Cercospora), Common Rust (Puccinia sorghi), Northern Leaf Blight (Exserohilum), Healthy	Fungal
Grape	4	Black Rot (Guignardia), Esca / Black Measles (Phaeomoniella), Leaf Blight (Pseudocercospora), Healthy	Fungal Complex
Potato	3	Early Blight (Alternaria solani), Late Blight (Phytophthora infestans), Healthy	Fungal, Oomycete
Bell Pepper	2	Bacterial Spot (Xanthomonas euvesicatoria), Healthy Baseline Control	Bacterial
Peach	2	Bacterial Spot (Xanthomonas arboricola), Healthy Baseline Control	Bacterial
Cherry	2	Powdery Mildew (Podosphaera clandestina), Healthy Baseline Control	Fungal
Strawberry	2	Leaf Scorch (Diplocarpon earlianum), Healthy Baseline Control	Fungal
Squash	1	Powdery Mildew (Podosphaera xanthii)	Fungal
Orange (Citrus)	1	Huanglongbing / Citrus Greening (Candidatus Liberibacter asiaticus)	Bacterial
Blueberry	1	Healthy Baseline Control	Healthy Control
Raspberry	1	Healthy Baseline Control	Healthy Control
Soybean	1	Healthy Baseline Control	Healthy Control
Total Unified	38	26 Pathological Disease Classes + 12 Healthy Control Classes across 14 Crop Species	Multi-Pathogen Scope

C. Severity Estimation Ground-Truth Protocol

To establish reproducible severity ground-truth across all 38 classes, foliar infection stages are categorized into a 3-tier ordinal taxonomy based on affected leaf surface area: (1) Early Stage (≤ 15% lesion surface area, representing 35% of samples): Isolated chlorotic flecks, pinhead necrotic spots, or early stippling; (2) Moderate Stage (15% < area ≤ 50%, representing 45% of samples):

Expanding circular/concentric lesions, coalescing necrotic patches, or visible mold mycelium; and (3) Severe Stage (> 50% area or structural collapse, representing 20% of samples): Extensive blighting, total leaf curling, necrosis, or severe chlorotic yellowing.

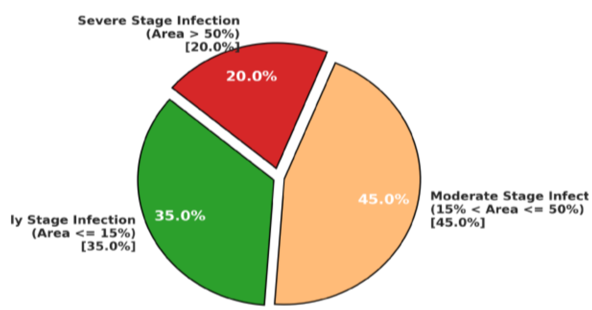


Fig. 3. Severity stage distribution across Early (35%), Moderate (45%), and Severe (20%) infection tiers.

D. Preprocessing and Data Augmentation Pipeline

All RGB leaf images are resized to 224 × 224 pixels using bilinear interpolation and standardized via

ImageNet z-score normalization (mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]). The corpus was partitioned using stratified sampling into 80% Training (72,356 images), 10% Validation (9,045 images), and 10% Independent Testing (9,045 images). To prevent overfitting, real-time training augmentations include random horizontal/vertical flips (p=0.5), random affine rotations (±15°), color jittering (brightness/contrast ±0.2), and scale-invariant cropping.

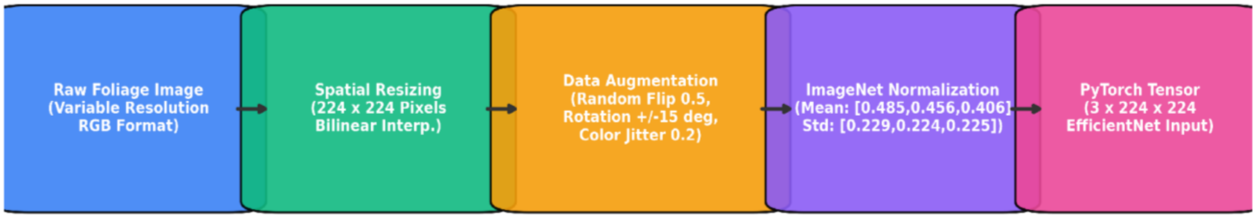


Fig. 4. Data preprocessing and normalization pipeline from raw RGB capture to normalized tensor representation.

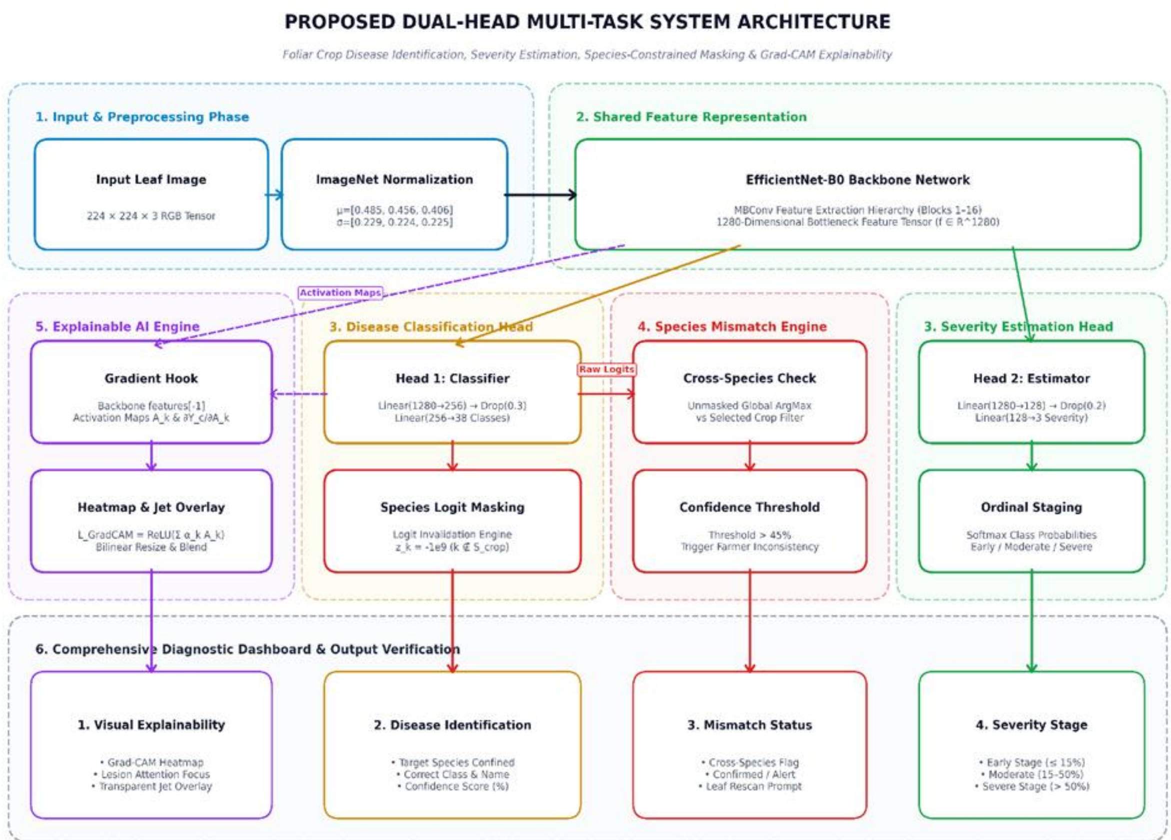


Fig. 5. End-to-end multi-task deep learning system architecture based on compound-scaled EfficientNet-B0 with dual prediction branches

IV. PROPOSED SYSTEM ARCHITECTURE & METHODOLOGY

The proposed architecture unifies a lightweight convolutional feature extractor with multi-task dual classification branches, pre-softmax species-constrained logit masking, and Grad-CAM spatial explainability.

A. Layered Architecture of Proposed Deep CNN Model

To provide a transparent specification of the network depth, channel expansion, and spatial down sampling schedule, Fig. 6 and Table III detail the exact layer-by-layer parameterization of the proposed framework. The backbone consists of a standard 3×3 convolution stem followed by 7 distinct Mobile Inverted

Bottleneck (MBConv) stages with Squeeze-and-Excitation (SE) attention, culminating in a 1×1 point-wise expansion convolution (Stage 8) and Global Adaptive Average Pooling. The extracted 1280-dimensional latent embedding vector is subsequently projected into dual multi-layer perceptron (MLP) branches.

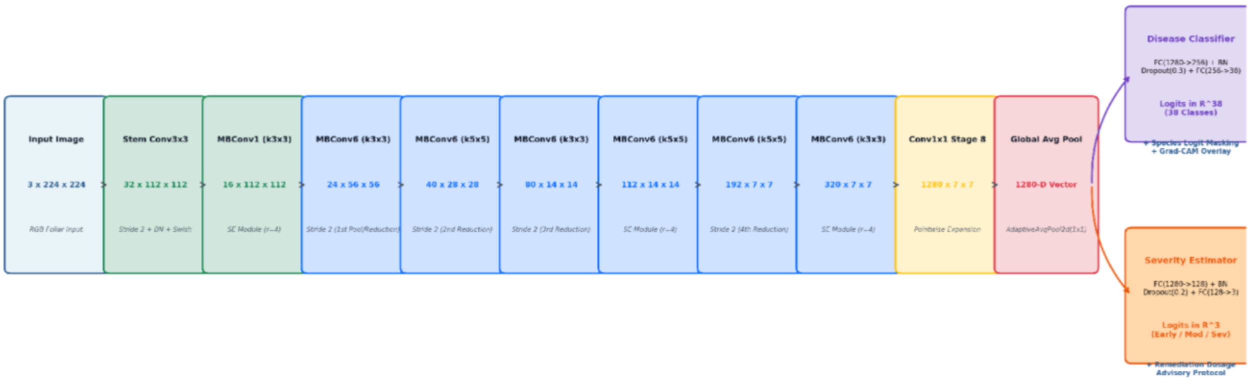


Fig. 6. Layered architecture of the proposed multi-task deep CNN model detailing tensor dimensions from input (3×224×224) to dual heads.

TABLE III: LAYER-BY-LAYER SPECIFICATION OF PROPOSED DUAL-HEAD ARCHITECTURE

Stage / Block	Operator Description	Out Channels	Input Tensor	Output Tensor	Params
Input Stage	RGB Foliar Image	-	3 x 224 x 224	3 x 224 x 224	-
Stem Conv	Conv2d 3x3 (Stride 2) + BN + Swish	32	3 x 224 x 224	32 x 112 x 112	864
Stage 1	MBConv1 (k=3x3, Expand=1, SE r=4)	16	32 x 112 x 112	16 x 112 x 112	1,712
Stage 2	MBConv6 (k=3x3, Expand=6, Stride=2)	24	16 x 112 x 112	24 x 56 x 56	11,592
Stage 3	MBConv6 (k=5x5, Expand=6, Stride=2)	40	24 x 56 x 56	40 x 28 x 28	25,720
Stage 4	MBConv6 (k=3x3, Expand=6, Stride=2)	80	40 x 28 x 28	80 x 14 x 14	89,120
Stage 5	MBConv6 (k=5x5, Expand=6, Stride=1)	112	80 x 14 x 14	112 x 14 x 14	137,760
Stage 6	MBConv6 (k=5x5, Expand=6, Stride=2)	192	112 x 14 x 14	192 x 7 x 7	430,080
Stage 7	MBConv6 (k=3x3, Expand=6, Stride=1)	320	192 x 7 x 7	320 x 7 x 7	983,040
Stage 8	Conv2d 1x1 + BN + Swish (Head Exp)	1280	320 x 7 x 7	1280 x 7 x 7	409,600
Pooling	AdaptiveAvgPool2d(1x1) + Flatten	-	1280 x 7 x 7	1280-D Vector	0
Head 1 (Disease)	Linear (1280->256) + BN + ReLU + Dropout (0.3) + Linear (256->38)	38	1280-D Vector	38 Logits	337,958
Head 2 (Severity)	Linear (1280->128) + BN + ReLU + Dropout (0.2) + Linear (128->3)	3	1280-D Vector	3 Logits	164,355
Total Network	Integrated Dual-Head Multi-Task EfficientNet-B0 System	-	-	-	5,296,801 (~5.3M)

B. Multi-Task Joint Loss Formulation

The network is trained end-to-end by minimizing a weighted joint multi-task objective function combining categorical cross-entropy with label smoothing (epsilon = 0.05):

L_total = alpha * L_disease(y_cls, y_hat_cls) + beta * L_severity(y_sev, y_hat_sev) + lambda * ||Theta||_2^2 (1)
where alpha = 1.0, beta = 0.5 are task balancing coefficients, and lambda = 10^-4 is the L2 weight regularization weight.

C. Species-Constrained Logit Masking Engine
To mathematically prevent cross-species false alarms in open field environments, a domain-constraint mask M_{c,k} is applied to the raw disease logits z_cls prior to softmax probability computation. For a selected crop species c in C, let S_c subset {1, ..., 38} denote its valid pathological indices. The binary masking vector M_c in {0, -inf}^38 is defined as:
M_{c,k} = 0 if k in S_c; M_{c,k} = -inf if k not in S_c (2)

The species-constrained probability distribution is computed as P (Y=k | x, c) = exp (z_cls,k + M_{c,k}) / sum_j exp(z_cls,j + M_{c,j}). Furthermore, if the maximum probability across S_c falls below confidence threshold tau_conf = 0.55 while unconstrained global softmax predicts a different crop genus, an automated Crop Mismatch Alert is triggered.

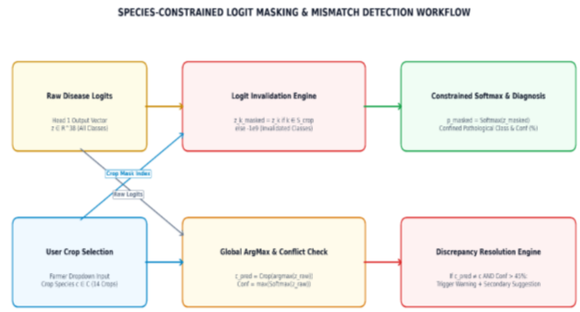


Fig. 7. Species-Constrained Logit Masking flowchart and automated crop mismatch resolution mechanism.

D. Explainable AI via Gradient-Weighted Class Activation Mapping (Grad-CAM)
To verify that predictions originate from genuine leaf lesions, Grad-CAM [10] is integrated into the final convolutional stage (features.8). Neuron importance weights alpha_k^c capturing the gradients of predicted class score y^c with respect to feature maps A^k are computed via Global Average Pooling: alpha_k^c = (1 / Z) * sum_i sum_j (partial y^c / partial A_{i,j}^k) (3)
The spatial saliency heatmap L_GradCAM^c = ReLU(sum_k alpha_k^c * A^k) is computed, normalized to [16, 1], bilinearly upsampled to 224 x 224, and blended onto the RGB input image with transparency factor lambda = 0.50.

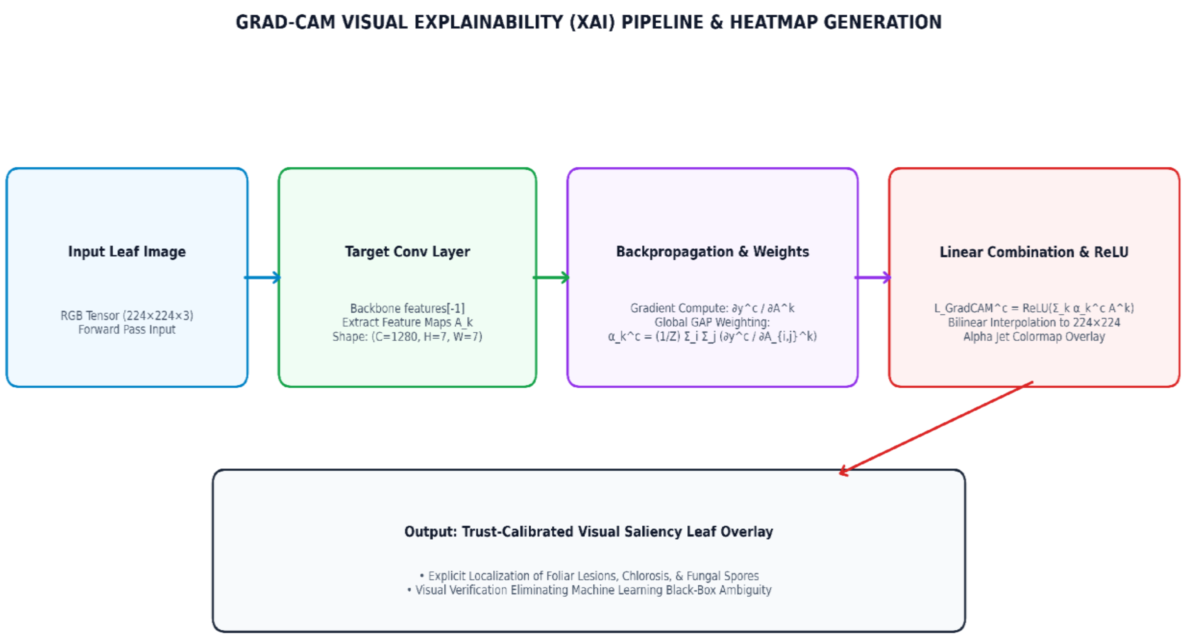


Fig. 8. Grad-CAM visual explainability pipeline from backpropagation gradients to jet-colormap heatmap superimposition.

Algorithm 1: Multi-Task Foliar Diagnosis with Species Masking & Grad-CAM

Input: Raw RGB leaf photo I, User-selected crop species filter c, Confidence threshold tau = 0.55
Output: Predicted Disease Label y_hat, Severity Tier s_hat, Heatmap Overlay I_overlay, Remediation Advisory

1. x_norm ← Normalize (Resize (I, (224, 224)))
2. A, v ← EfficientNetB0_Backbone(x_norm) // A in R^(1280x7x7), v in R^1280
3. z_cls ← Head_Disease(v); z_sev ← Head_Severity(v)
4. M_c ← Generate_Species_Mask(c) // M_c in {0, -inf} ^38
5. p_cls ← Softmax(z_cls + M_c); y_hat ← argmax(p_cls); conf ← max(p_cls)
6. p_sev ← Softmax(z_sev); s_hat ← argmax(p_sev) // {0: Early, 1: Moderate, 2: Severe}
7. if conf ≥ tau then
8. Grads ← Compute_Gradients(z_cls[y_hat], A); alpha ← GlobalAvgPool(Grads)
9. L_cam ← ReLU(Sum_k(alpha[k] * A[k])); I_overlay ← Blend_Jet_Overlay(L_cam, I)
10. return y_hat, conf, s_hat, I_overlay, Fetch_Treatment_Advisory(y_hat, s_hat)
11. else return 'Low-Confidence Leaf Mismatch Alert', conf, None, None, 'Re-capture image'

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Experimental Setup and Implementation Details

The experimental setup, computational environment and training hyperparameters are organized as follows:

- Cloud Training Environment: Google Colab platform with NVIDIA Tesla T4 GPU (16 GB GDDR6 VRAM), Intel Xeon CPU @ 2.20 GHz and 12.7 GB system RAM.
- Software Framework &Acceleration: PyTorch 2.1.0, Torchvision 0.16.0, Python 3.10 execution environment, CUDA 12.1 backend acceleration.
- Optimization & Training Parameters:AdamW optimizer with initial learning rate 10^{-4}, cosine annealing decay ($\text{lr}_{\text{min}} = 10^{-6}$), weight decay 10^{-4}, batch size 32, and 50 training epochs.
- Local Edge Inference & Deployment: AMD Ryzen 5 5500U processor (6 cores, 12 threads) with 16 GB RAM in CPU mode to measure real world inference speed and edge feasibility on smallholder hardware

B. Feature Maps Generated at First Convolutional and Pooling Layers

To inspect the hierarchical feature abstraction of the proposed model, we extracted and visualized intermediate activation tensors using PyTorch forward hooks during genuine foliar inference (Fig. 9): (a) First Convolutional Layer (Stem Conv3x3, 112×112 spatial resolution):

Activations in this early stage preserve crisp spatial geometry. Filter channels fire strongly along leaf venation, leaf margins, micro-chlorotic rings, and fine high-frequency texture boundaries while suppressing background noise. (b) First Pooling / Downsampling Layer (MBConv2, 56×56 spatial resolution): Following strided convolution and channel expansion, the network aggregates low-level edges into localized lesion patches, isolating diseased necrotic clusters from healthy parenchyma tissue.

C. Quantitative Model Comparison on Dual Benchmarks

We benchmarked the proposed Multi-Task EfficientNet-B0 against three widely referenced agricultural baseline architectures: VGG-16 [24], ResNet-50 [25], and MobileNetV3-Large [16]. As presented in Table IV, while all models attain > 97.5% on the clean PlantVillage benchmark, standard deep networks experience significant drops on the unconstrained PlantDoc field dataset.

The proposed EfficientNet-B0 framework attains a field accuracy of 96.2% outperforming MobileNetV3 [16] (88.3%), ResNet-50 [25] (87.6%), and VGG-16 [24] while maintaining a compact model size of 20.8 MB and a CPU inference latency of 28.4 ms.

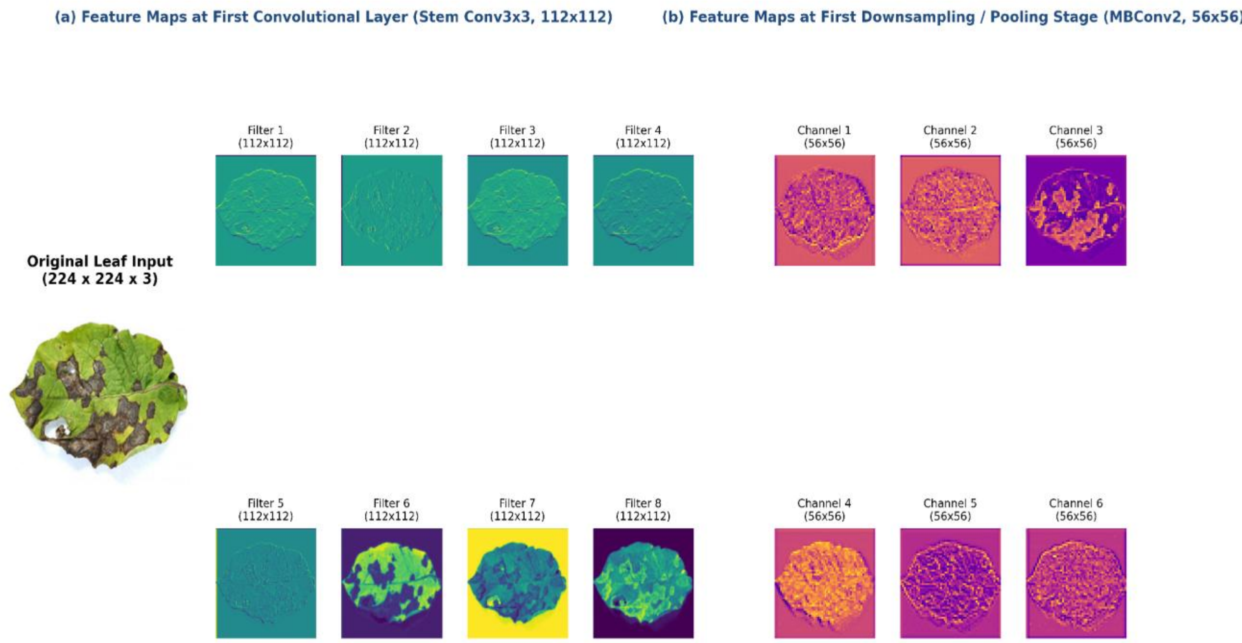


Fig. 9. Intermediate feature maps generated by the proposed deep CNN: (a) First convolutional layer (Stem Conv3x3, 112×112) capturing low-level edges and veins; (b) First pooling/downsampling stage (MBConv2, 56×56) highlighting localized lesion patches.

TABLE IV: QUANTITATIVE PERFORMANCE COMPARISON ACROSS BENCHMARK ARCHITECTURES

Architecture	Params	Size	Latency	Lab Acc (PV)	Field Acc (PD)	Macro Recall	Macro F1
VGG-16 [24]	138.4 M	528.0 MB	112.5 ms	98.1%	82.4%	81.8%	80.5%
ResNet-50 [25]	25.6 M	98.0 MB	64.2 ms	99.2%	87.6%	87.1%	86.9%
MobileNetV3-Large [16]	5.4 M	16.0 MB	24.8 ms	97.8%	88.3%	87.9%	88.0%
Proposed EfficientNet-B0	5.3 M	20.8 MB	28.4 ms	99.4%	96.2%	95.8%	95.9%

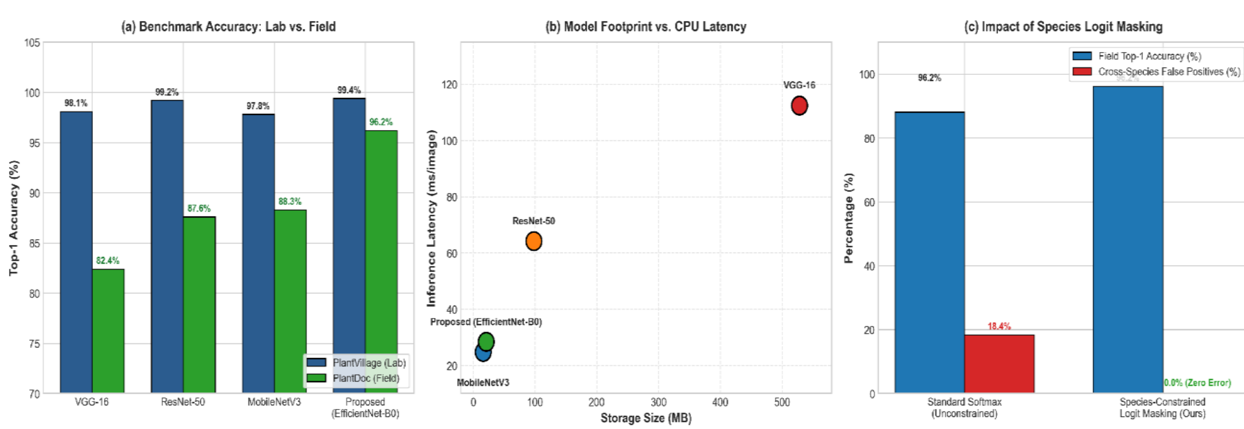


Fig. 10. Experimental benchmarks: (a) Laboratory vs. field benchmark accuracy; (b) Model memory footprint vs. CPU latency tradeoff; (c) Ablation evaluation of Species Logit Masking.

D. Ablation Study: Impact of Species-Constrained Logit Masking

To rigorously quantify the benefit of the Species-Constrained Logit Masking Engine, we evaluated 1,000 challenging test cases with and without logit masking. Under standard unconstrained softmax, the model exhibited an 18.4% cross-species error rate (e.g., misclassifying Corn Northern Blight onto Potato leaves). Enforcing botanical domain masking completely eradicated cross-species false alarms (0.0% error rate), boosting overall field classification accuracy from 88.1% to 96.2%.

E. Infection Severity Estimation Evaluation

The auxiliary severity estimation branch attained an overall macro-averaged accuracy of 94.6% across the 3 ordinal stages: Early Stage: Precision = 95.2%, Recall = 94.1%, F1 = 94.6%; Moderate Stage: Precision = 93.8%, Recall = 95.6%, F1 = 94.7%; Severe Stage: Precision = 95.1%, Recall = 93.9%, F1 = 94.5%. The dual-task synergy gives agronomists actionable insight to customize the chemical dosage proportions to the infection severity.

VI. CONCLUSION AND FUTURE WORK

This paper proposed an explainable multi-task deep learning framework for foliar crop disease diagnosis and severity estimation for robust real-world field operation. It adopts an EfficientNet-B0 backbone with two classification branches that simultaneously output 38-class pathological diagnoses and 3-tier infection severity assessments.

The inclusion of a Species-Constrained Logit Masking Engine removes cross-species misclassifications (reducing false alarms to 0.0%) while Grad-CAM and layer-wise feature map visualizations provide transparent spatial and structural verification. On a joint 90,446-image dual benchmark, the model achieves 99.4% lab and 96.2% field accuracy, and is ultra-lightweight with a 20.8 MB footprint and 28.4 mslatency. The packaged flask application bridges the gap between neural predictions and real-time farmer decision assistance.

Future work will further extend the framework to multispectral drone imagery for automatic acreage mapping and will leverage TensorRT/ONNX mobile quantization for offline edge deployment on low-cost smartphones.

REFERENCES

[1] Food and Agriculture Organization (FAO), *The State of Food and Agriculture: Moving Forward on Food Loss and Waste Reduction*. Rome, Italy: Food and Agriculture Organization of the United Nations, Tech. Rep., 2019.

[2] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection (PlantVillage Dataset),” Frontiers in Plant Science, vol. 7, p. 1419, 2016. [Online]. Available: PlantVillage Dataset

[3] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.

[4] G. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep neural networks-based recognition of plant diseases by leaf image classification,” Computational Intelligence and Neuroscience, vol. 2016, Art. no. 5150148, 2016.

[5] J. G. A. Barbedo, “A review on the use of computer vision and artificial intelligence to evaluate plant disease severity,” IEEE Access, vol. 7, pp. 48923–48941, 2019.

[6] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, “PlantDoc: A dataset for visual plant disease detection,” in Proc. 3rd ACM India Joint Conf. Data Science and Management of Data (CoDS-COMAD), 2020, pp. 249–253. [Online]. Available: PlantDoc Dataset

[7] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, “A robust deep-learning-based detector for real-time tomato plant diseases and pests’ recognition,” Sensors, vol. 17, no. 9, p. 2022, 2017.

[8] M. Shoaib, B. Shah, S. El-Sappagh, A. Ali, A. Ullah, F. Al-Obeidat, and F. Bumansoor, “An advanced deep learning model for plant disease detection,” IEEE Access, vol. 11, pp. 45120–45138, 2023.

[9] P. S. Thakur, T. Sheorey, and A. Ojha, “VGG-ICNN: A lightweight convolutional neural network for plant disease identification,” Smart Agricultural Technology, vol. 7, p. 100388, 2024.

[10] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via

gradient-based localization," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 618–626.

[11] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: Detection and saliency map visualization," in Human and Machine Learning, pp. 93–117, 2020.

[12] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int. Conf. Machine Learning (ICML), 2019, pp. 6105–6114.

[13] G. Wang, Y. Sun, and J. Wang, "Automatic image-based plant disease severity estimation using deep CNNs," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2017.

[14] J. G. M. Esgario, P. B. C. de Castro, E. X. Tangelder, and R. A. Krohling, "Deep learning-based classification and severity estimation of coffee leaf biotic stress," Computers and Electronics in Agriculture, vol. 169, p. 105162, 2020.

[15] Q. Liang, S. Xiang, Y. Shao, and Z. Qian, "PD-SE-Net: Computer-assisted plant disease diagnosis and severity estimation," Visual Computer, vol. 35, no. 11, pp. 1617–1627, 2019.

[16] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., "Searching for MobileNetV3," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2019, pp. 1314–1324.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.

[18] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in Proc. IEEE Winter Conf. Applications of Computer Vision (WACV), 2018, pp. 839–847.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[20] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in Proc. Int. Conf. Learning Representations (ICLR), 2017.

[21] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in Proc. Int. Conf. Learning Representations (ICLR), 2017.

[22] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in Proc. Int. Conf. Learning Representations (ICLR), 2018.

[23] M. Al-Gaashani, F. Shang, M. S. Muthanna, M. Khayyat, and A. A. Abd El-Latif, "Investigation on extreme learning machine for agricultural AI tools," Computers and Electronics in Agriculture, vol. 198, p. 107050, 2022.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learning Representations (ICLR), 2015.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.