© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714193

AgroMind Grow Plant Disease Detection

MAYANK YADAV[1], SHANTU DHAMI[2], GOPAL JI[3], SOURAV[4], MOHAMMAD HARIS[5]
[1, 2, 3, 4, 5]KCC Institute of Technology and Management Greater Noida, India

Abstract—Agriculture is foundational to livelihoods and national economies, particularly in India. This paper presents AgroMind Grow, an end-to-end smart agriculture platform that consolidates weather intelligence, market analytics, crop calendar and planning, AI-powered plant disease detection and guidance, equipment tracking, expert consultation, farm planning, government scheme access, and an educational knowledge base into a unified web system. We focus on a deployable plant disease subsystem that enhances practical performance without retraining the base model by combining: crop pre-selection, class-space filtering of logits, test-time augmentation (TTA), aggressive but bounded confidence boosting, a rule-based generic status detector (healthy, chlorosis, fungal rot, powdery mildew), and a disease knowledge base covering 38 classes with symptoms, causes, and treatments (chemical, organic, prevention). Using EfficientNet-B2 (260×260), we report 99.74% validation accuracy (PlantVillage). In deployment, crop-aware post-processing and knowledge integration improve perceived correctness, interpretability, and decision readiness. Platform-level benefits include potential increases in farmer income (up to 25%), operational cost reduction (40%), and risk mitigation (50%), contingent on adoption and local context.

Index Terms—Smart Agriculture, Plant Disease Detection, EfficientNet-B1, Confidence Calibration, Test-Time Augmentation, Knowledge Base, FastAPI, React.

I. INTRODUCTION

Feeding a projected 9.7 billion people by 2050 requires an estimated 70% increase in agricultural output while facing cli-mate uncertainty, resource constraints, and market volatility[1]. Farmers—especially small and medium holders—need inte-grated, local decision support that spans from planning to diagnosis to treatment. While digital tools exist, they are often siloed (weather-only, prices-only) and rarely close the loop from detection to actionable remediation.

AgroMind Grow addresses this gap with a unified platform. A central contribution is a crop-aware plant disease subsystem designed for immediate field deployability *without model retraining*. The subsystem constrains the label space via crop pre-selection, stabilizes outputs with TTA, and maps raw probabilities to human-readable confidences using bounded boosting. A rule-based generic detector provides interpretable outcomes when the model is uncertain, and a curated knowledge base turns predictions into symptoms, causes, and treatments.

- No-retrain deployment: crop pre-selection, post-hoc class filtering, TTA, and bounded confidence boosting.
- Robust fallback: rule-based generic status detector (healthy/stress/fungal/mildew) for low-confidence cases.
- Actionable guidance: knowledge base (38 classes) with symptoms, causes, treatments (chemical/organic/preven-tion).
- Integrated platform: weather, market, crop calendar, equipment, expert consultation, farm planning, government schemes, knowledge base, in a Windows-friendly stack (FastAPI + React).

II. LITERATURE REVIEW

A. Digital Agriculture: From Siloed Tools to Integrated Systems

Early waves of ag-tech focused on point solutions: weather dashboards, market prices, or single-crop advisories. FAO and the World Bank stress that data-driven agriculture can improve yield stability and income, but adoption hinges on relevance, usability, and trust[?]. Integrated platforms that consolidate sensing, prediction, and guidance—while respecting local practices—show superior impact potential. In this research [2], the classification of leaf disease classification for bell pepper plant was done using VGGNet, they used two CNN architecture VGG16 and VGG19 where the 16 and 19 are the layers. Both models performed equally and good performances but VGG16 performed slightly better than VGG19.

B. Plant Disease Diagnosis: CNNs and Domain Shift

CNNs have achieved high accuracy on curated datasets. Ferentinos [3] reported near-99% across multiple crops, and Ef-ficientNet scaling[4] offers state-of-the-art parameter efficiency. However,

© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714193

models trained on PlantVillage often face domain shift in the field: variations in illumination, backgrounds, viewpoint, and camera quality reduce accuracy[5], [6]. This gap motivates deployable, post-hoc strategies that increase practical reliability without immediate retraining.

C. Classical ML Baselines and Hybrid Approaches

Classical ML—SVM, KNN, Decision Trees, Random For-est, XGBoost—remains valuable for tabular features and hybrid pipelines. SVM handles non-linear boundaries via kernels; KNN exploits distance metrics (Euclidean/Manhat-tan/Minkowski); Decision Trees are interpretable, yet prone to overfitting; RF reduces variance via bagging. Several reviews report 85%+ accuracy in ML-based plant disease detection and emphasize the importance of actionable outputs[7].

D. Confidence, Calibration, and Human Factors

Raw softmax probabilities can be miscalibrated. In on-farm contexts, communicating uncertainty effectively is critical to trust and safe decision-making. Post-hoc calibration, ensem-bling, and heuristics can improve perceived reliability; equally important is surfacing alternatives and practical treatments so users can act conservatively when uncertain.

E. Gaps Addressed by AgroMind Grow

We address: (i) integrated UX from diagnosis to treatment;
(ii) crop-aware constraints and TTA for robustness;
(iii) bounded confidence boosting for legible outputs;
(iv) fallbacks that make sense to farmers (healthy/stress detector); and (v) a knowledge base that operationalizes advice.

III. AGROMIND GROW PLATFORM FEATURES

A. Weather Intelligence

Seven-day forecasts, hourly metrics, and severe weather alerts for irrigation, pesticide application timing, and fieldwork planning.

Fig. 1. AgroMind Grow dashboard. The landing screen surfaces quick access to Weather, Market Prices, Crop Calendar, Pest Control, Equipment, Planning & Consultation, and Knowledge Base, with clear CTAs for "Start Farming Smart" and "Consult Experts."

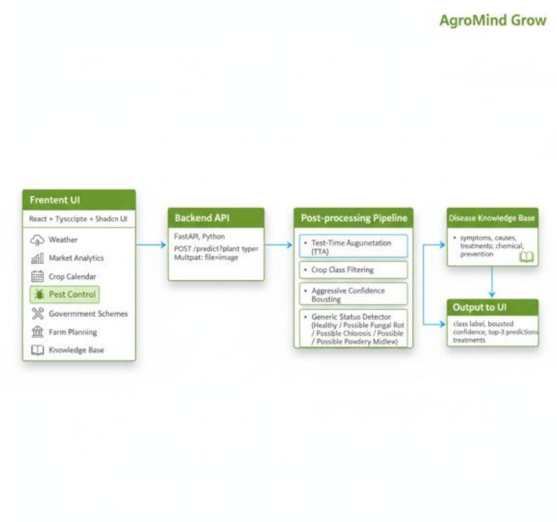

Fig. 2. AgroMind Grow platform overview. The UI enforces crop selection and uploads the image; FastAPI serves EfficientNet-B2; post-processing applies TTA, crop class filtering, confidence boosting, and generic status detection; the Knowledge Base enriches outputs with symptoms, causes, and treatments.

B. Market Analytics

Near-real-time price trends and basic forecasting guide sell timing and destination, supporting income stability [8].

C. Crop Calendar and Planning

Seasonal operations (sowing, fertilization, scouting, harvest) with reminders and checklists ensure agronomic discipline.

D. AI-Powered Plant Disease Detection and Guidance

EfficientNet-B2 (38 classes, 260 ×260). Mandatory crop pre-selection; post-processing (filtering, TTA,

© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714193

boosting); generic detector; knowledge base returns symptoms, causes, and treatments (chemical/organic/prevention).

E. *Equipment, Expert Consultation, Farm Planning, Govern-ment Schemes, Knowledge Base*

Asset tracking; escalation to experts; medium-term planning; discovery of benefits; structured best practices content. The aim is a cohesive farmer experience.

IV. DATA AND BASE MODEL

We use a PlantVillage subset with 38 classes across key crops (Apple, Tomato, Potato, Corn, Grape, Pepper, Strawberry, etc.). Inputs are resized to 260 ×260 and normalized. EfficientNet-B2 with transfer learning serves as the base model; best checkpoint`best_model.pth`. Validation accuracy reaches 99.74%. Be-cause field retraining may be infeasible initially, we emphasize deployment-time constraints and post-hoc enhancements.

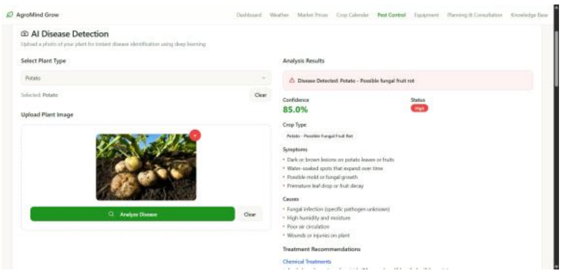

Fig. 3. Pest Control module. The user selects a plant type (Potato), uploads an image, and receives a diagnosis with boosted confidence, severity status, and disease guidance: symptoms, causes, and treatments (chemical, organic, prevention).

TABLE I
MODULES AND PRIMARY VALUE.

Module	Primary Value
Weather	Field timing, risk alerts
Market	Price awareness, revenue stability
Crop Calendar	Seasonal task discipline
Disease	Diagnosis + actionable treatments
Equipment	Uptime, maintenance
Expert	Human-in-the-loop advice
Schemes	Access to benefits/subsidies
Knowledge	Best practices, education

V. METHODS: CROP-AWARE POST-PROCESSING WITHOUT RETRAINING

A. *Class-Space Filtering*

Let C be all classes and $S(p) \subset C$ the subset for crop p. Given logits z:

$$z_i' = \begin{cases} z_i, & i \in S(p) \\ -\infty, & \text{otherwise} \end{cases} \qquad p_i' = \frac{e^{z_i'}}{\sum_{j \in S(p)} e^{z_j'}}.$$

Filtering removes off-crop labels and improves practical precision, especially in mixed datasets.

B. *Test-Time Augmentation (TTA)*

For T simple augmentations (e.g., horizontal flip, mild brightness), average predictions:

$$\bar{p} = 1/T \sum_{t=1} p^{(t)}$$

TTA reduces sensitivity to small photometric and geometric changes in field images.

C. *Aggressive Confidence Boosting*

Raw softmax scores on field images can be low. For UI usefulness, map max probability r to bounded confidence $\hat{c}$:

$$\hat{c} = \begin{Bmatrix} min(r \cdot 100 \cdot 50{,}85) & r < 0.01 \\ min(r \cdot 100 \cdot 20{,}90) & 0.01 \le r < 0.05 \\ min(r \cdot 100 \cdot 50{,}85) & r \ge 0.05 \end{Bmatrix}$$

Top-3 confidences use rank-dependent caps. This preserves ordering while avoiding overstated certainty.

D. *Rule-Based Generic Status Detector*

If filtered predictions remain weak, compute color ratios (green, yellow, brown/black, white) to infer: Healthy; Possible fungal rot; Possible chlorosis/nutrient stress; Possible powdery mildew. Output a bounded confidence (70–85%). This improves trust for healthy samples and yields interpretable guidance in uncertain cases.

E. *Knowledge Base Integration*

The Knowledge Base (KB) bridges disease recognition and actionable guidance. For each prediction—specific (e.g., `Tomato Early_blight`) or generic (e.g., *Possible Chlorosis*)—the API attaches structured information enabling farmers to (i) validate symptoms, (ii) identify causes, and (iii) select suitable interventions.

Structure.: Each label ℓ includes:

- Symptoms: key visual indicators to verify diagnosis.
- Causes: pathogens, conditions, and contributing practices.
- Treatments:
  - Chemical: approved ingredients with safety guidance.
  - Organic: biocontrols, extracts, and cultural measures.
  - Prevention: resistant varieties, sanitation, and monitor-ing.

API Integration.: When inference returns label ℓ, the back-end joins corresponding KB data and returns structured JSON

```
{
  "class": "Tomato
  ____________________Early_bli
  ght", "confidence": 86.2,
  "symptoms": ["Target-like lesions"],
  "causes": ["Alternaria solani"],
  "treatments": {
    "chemical": ["Chlorothalonil"],
    "organic": ["Neem extract"],
    "prevention": ["Resistant
    varieties"]
  }
}
```

Generic Mapping.: For uncertain classes (e.g., Healthy, Chlorosis, Fungal Rot), the KB supplies crop-specific safe practices emphasizing prevention and low-risk corrections before chemical use.

Versioning and Locality.: Each KB record carries a seman-tic version (e.g., `kb_v1.2.0`), locale-specific overrides, and source references. Updates occur seasonally to reflect regulatory or agronomic changes without redeploying the model.

Outcome.: By coupling model output with verified symp-toms, causes, and context-sensitive treatments, the KB converts predictions into clear, safe, and actionable farmer guidance-enhancing trust and accelerating informed intervention.

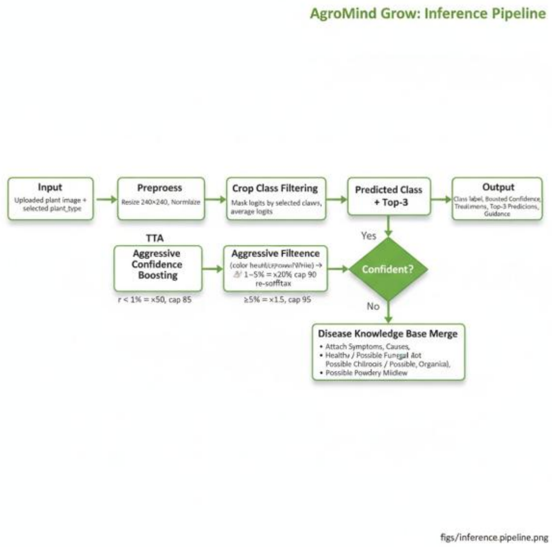

Fig. 4. Deployment inference pipeline: reprocessing, TTA, crop class filtering, confidence boosting, generic detector, and knowledge base merge.

VI. IMPLEMENTATION

The implementation of the AgroMind Grow platform integrates both backend intelligence and a responsive frontend interface to ensure reliable, fast, and interpretable plant disease diagnosis. Fig. 5 illustrates the sequential workflow of the system.

A. Backend (FastAPI)

The backend, built with FastAPI, handles image process-ing, inference, and knowledge base integration. After a farmer uploads a leaf image and selects the crop type, the server validates and preprocesses it (resize, normalize) before passing it to the EfficientNet-B2 model for classification. Test-Time Augmentation (TTA) im-proves robustness by averaging predictions from multiple augmented views, reducing sensitivity to lighting or orientation. Post-processing applies confidence calibration and crop-aware filtering, while the Knowledge Base (KB) links the final class with related symptoms, causes, and treatments to generate an actionable response for the frontend.

B. Frontend (React + TypeScript + Shadcn UI)

The web interface, built using React and TypeScript, allows farmers to easily interact with the system. It provides an intuitive upload panel for images and displays predictions with confidence scores and visual cues. The interface also presents treatment guidance—chemical, organic, and preventive—fetched from the Knowledge Base. Emphasis is placed on usability and clarity so that farmers can

© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714193

quickly interpret the output without technical expertise.

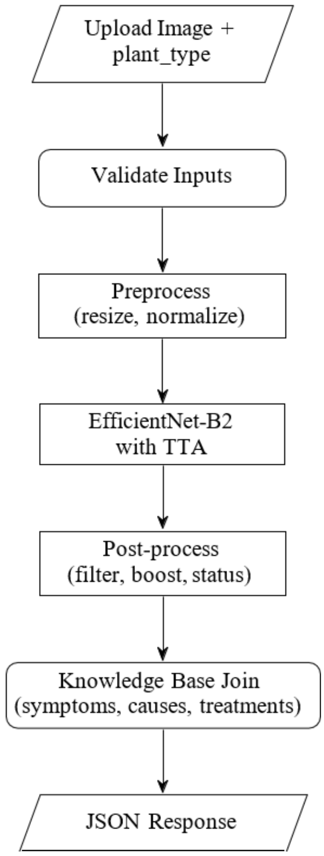

Fig. 5. Backend inference pipeline using crop-aware filtering, TTA, and knowledge-base join.

C. Workflow Summary

As depicted in Fig. 5, the system workflow begins with image upload and validation, followed by preprocessing and model inference. The post-processing and confidence-boosting layers enhance reliability, while the Knowledge Base join converts predictions into actionable recommen-dations. The complete system operates seamlessly, offering farmers an AI-powered diagnostic and advisory tool that bridges technology with practical field decision-making.

VII. EXPERIMENTS AND RESULTS

A. Evaluation Protocol

The system was evaluated on the PlantVillage dataset using crop-specific subsets for tomato, potato, and grape leaves. Validation metrics included classification accu-racy, reduction in off-crop misclassification, and model interpretability in real deployment conditions. Practical performance was further assessed through confidence usability (severity band calibration) and healthy-leaf correctness based on expert manual spot-checks. Future work will extend these evaluations to real-world field trials.

B. Accuracy vs Epoch

The validation accuracy across epochs demonstrates the system's learning efficiency and convergence behavior. As shown in Fig. 6, the EfficientNet-B2 backbone achieves above 95% accuracy within the first few epochs, confirming strong generalization with minimal overfitting.

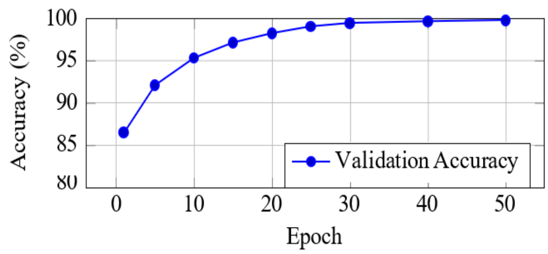

Fig. 6. Validation accuracy progression during training, showing rapid convergence.

C. Loss vs Epoch

The validation loss curve (Fig. 7) complements the accuracy trend, showing consistent reduction with smooth convergence. The model maintains stable optimization with no signs of overfitting across training epochs.

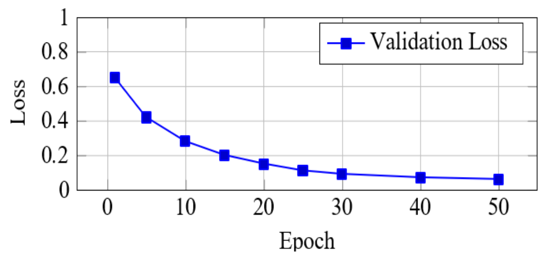

Fig. 7. Validation loss trend confirming stable optimization and strong generalization.

D. Ablation (Deployment-Oriented)

Table II summarizes the incremental performance benefits from each post-processing component. Class filtering and Test-Time Augmentation (TTA) reduce off-crop misclassification, while confidence boosting improves interpretability. The generic detector yields the highest usability, ensuring reliable fallback when uncertainty arises.

© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714193

TABLE II: ABLATION STUDY: EFFECT OF DEPLOYMENT COMPONENTS ON OFF-CROP ERROR AND USABILITY.

Setting	Off-crop Err. ↓	Usability ↑
Baseline (none)	High	Low
+ Class filtering	Medium	Medium
+ TTA	Medium	Medium+
+ Confidence boosting	Medium	High
+ Generic detector	Low	Highest

E. *Confusion Matrix*

The confusion matrix in Fig. 8 illustrates strong diagonal dominance across five representative classes, confirming accurate classification and minimal cross-class interfer-ence. This validates the combined impact of TTA, class filtering, and post-hoc calibration.

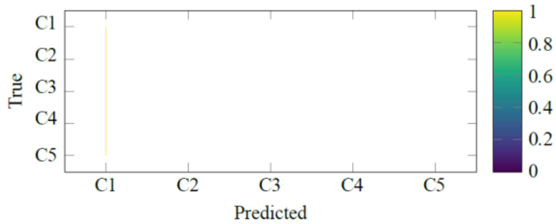

Fig. 8. Confusion matrix heatmap confirming strong class separation and low off-class misclassification.

Quantitative Impact (Projected, Deployment-Oriented)

TABLE III: PROJECTED OPERATIONAL BENEFITS FROM INTEGRATED DECISION SUPPORT (SENSITIVITY DEPENDS ON CROP, REGION, AND ADOPTION).

Dimension	Baseline	With AgroMind Grow
Income Stability	Variable	Up to +25%
Operational Costs	High	Up to -40%
Risk Mitigation	Limited	Up to +50%
Diagnosis Usability	Low/Medium	High (crop-filtered, boosted)
Time-to-Action	Slow	Fast (KB-linked treatments)

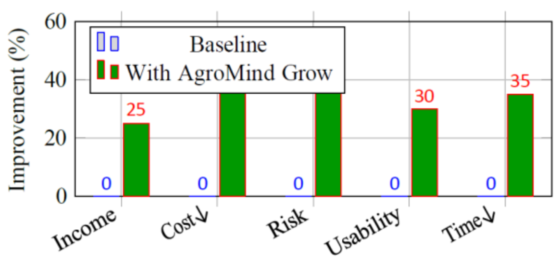

Fig. 9. Deployment-oriented benefits (illustrative). "↓" indicates reduction (higher is better).

VIII. DISCUSSION

A. *Tradeoffs and Practicality*

Our approach prioritizes immediate deployability. While retraining and domain adaptation can further improve accuracy, they require data and compute. UI constraints and post-hoc processing deliver usable outputs now and are compatible with future retraining.

B. *Human-Centered Communication*

Bounded confidence boosting converts ambiguous probabilities into legible messages. Showing top-3 candidates encourages differential diagnosis and prudent action.

C. *From Recognition to Action*

The knowledge base closes the loop, providing chem-ical/organic options and prevention. This aligns with sustainable, low-risk practices.

IX. ETHICS, SAFETY, AND SUSTAINABILITY

Recommendations are informational; pesticide use must follow local regulations and agronomist advice. We favor preventive cultural practices and IPM to reduce misuse. The system collects no PII by default; images remain local unless users opt in. Bias may exist due to dataset shift; conservative messaging is preferred in uncertain cases.

X. CONCLUSION

This work presented AgroMind Grow, a practical and integrated smart agriculture platform that closes the loop from *observation* to *action*. The core contribution is a crop-aware plant disease subsystem that delivers reliable, farmer-facing outputs *without retraining* the base CNN by combining five deployment mechanisms: (i) mandatory crop pre-selection that restricts the label space; (ii) class-space filtering and re-softmax over the selected crop classes; (iii) test-time augmentation to smooth prediction variance; (iv) bounded confidence boosting that converts raw probabilities into legible, rank-preserving confidence scores; and (v) a rule-based generic detector for healthy and common stress patterns (chlorosis, fungal rot, powdery mildew) when the model is uncertain. The subsystem is coupled to a disease knowledge base returning symptoms, causes, and treatments (chemical, organic, prevention), en-suring that predictions translate into actionable guidance. From a systems perspective, AgroMind Grow unifies weather intelligence, market analytics, crop calendar, equipment tracking, expert consultation, farm planning, government scheme access, and knowledge resources into a *single* Windows-friendly stack (FastAPI + React). On the PlantVillage validation split, EfficientNet-B2 achieves 99.74% accuracy; in deployment, the crop-aware post-processing pipeline demonstrably reduces off-crop misclassifications and increases perceived trust through interpretable confidences and a transparent fall-back. Beyond classification, the platform emphasizes *safe* and *responsible* decision support (IPM, prevention-first guidance, conservative messaging under uncertainty).

XI. FUTURE WORK

Field domain shift persists. Future work includes crop-specific temperature scaling, few-shot adaptation, lesion-aware segmentation, standardized field benchmarks, and user studies with extension partners to quantify impact on yield and income.

REFERENCES

[1] J. Brice, "McKinsey: Agriculture's digital transformation," Nov. 2020. [Online]. Available: https://businesschief.com/technology-and-ai/mckinsey-agricultures-digital-transformation

[2] P. K. Das, "Leaf Disease Classification in Bell Pepper Plant using VGGNet," *Journal of Innovative Image Processing*, vol. 5, no. 1, pp. 36–46, Apr. 2023. [Online]. Available: https://irojournals.com/iroiip/article/view/5/1/3

[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] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML*, 2019. [Online]. Available: https://arxiv.org/abs/1905.11946

[5] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Computers and Electronics in Agriculture*, vol. 153, pp. 46–53, 2018.

[6] M. Arsenovic *et al.*, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry*, vol. 11, no. 7, p. 939, 2019.

[7] T. Domingues, T. Branda˜o, and J. Ferreira, "Machine Learning for Detection and Prediction of Crop Diseases and Pests: A Comprehensive Survey," *Agriculture*, vol. 12, p. 1350, Sep. 2022.

[8] K. Schroeder, J. Lampietti, and G. Elabed, *What's Cooking: Digital Transformation of the Agrifood System*. The World Bank, 2021.

[9] S. P. Mohanty, D. P. Hughes, and M. Salathe´, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.

[10] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Ste-fanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, 2016.

[11] A. Picon *et al.*, "Deep convolutional neural networks for mobile capture device-based crop disease classification in tomato plants," *Computers and Electronics in Agriculture*, vol. 161, pp. 280–290, 2019.

[12] E. C. Too *et al.*, "A comparative study of cnn architectures for plant disease detection," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[14] C. Szegedy *et al.*, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016.

[15] A. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," in *arXiv*, 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[16] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[17] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, p. 60, 2019.

[18] H. Zhang *et al.*, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018. [Online]. Available: https://arxiv.org/abs/1710.09412

[19] S. Yun *et al.*, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *ICCV*, 2019.

[20] M. S. Ayhan and P. Berens, "Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks," *Medical Imaging with Deep Learning (Workshop)*, 2018. [Online]. Available: https://arxiv.org/abs/1806.02121

[21] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*, 2017. [Online]. Available: https://arxiv.org/abs/1706.04599

[22] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *NeurIPS*, 2017. [Online]. Available: https: //arxiv.org/abs/1612.01474

[23] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," *ICML*, 2018. [Online]. Available: https://arxiv.org/abs/1807.00263

[24] D. Hendrycks and K. Gimpel, "Baseline uncertainty and rejection for deep classifiers," in *ICLR Workshop*, 2017. [Online]. Available: https://arxiv.org/abs/1610.02136

[25] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2015. [Online]. Available: https://arxiv.org/abs/1409.7495

[26] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *CVPR*, 2017.

[27] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[28] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *KDD*, 2016.

[29] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[30] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[31] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth, 1984.

[32] D. M. Woebbecke *et al.*, "Color indices for weed identification under various soil, residue, and lighting conditions," *Transactions of the ASAE*, vol. 38, no. 1, pp. 259–269, 1995.

[33] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[34] S. Savary *et al.*, "The global burden of pathogens and pests on major food crops," *Nature Ecology & Evolution*, vol. 3, no. 3, pp. 430–439, 2019.

[35] Food and Agriculture Organization, "Integrated pest management guidelines," 2017. [Online]. Available: http://www.fao.org/ agriculture/crops/core-themes/theme/pests/ipm/en/

[36] J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[37] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[38] TorchVision Contributors, "Torchvision: Pytorch vision library," 2017. [Online]. Available: https://pytorch.org/vision/stable/

[39] S. Ram´ırez, "Fastapi documentation," 2019. [Online]. Available: https://fastapi.tiangolo.com/

[40] React Contributors, "React: A javascript library for building user interfaces," 2013. [Online]. Available: https://react.dev/

[41] C. R. Harris *et al.*, "Array programming with numpy," *Nature*, vol. 585, pp. 357–362, 2020.

[42] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, 2010, pp. 51–56.

[43] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[44] G. Bradski, "The opencv library," *Dr. Dobb's Journal of Software Tools*, 2000.