

Rice Leaf Disease Detection Using EfficientNetB0 Transfer Learning with a Weather-Aware Rule-Based Advisory System

Md Mahmudul Hassan, Md Shahinul Alam Murad, Nadia Afrin Ritu

Department of Computer Science and Engineering, Jahangirnagar University, Savar, Dhaka, Bangladesh

Email: mahmuduljucse30@gmail.com, shahinul.alam007@gmail.com, nadiaritu@juniv.edu

How to cite this paper: Hassan, M.M., Murad, M.S.A. and Ritu, N.A. (2026) Rice Leaf Disease Detection Using EfficientNetB0 Transfer Learning with a Weather-Aware Rule-Based Advisory System. *Journal of Computer and Communications*, 14, 74-97.

<https://doi.org/10.4236/jcc.2026.1410004>

Received: August 30, 2026

Accepted: October 24, 2026

Published: October 27, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Rice is Bangladesh's most important staple crop, and its yield is repeatedly threatened by disease outbreaks that are difficult to diagnose quickly given the limited number of agricultural extension officers available to smallholder farmers. This paper presents a hybrid diagnosis-and-advisory system for rice leaf disease that couples a deep-learning image classifier with an independent, rule-based weather-risk expert system. An EfficientNetB0 backbone, pre-trained on ImageNet, is fine-tuned through a two-phase transfer-learning procedure on a 531-image, five-class, field-collected Bangladeshi dataset covering Bacterial leaf blight, Blast, Brown spot, Leaf smut and Tungro, using computed class weighting to counter marked class imbalance. On a held-out test split of 80 images the classifier reaches 96.25% accuracy and a 96.40% weighted F1-score, outperforming VGG16, ResNet50 and AlexNet fine-tuned on the same dataset and evaluated on the same test split, while using markedly fewer parameters. Independently of the classifier, a forward-chaining expert system evaluates temperature, humidity and rainfall against fifteen agronomically grounded IF-THEN rules, three per disease and each a graded test contributing a full or reduced weight, to produce a HIGH/MODERATE/LOW risk level and a disease-specific treatment advisory. The classifier's diagnosis and the expert system's risk assessment are then combined into a single interactive report. The results show that transfer learning with a compact classification head, strong data augmentation and computed class weighting can achieve high accuracy even on a small, imbalanced, real-world dataset, and that pairing a CNN-based diagnosis with an interpretable rule-based advisory layer yields a more directly actionable output for farmers and extension officers than a bare classification label.

Keywords

Rice Leaf Disease, Deep Learning, Transfer Learning, EfficientNetB0, Convolutional Neural Network, Expert System, Weather-Aware Advisory, Precision Agriculture

1. Introduction

Rice is grown on more than three-quarters of Bangladesh's cultivated land and underpins both national food security and the livelihoods of a large share of the rural population. Because so much of the country's arable land is devoted to a single crop, even a moderate disease outbreak can have an outsized effect on national yield: field surveys in coastal Bangladesh found that the large majority of farmers had experienced a rice blast outbreak within the previous five growing seasons, and separate studies have linked inadequate disease recognition to substantial annual production losses. Diagnosis today still depends heavily on a farmer's own visual judgement or on a scarce number of agricultural extension officers, which is slow, inconsistent, and difficult to scale to the millions of small-holder plots that need monitoring every season.

This work addresses that gap by building an image classifier for five economically important rice leaf diseases-Bacterial leaf blight, Blast, Brown spot, Leaf smut and Tungro-using EfficientNetB0 transfer learning tuned specifically for a small, class-imbalanced, real-world dataset of 531 images (**Figure 1**). The model is trained in two phases: first with the pretrained backbone frozen and only a light-weight classification head learning on top, then with a portion of the backbone unfrozen and fine-tuned using class-weighted training so that the majority classes do not dominate learning at the expense of rarer diseases. Rather than stopping at a bare label prediction, the system pairs the classifier's output with a rule-based weather-risk advisory layer: given the predicted disease and current weather conditions, the system evaluates a set of expert-defined rules to determine a risk level, records which rules fired, and generates a concrete, prioritised advisory message. This turns a single-word diagnosis into an actionable recommendation that a farmer or extension officer can act on immediately, rather than a classification result that still requires separate expert interpretation.

Beyond training a classifier, this work also set out to prepare a small, field-collected dataset properly: splitting it into training, validation, and test subsets, and, with only 531 images spread unevenly across five classes (as few as 40 images for Leaf smut), choosing training settings carefully to sit in the narrow band between underfitting and overfitting rather than reusing settings tuned for a much larger dataset. Read against the literature surveyed in Section 2, the contribution of this work is not a new deep-learning architecture-EfficientNetB0 is used as published, not modified-but a small-dataset training methodology and a genuinely integrated diagnosis-plus-advisory pipeline that few prior studies combine. Specifically, this

paper (i) fine-tunes EfficientNetB0 for five-class rice leaf disease classification using a two-phase, class-weighted training procedure suited to a small, imbalanced dataset; (ii) extends a rule-based weather-risk expert system to five disease-specific IF-THEN rule groups parameterised with agronomically grounded temperature, humidity and rainfall thresholds; (iii) packages the complete pipeline-data preparation, augmentation, transfer learning, evaluation, rule-based advisory, and an interactive prediction widget-into a single, reproducible notebook; and (iv) empirically compares the EfficientNetB0 architecture against VGG16, ResNet50 and AlexNet trained on the same 531-image dataset, rather than relying on accuracy figures quoted from those architectures' original, much larger benchmark datasets.

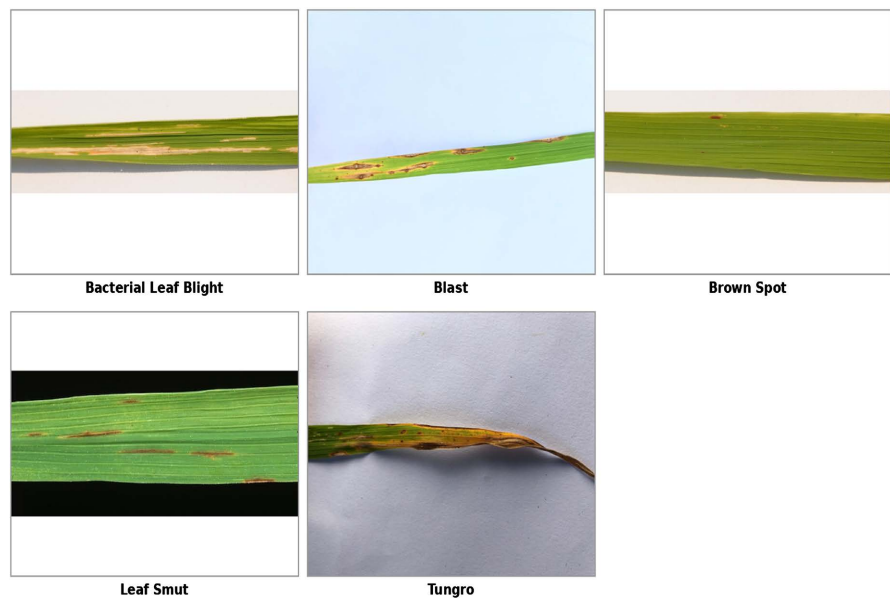


Figure 1. Sample leaf images for the five target diseases: Bacterial Leaf Blight, Blast, Brown Spot, Leaf Smut, and Tungro.

The remainder of this paper is organised as follows. Section 2 reviews related work on rice leaf disease detection and weather-based disease advisory systems. Section 3 describes the dataset, the model architecture and training procedure, and the rule-based weather advisory system. Section 4 presents the classification results, compares them against prior literature and against three additional backbone architectures trained on the same dataset, and discusses the hyperparameter choices made. Section 5 concludes the paper and outlines directions for future work.

2. Related Work

This section reviews recent work (2018-2025) on rice leaf disease detection, with a deliberate emphasis on studies conducted in, or using data collected from, Bangladesh, since the dataset and intended use case of this work are both Bangladeshi. Where a Bangladesh-specific study was not available for a particular sub-topic, closely related international work is included instead. **Table 1** summarises the studies reviewed.

Table 1. Summary of reviewed literature on rice leaf disease detection.

Ref.	Algorithm/Methodology	Results	Limitations
[1]	Custom light CNN vs. InceptionNetV2, EfficientNetV2, MobileNetV2 transfer learning; Bangladesh field dataset (RiceLeafBD)	EfficientNetV2 strongest, 91.5% accuracy	Single-country field dataset; no weather/advisory component
[2]	Classical ML (Logistic Regression, KNN, Decision Tree J48, Naive Bayes) on hand-crafted leaf features; 3 diseases	Decision Tree (J48) best, 97.92% test accuracy	Only 3 classes; hand-crafted features; no weather integration
[3]	AlexNet CNN, 3 diseases, 70/30 split with augmentation	99.42% test accuracy	Only 3 diseases; heavy reliance on augmentation; no advisory module
[4]	GLCM texture + Local Binary Pattern + Hu-moment features with KNN	98% accuracy	Hand-crafted feature engineering, not end-to-end deep learning
[5]	Qualitative regional review of climate-change impacts on crop diseases across Southeast Asia	Synthesises links between climate factors and crop disease risk	No predictive model or dataset of its own; broad multi-crop scope
[6]	DC-GAN-MDFC-ResNet: GAN augmentation + feature-compensation ResNet; IoT-oriented; PlantVillage benchmark	95.99% accuracy	Lab-condition images, not field/Bangladesh-specific; no weather advisory
[7]	LSTM forecasting next-year rice blast incidence from disease history plus weather data (South Korea)	Best variant reached 54.4%-67.4% accuracy across regions	Statistical forecasting only, no image-based diagnosis
[8]	Faster R-CNN object-detection pipeline for real-time field-deployed diagnosis	Demonstrated real-time-capable detection	Detection-focused rather than fine-grained classification
[9]	Classical ML classifiers with preprocessing on Bangladeshi rice-leaf data	Exceeds 97% accuracy	Strong preprocessing dependence; generalisation unclear
[10]	SVM-based disease recognition with automatic remedy suggestion (Dhaka-area farmers)	Early working diagnosis-plus-advisory system	Remedy suggestions not weather-aware; small-scale evaluation
[11]	VGG, ResNet, DenseNet compared on a BRRI-sourced dataset against two other data sources	Performance varied noticeably by dataset	No single best architecture across sources
[12]	MobileNetV2 and EfficientNetB3 fine-tuned on two merged Bangladeshi rice-disease datasets	EfficientNetB3 reached 91% test accuracy	Merged-dataset evaluation may mask distribution shift
[13]	Farmer- and expert-interview study on weather-information needs, coastal Bangladesh	Poor access to timely weather information identified as a major barrier	Qualitative study, not a technical model
[14]	Weather-forecast-driven early warning system for wheat blast (Bangladesh and Brazil)	Location-specific spray/no-spray advisories delivered at scale	Built for wheat blast, not rice

Continued

[15]	TRiP: SENet-based transfer learning platform, same 5-disease taxonomy (China)	95.73% accuracy	Non-Bangladeshi dataset and field conditions
[16]	MobileNetV3-Large vs. EfficientNet-B0 on a 6-class Kaggle rice dataset	MobileNetV3-Large slightly ahead (95.83% vs 93.18%)	Kaggle dataset, not field-collected; 6-class taxonomy
[17]	ESforRPD2: rule-based expert system for rice plant disease diagnosis	Functional rule-based diagnosis engine	Rule-base maintainability trade-off; not paired with a classifier
[18]	RiceMan: ontology-based expert system for disease identification and control	Functional ontology-based reasoning system	Same maintainability trade-off; not paired with a classifier
[19]	EfficientNet compound scaling method (foundational architecture paper)	State-of-the-art ImageNet accuracy/efficiency trade-off	General-purpose, not agriculture-specific
[20]	Deep Residual Learning (ResNet, foundational architecture paper)	State-of-the-art ImageNet performance via residual connections	General-purpose, not agriculture-specific

Seven studies form the core comparison set for this work, spanning image-based disease classification, classical machine-learning baselines and weather-based disease forecasting. Rimi *et al.*'s RiceLeafBD study is the closest direct comparison: using a Bangladesh field-collected dataset, they benchmarked a custom light CNN against InceptionNetV2, EfficientNetV2 and MobileNetV2 transfer-learning backbones, finding EfficientNetV2 the strongest performer at 91.5% accuracy [1]—a result this work's own EfficientNetB0 classifier can be read directly against. Two further Bangladeshi studies show that even non-deep-learning approaches can reach very high accuracy when the number of disease classes is small: Ahmed *et al.* compared Logistic Regression, KNN and Decision Tree classifiers on hand-crafted leaf features across three diseases and found a Decision Tree (J48) model the strongest at 97.92% test accuracy [2], while Matin *et al.* trained an AlexNet CNN on the same three-disease setting and reported 99.42% test accuracy after augmentation [3]. Saha combined GLCM texture, Local Binary Pattern and Hu-moment features with a KNN classifier to reach 98% accuracy [4]. Read together, these three studies show that on small, few-class Bangladeshi datasets, both classical machine learning and simple CNNs can match or exceed the accuracy of larger transfer-learning models-but, as this work's own five-disease, 531-image dataset illustrates, that advantage narrows sharply as the number of classes grows and background variation increases.

The remaining priority studies motivate the weather-integration half of this work rather than its classifier. Lim *et al.* surveyed how climate change is reshaping crop-disease pressure across Southeast Asia, arguing that rising temperatures and shifting rainfall patterns are already changing which diseases threaten staple crops in ways a static disease-detection model cannot capture on its own [5]. Wang *et al.* addressed robustness under limited field data with a GAN-based augmentation and multidimensional-feature-compensation ResNet built for IoT-deployed rice-disease sensors, reaching 95.99% accuracy on the PlantVillage benchmark [6].

Most directly relevant to this work's rule-based advisory module, Kim, Roh and Kim built an LSTM network that forecasts a region's rice blast risk a year in advance from prior disease scores together with temperature, humidity and sunshine-hour data, reaching 54.4% - 67.4% accuracy depending on region [7]—modest numbers that illustrate how difficult pure long-range statistical forecasting of disease risk is, compared to the short-horizon, rule-based weather-risk flags generated in this work from a single live reading.

Beyond these seven priority studies, Bangladeshi datasets and studies make up the wider set of sources most directly comparable to this work [8]-[20]. Several groups have independently collected their own field data (RiceLeafBD [1], the BRRI-sourced set used in [11]) precisely because, as this work also found, off-the-shelf international datasets do not always reflect local rice varieties, field backgrounds, or camera conditions. Bari *et al.* demonstrated a real-time-capable Faster R-CNN pipeline for field-deployed diagnosis [8], while Khasim *et al.* showed that classical machine-learning classifiers can exceed 97% accuracy on Bangladeshi rice-leaf data once appropriate preprocessing is applied [9]. A BRRI-sourced dataset was used to compare VGG, ResNet and DenseNet directly against two other data sources, finding that performance varied noticeably by dataset [11]. Most recently, a multi-university Bangladeshi collaboration merged two separate rice-disease datasets and fine-tuned MobileNetV2 and EfficientNetB3, with EfficientNetB3 reaching 91% test accuracy [12].

The weather-and-advisory side of the literature is thinner but directly relevant to the rule-based module in this work. Pinki *et al.*'s early SVM-based system already paired disease recognition with an automatic remedy suggestion for Dhaka-area paddy farmers [10]. Uddin and Ludwig's farmer- and expert-interview study of coastal Bangladesh found that most farmers had experienced a blast outbreak in the last five years and identified poor access to timely, location-specific weather information as a major barrier to proactive disease management—direct evidence for a weather-driven rule engine, rather than a bare disease label, as the more useful output for a Bangladeshi user. A large-scale, currently operating example of the same idea is the wheat-blast early warning system jointly run in Bangladesh and Brazil, which converts gridded weather forecasts into location-specific advisories delivered to thousands of registered farmers and officers [14]; this work's smaller rule-based engine follows the same weather-to-advisory logic, applied instead to five rice diseases. Internationally, TRiP is methodologically the closest comparison point, as it targets the exact same five disease classes with a SENet-based transfer-learning model, reaching 95.73% accuracy [15], while a separate comparison of MobileNetV3-Large and EfficientNet-B0 on a Kaggle rice dataset found MobileNetV3-Large slightly ahead after fine-tuning [16]. Two international expert-system papers—ESforRPD2 [17] and RiceMan [18]—informed the design of the IF-THEN rule base used in this work, in particular the trade-off both flag between interpretability and long-term rule-base maintainability as more diseases and conditions are added.

3. Materials and Methods

Figure 2 gives a single overview of the complete methodology before this section details each stage. The raw 531-image, five-class dataset is augmented and split (Sections 3.1-3.2), then feeds an EfficientNetB0 transfer-learning classifier trained in a two-phase procedure (Section 3.4). Entirely independently of the image pipeline, a rule-based weather-risk expert system (Section 3.6) turns a live temperature/humidity/rainfall reading into a disease-specific risk level and treatment advisory. The two outputs are combined into a single diagnosis-plus-advisory report (Section 3.7).

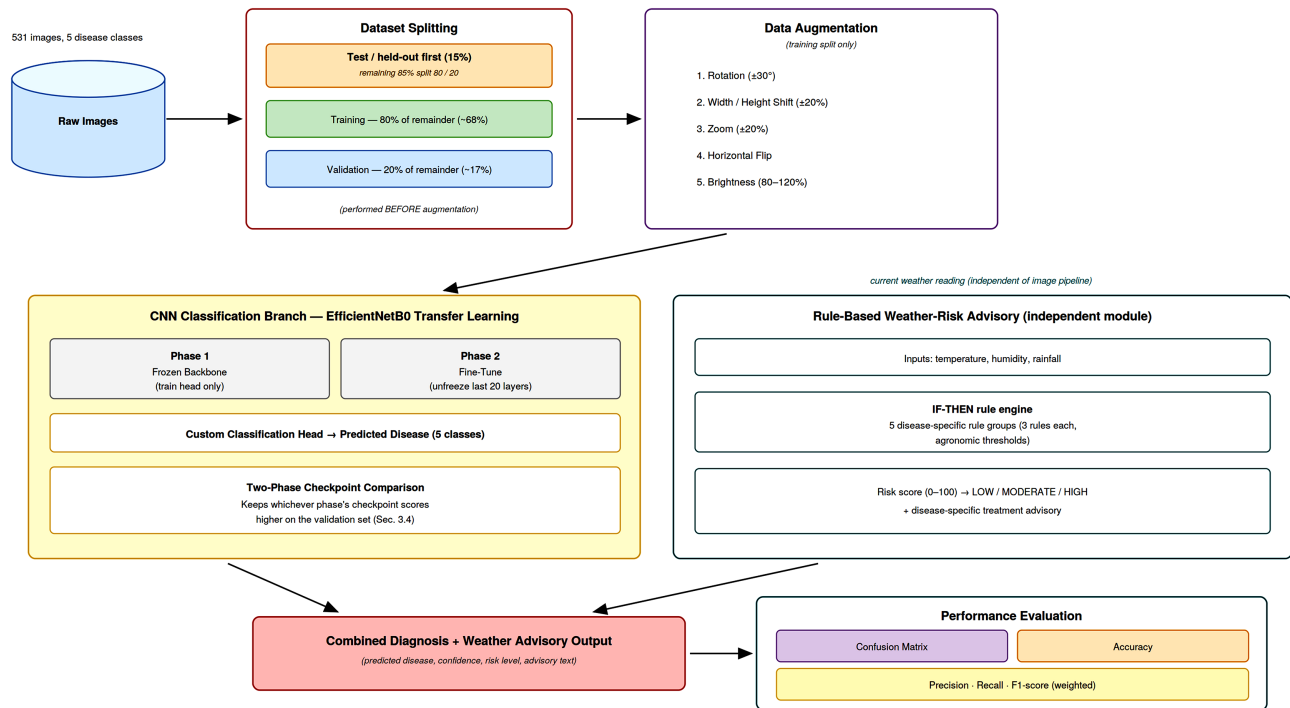


Figure 2. Proposed framework.

3.1. Dataset

This work uses a rice leaf disease image dataset containing 531 images spread unevenly (**Figure 3**) across five classes: Bacterial leaf blight, Blast, Brown spot, Leaf smut, and Tungro. **Table 2** gives the per-class breakdown. The class distribution is visibly imbalanced—Brown spot (211 images) has more than five times as many images as Leaf smut (40 images)—which is addressed with computed class weighting (Section 3.4) rather than discarding data from the larger classes.

For each class, scikit-learn's `train_test_split` function is used twice: first, 15% of that class's images are held out as a test set; the remaining 85% are then split again, with 20% used for validation and the rest for training. This is done independently per class so that the overall five-class balance is preserved in each of the three subsets, yielding 359 training, 92 validation, and 80 test images. Only the training folder is augmented; the validation and test folders are copied once and never modified again until evaluation.

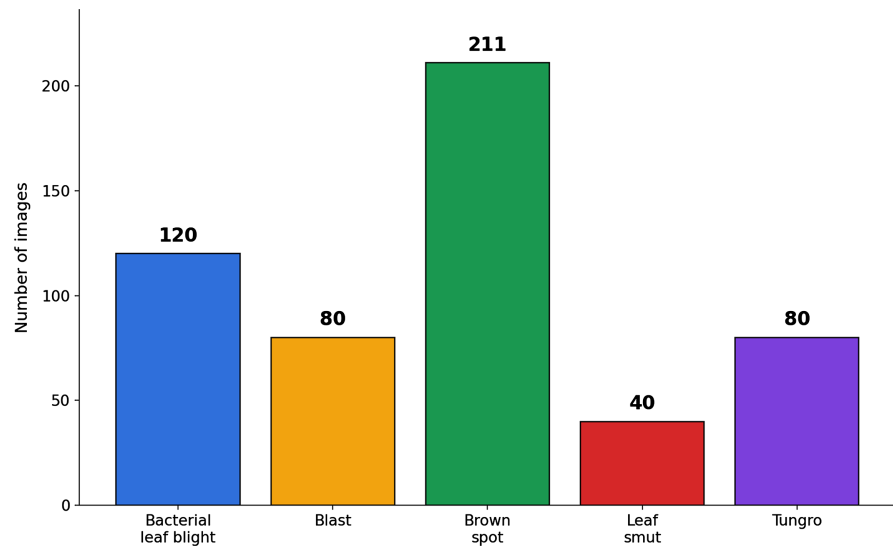


Figure 3. Class distribution across the 531-image, five-class dataset.

Table 2. Per-class image distribution.

Class	Mapped Name	Images	% of Total
Bacterial leaf blight	Bacterialblight	120	22.6%
Blast	Blast	80	15.1%
Brown spot	Brownsplot	211	39.7%
Leaf smut	Leafsmut	40	7.5%
Tungro	Tungro	80	15.1%
Total	-	531	100%

3.2. Data Augmentation

All images are resized to 224×224 pixels and loaded in batches of 16. Keras' ImageDataGenerator applies augmentation to the training images only, using EfficientNet's own preprocessing function together with random rotation of up to 30 degrees, width and height shifts of up to 20%, zoom of up to 20%, brightness adjustment between 80% and 120% of the original, and random horizontal flips. The validation and test generators use the same EfficientNet preprocessing but no augmentation, so that the reported metrics reflect an honest, unmodified read of model performance. Because augmentation happens on the fly rather than being pre-computed, it does not change the stored dataset size; what changes is how many distinct transformed versions of each image the network is exposed to over the course of training-roughly 359×32 (approximately 11,488) transformed presentations during the 32-epoch Phase 1 alone.

3.3. Model Architecture

Transfer learning reuses a network already trained on a large, general-purpose

dataset (ImageNet) as the starting point for a task with a much smaller labelled dataset, on the premise that the low- and mid-level visual features learned from ImageNet-edges, textures, colour gradients-are broadly useful even for a visually different task such as classifying diseased rice leaves. EfficientNetB0 [19] was selected as the backbone because its compound scaling of depth, width and input resolution gives it a strong accuracy-to-parameter trade-off relative to alternative ImageNet-pretrained backbones such as VGG16, ResNet50 [20] and MobileNetV2, which suits both the small dataset available and the modest compute budget used in this work; this choice is examined empirically in Section 4.3.

The classifier is built on an EfficientNetB0 base (input shape $224 \times 224 \times 3$, weights pretrained on ImageNet, `include_top = False`), initially frozen so its pre-trained weights are not modified. On top of this backbone sits a custom classification head: GlobalAveragePooling2D to collapse the backbone's feature maps into a single vector, BatchNormalization, a Dense layer of 128 units with ReLU activation, Dropout at a rate of 0.4, a second Dense layer of 64 units with ReLU activation, Dropout at a rate of 0.3, and a final Dense layer of 5 units with softmax activation producing one probability per disease class (Figure 4). Keeping the head compact limits the number of parameters that must be learned from a limited amount of training data, reducing overfitting risk while still benefiting from the general-purpose visual features already captured by the pretrained backbone.

Model Architecture: EfficientNetBo Backbone + Compact Classification Head

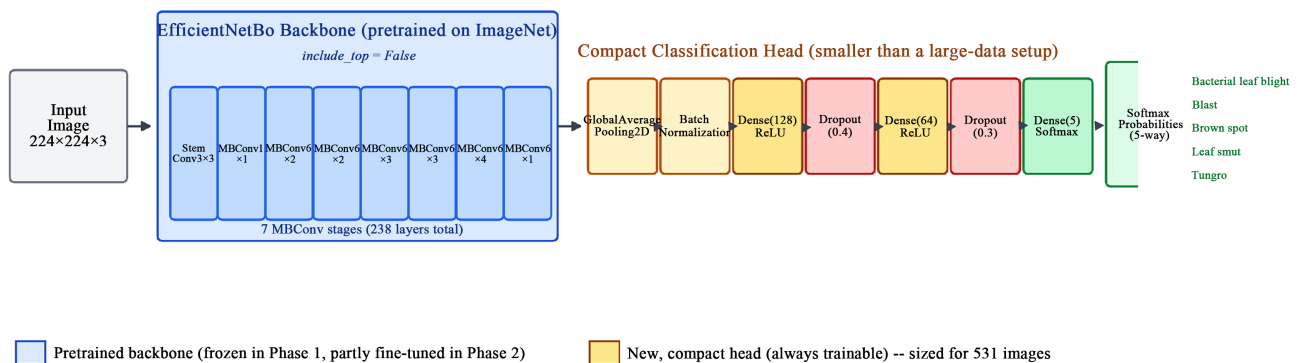


Figure 4. Model architecture: EfficientNetB0 backbone with a compact classification head sized for this dataset.

3.4. Two-Phase Training

Training proceeds in two phases (Figure 5). In Phase 1, the EfficientNetB0 backbone is kept frozen and only the new classification head is trained for up to 50 epochs at a learning rate of 0.001, EarlyStopping in practice halting Phase 1 at epoch 32, monitored with EarlyStopping (patience = 15, restoring the best weights), ModelCheckpoint, ReduceLROnPlateau (halving the learning rate if validation loss plateaus for 5 epochs), and CSVLogger. In Phase 2, the last 20 of Effi-

cientNetB0's 238 layers are unfrozen and the whole network is fine-tuned for up to 50 more epochs, EarlyStopping again halting training at epoch 48 at a much smaller learning rate (5×10^{-5}), using the same class of callbacks. This conservative unfreeze depth keeps most of the pretrained backbone untouched, limiting how much of it can overfit to the small number of training images available per class. After both phases complete, the notebook compares the best validation accuracy reached in Phase 1 against Phase 2 and retains whichever checkpoint performed better. The network is trained to minimise categorical cross-entropy loss using the Adam optimiser, which combines momentum with adaptive per-parameter learning rates.



Figure 5. Phase 1 (frozen backbone) versus Phase 2 (partial fine-tuning)—a conservative fine-tuning approach chosen for this dataset's small size.

Because the dataset's classes are not evenly represented, a balanced class-weight dictionary is computed from the training split using scikit-learn's `compute_class_weight` ("balanced", ...) function and passed into the `class_weight` argument of `model.fit()` in both training phases. This increases the loss contribution of under-represented classes such as Leaf smut during training, discouraging the model from achieving seemingly good overall accuracy simply by favouring the majority classes.

3.5. Evaluation Metrics

Accuracy, precision, recall and F1-score are monitored on the validation split

during development, where they guide checkpoint selection, and the same metrics are then computed unchanged on the held-out test split for a final, unbiased read of performance; only the test-split values are reported in Section 4. Accuracy is the fraction of all predictions that were correct, but can be misleading when classes are imbalanced, as they are here, since a model could achieve deceptively high accuracy by leaning toward the majority classes. Precision ($TP/(TP + FP)$) and recall ($TP/(TP + FN)$) are therefore also reported, together with their harmonic mean, the F1-score. Because there are five classes, these metrics are averaged across classes using weighted averaging, and a confusion matrix is used to visualise which specific pairs of diseases the model tends to confuse with one another.

3.6. Rule-Based Weather Risk Expert System

Independently of the CNN, a rule-based weather assessment function implements a forward-chaining IF-THEN inference engine with five disease-specific rule groups of three rules each: BB-1/2/3 for Bacterial leaf blight, BL-1/2/3 for Blast, BS-1/2/3 for Brown spot, LS-1/2/3 for Leaf smut, and TG-1/2/3 for Tungro. Within each group, one rule tests temperature, one tests relative humidity and one tests rainfall. **Table 3** summarises the agro-meteorological profile of each disease that motivates these rules, and **Table 4** gives the complete operational specification of all fifteen: the variable tested, the band at which the rule contributes its full weight, the band at which it contributes a reduced weight, and the agronomic source supporting each threshold.

Each rule is a graded three-way test rather than a binary one. If the observed value falls inside the primary band most favourable to that disease, the rule contributes its full weight; if it falls inside an adjacent secondary band, the rule contributes a reduced weight, reflecting conditions that are conducive but not optimal; otherwise the rule contributes nothing. The three bands are mutually exclusive and jointly exhaustive, so exactly one outcome is selected per rule and every possible reading is covered. In each case the engine records a human-readable explanation of why the rule did or did not fire.

Only the rule group belonging to the disease predicted by the classifier is evaluated; the other four groups are not consulted. The risk score is the unweighted arithmetic sum of the weights contributed by that group's three rules, and is then mapped to a risk level: HIGH at 70 or above, MODERATE from 40 to 69, and LOW below 40. Because the three full weights within each group sum to at most 100 by construction, the score is bounded above by 100 without any clipping being required; a defensive cap at 100 is nevertheless applied in the implementation. The maxima are not identical across diseases: Bacterial leaf blight, Blast, Brown spot and Leaf smut can each reach 100 under fully favourable conditions, whereas Tungro's three full weights (30, 30 and 25) sum to 85. A Tungro case therefore reaches the HIGH band, but never the top of the 0 - 100 scale; the scores should accordingly be compared across bands rather than read as directly commensurable point

values between diseases.

Each risk level is paired with a specific treatment advisory, for example applying Tricyclazole or Isoprothiolane fungicide and avoiding excess nitrogen under HIGH blast risk, or applying a recommended insecticide against the green leafhopper vector under HIGH Tungro risk. The classifier's confidence score is reported to the user alongside the risk assessment but is deliberately not an input to the risk calculation: the two modules remain independent, so that a weather-driven risk level cannot be inflated or suppressed by the classifier's certainty about the visual diagnosis. The central advantage of this rule-based approach over a neural network for this part of the task is interpretability: every risk score produced can be traced back to the exact rule or rules that fired, which matters for an advisory tool that a farmer or extension officer needs to trust and act on, directly echoing the transparency gap identified by farmer interviews in coastal Bangladesh [13].

The provenance of the thresholds themselves is as follows. The temperature and humidity bands are adapted from published rice disease epidemiology and from the two rule-based rice expert systems that informed this rule base [17] [18], and are consistent with the agro-meteorological profiles summarised in Table 3. The rainfall bands are author-defined operational values: accumulated rainfall in millimetres is not a standard epidemiological threshold variable, and these bands were chosen to represent the accumulation levels at which splash dispersal and prolonged leaf wetness become likely under Bangladeshi field conditions. This provenance applies rule by rule: every temperature and humidity band in Table 4 is adapted from those sources, and every rainfall band is an author-defined operational value.

Table 3. Agro-meteorological risk factors for the five diseases.

Disease	Causal Organism	Typical Symptoms	Favourable Weather Pattern
Bacterial leaf blight	<i>Xanthomonas oryzae</i> (bacterium)	Water-soaked lesions from leaf tips/margins turning yellow-to-grey and drying out.	Warm (28°C - 35°C), high humidity, standing water/heavy rainfall.
Blast	<i>Magnaporthe oryzae</i> (fungus)	Diamond-shaped grey-centred lesions with brown margins on leaves and nodes.	Cool nights (20°C - 28°C), high humidity (≥90%), long dew periods.
Brown spot	<i>Bipolaris oryzae</i> (fungus)	Small, oval, brown lesions with grey centres across the leaf blade.	Prolonged leaf wetness, moderate temperature, nutrient-deficient soil.
Leaf smut	<i>Entyloma oryzae</i> (fungus)	Small, angular, black-to-dark-brown raised spots, mostly on older leaves.	Cooler, persistently humid conditions later in the growing season.
Tungro	Rice tungro virus, green leafhopper vector	Yellow-orange leaf discolouration, stunted growth, reduced tillering.	Warm (25°C - 32°C), humid conditions favouring leafhopper activity.

Table 4. Complete specification of the fifteen weather rules. Each rule contributes its full weight inside the primary band, a reduced weight inside the secondary band, and nothing otherwise. Maximum attainable score per disease: Bacterial leaf blight 100, Blast 100, Brown spot 100, Leaf smut 100, Tungro 85. Risk bands: HIGH ≥ 70 , MODERATE 40 - 69, LOW < 40 . T = temperature in $^{\circ}\text{C}$, H = relative humidity in %, R = accumulated rainfall in mm.

ID	Disease	Variable	Primary band (full weight)	Secondary band (reduced weight)	Rationale recorded by the engine
BB-1	Bacterial leaf blight	Temperature	$28^{\circ}\text{C} \leq T \leq 35^{\circ}\text{C}$ (+35)	$T > 35^{\circ}\text{C}$ (+20)	High spread risk
BB-2	Bacterial leaf blight	Humidity	$H \geq 80\%$ (+35)	$70\% \leq H < 80\%$ (+20)	Strongly favours the disease
BB-3	Bacterial leaf blight	Rainfall	$R \geq 150$ mm (+30)	$80 \leq R < 150$ mm (+15)	Water splash spreads the bacterium
BL-1	Blast	Temperature	$20^{\circ}\text{C} \leq T \leq 28^{\circ}\text{C}$ (+35)	$28^{\circ}\text{C} < T \leq 32^{\circ}\text{C}$ (+15)	Optimal for the blast fungus
BL-2	Blast	Humidity	$H \geq 90\%$ (+40)	$80\% \leq H < 90\%$ (+20)	Ideal for spore germination; long dew period
BL-3	Blast	Rainfall	$50 \leq R \leq 150$ mm (+25)	$R > 150$ mm (+10)	Intermittent wetting and drying favours sporulation
BS-1	Brown spot	Temperature	$20^{\circ}\text{C} \leq T \leq 30^{\circ}\text{C}$ (+35)	$30^{\circ}\text{C} < T \leq 35^{\circ}\text{C}$ (+15)	Optimal for the fungus
BS-2	Brown spot	Humidity	$H \geq 85\%$ (+40)	$70\% \leq H < 85\%$ (+20)	Ideal for spore germination
BS-3	Brown spot	Rainfall	$R \geq 100$ mm (+25)	$50 \leq R < 100$ mm (+12)	Extends leaf wetness
LS-1	Leaf smut	Temperature	$20^{\circ}\text{C} \leq T \leq 28^{\circ}\text{C}$ (+40)	$28^{\circ}\text{C} < T \leq 32^{\circ}\text{C}$ (+15)	Optimal for leaf smut
LS-2	Leaf smut	Humidity	$H \geq 90\%$ (+35)	$80\% \leq H < 90\%$ (+20)	Optimal for spore release
LS-3	Leaf smut	Rainfall	$R \geq 150$ mm (+25)	$80 \leq R < 150$ mm (+12)	Rain disperses spores
TG-1	Tungro	Temperature	$25^{\circ}\text{C} \leq T \leq 32^{\circ}\text{C}$ (+30)	$T > 32^{\circ}\text{C}$ (+15)	Favours green leafhopper vector activity
TG-2	Tungro	Humidity	$H \geq 75\%$ (+30)	$60\% \leq H < 75\%$ (+15)	Favours leafhopper vector breeding
TG-3	Tungro	Rainfall	$50 \leq R \leq 200$ mm (+25)	$R > 200$ mm (+10)	Favours vector movement between fields

Two properties of the implementation should be stated explicitly. First, the rainfall term R is supplied to the assessment function as a single accumulated total in millimetres; the implementation does not fix the accumulation period internally, so R is taken from whatever weather record the user provides and that period must be held constant across queries for scores to remain comparable. Fixing this window explicitly, for example as a rolling seven-day total drawn from a live feed, is identified as future work in Section 5. Second, the advisory content in **Table 5** was compiled by the authors to accompany each risk level, drawing on standard rice disease management practice and on the control recommendations discussed in the expert systems that informed this rule base [17] [18]; the specific

products and dose rates should be verified against current national plant-protection recommendations before operational deployment.

Table 5. Treatment advisory issued at each risk level, for each disease.

Disease	Risk level	Advisory issued by the system
Bacterial leaf blight	HIGH	Immediate action required. Apply copper oxychloride at 3 g/L; drain flooded fields immediately; remove and burn infected plants; stop overhead irrigation; monitor the field every 3 days.
Bacterial leaf blight	MODERATE	Preventive treatment recommended. Apply a copper spray every 10 days; improve drainage; reduce nitrogen fertiliser; monitor every 7 days.
Bacterial leaf blight	LOW	Standard monitoring. Regular field monitoring is sufficient; maintain proper plant spacing; avoid excess nitrogen.
Blast	HIGH	Immediate fungicide application needed. Apply tricyclazole or isoprothiolane; avoid excess nitrogen fertiliser; drain the field intermittently to reduce humidity; remove volunteer rice plants and weed hosts; use resistant varieties in future planting.
Blast	MODERATE	Preventive fungicide recommended. Apply a preventive fungicide spray; reduce nitrogen application; monitor every 7 days.
Blast	LOW	Standard monitoring. Regular field monitoring is sufficient; maintain balanced fertilisation.
Brown spot	HIGH	Fungicide application urgent. Apply mancozeb at 2.5 g/L or iprodione; apply tricyclazole for systemic control; add potassium and silicon fertiliser; repeat the spray every 7 - 10 days.
Brown spot	MODERATE	Preventive fungicide recommended. Apply a mancozeb spray preventively; check soil potassium nutrition; monitor weekly.
Brown spot	LOW	Low-risk conditions. Maintain balanced fertilisation; avoid drought stress; scout the field every 10 days.
Leaf smut	HIGH	Immediate fungicide needed. Apply propiconazole or tebuconazole; apply carbendazim at 1 g/L as a soil drench; increase plant spacing for air circulation; avoid overhead irrigation; remove infected plant material.
Leaf smut	MODERATE	Preventive treatment advised. Apply a preventive fungicide spray; ensure good field drainage; increase plant spacing; monitor every 7 days.
Leaf smut	LOW	Standard care. Normal monitoring is sufficient; use resistant varieties where available; maintain proper spacing.
Tungro	HIGH	Immediate vector control required. Apply a recommended insecticide such as imidacloprid against the green leafhopper; rogue infected plants immediately; synchronise planting dates with neighbouring fields; avoid continuous rice cropping without a fallow period; use Tungro-resistant varieties where available.
Tungro	MODERATE	Preventive vector monitoring recommended. Monitor green leafhopper populations with light traps; apply insecticide only if the vector threshold is exceeded; monitor every 7 days.
Tungro	LOW	Standard monitoring. Routine vector population monitoring is sufficient; maintain synchronised planting practices.

3.7. Combined Prediction Pipeline

A predict-and-assess routine ties both modules together: it loads and preprocesses a single leaf image, obtains the CNN's predicted class and confidence, maps the internal class name to its rule-base key, runs the weather-risk assessment for that disease given a supplied temperature, humidity and rainfall reading, and renders a single composite report containing the input image, a confidence bar chart across all five classes, a risk gauge, the current weather inputs, the fired rules, and the final advisory text. The trained model, its class index mapping, and the class-name map are saved for reuse, and an interactive upload widget lets a user select a leaf photo and receive the full diagnosis-plus-advisory report. **Algorithm 1** summarises the end-to-end pipeline as pseudocode.

Algorithm 1. Rice leaf disease detection and weather-risk advisory pipeline.

```

Input:  raw rice-leaf image dataset D (5 classes); current weather reading W
Output: predicted disease label, confidence score, weather risk level, advisory text

1. D_train, D_val, D_test <- SPLIT(D)                // Section 3.1
2. D_train_aug <- AUGMENT(D_train)                  // Section 3.2
3. D_val_prep <- PREPROCESS(D_val) // resize/normalise only, no augmentation
4. D_test_prep <- PREPROCESS(D_test) // resize/normalise only, no augmentation
5. model <- EfficientNetB0_BACKBONE(pretrained = ImageNet)
6. model.head <- BUILD_CLASSIFICATION_HEAD(num_classes = 5) // Section 3.3
7. w <- COMPUTE_CLASS_WEIGHTS(D_train)              // Section 3.4
8. FREEZE(model.backbone)
9. model <- TRAIN(model, D_train_aug, D_val_prep, phase = 1, class_weights = w)
10. UNFREEZE(model.backbone, portion = partial)      // last 20 of 238 layers
11. model <- TRAIN(model, D_train_aug, D_val_prep, phase = 2, class_weights = w)
12. model <- SELECT_BEST_CHECKPOINT(phase1, phase2)   // by validation accuracy
13. metrics <- EVALUATE(model, D_test_prep)          // Section 3.5
14. procedure PREDICT_AND_ASSESS(image, W):
15.   label, confidence <- model.PREDICT(image)
16.   risk_level, fired_rules, advisory <- RULE_BASED_ASSESS(label, W) // Section 3.6
17.   return label, confidence, risk_level, advisory // Section 3.7
18. end procedure

```

Figure 6 shows a representative output of the combined pipeline on a single held-out leaf image. The classifier predicts Bacterial leaf blight with a confidence of 100.0%, with every other class scoring below 0.1%. The panel headed Inferred Environmental Conditions reports the agro-meteorological profile the interface associates with the predicted disease, rather than a single instantaneous measurement; for Bacterial leaf blight it shows 28°C - 35°C, 80% - 95% relative humidity and 150 - 300 mm accumulated rainfall. The temperature range is exactly rule BB-1's primary band; the humidity and rainfall ranges are displayed with an upper bound for readability, whereas rules BB-2 and BB-3 are open-ended above ($H \geq 80\%$ and $R \geq 150$ mm respectively, as **Table 4** specifies). The scoring itself operates on a specific reading supplied with the query. Taking the reading used in Section 3.6, 31.0°C, 86% humidity and 170 mm rainfall, which lies inside all three bands displayed, places all three Bacterial leaf blight rules in their primary bands, so each

contributes its full weight: the temperature falls inside the 28°C - 35°C high-spread-risk band (BB-1, +35), humidity exceeds the 80% threshold that strongly favours the disease (BB-2, +35), and accumulated rainfall exceeds the 150 mm threshold at which water splash is expected to spread the bacterium between plants (BB-3, +30). The three contributions sum to exactly 100, the maximum attainable for this disease, which is the HIGH risk level reported in the figure and which triggers the advisory shown there: apply copper oxychloride at 3 g/L, drain flooded fields, remove and burn infected plants, halt overhead irrigation, and monitor the field every three days. This illustrates the intended end-to-end behaviour described in Sections 3.1-3.7: a single leaf photograph and its associated weather context are converted into a disease label with class-wise confidence, an interpretable risk assessment grounded in that disease's agro-meteorological profile, and a concrete advisory output, rather than a bare classification label.

Rice Leaf Disease Detection Report

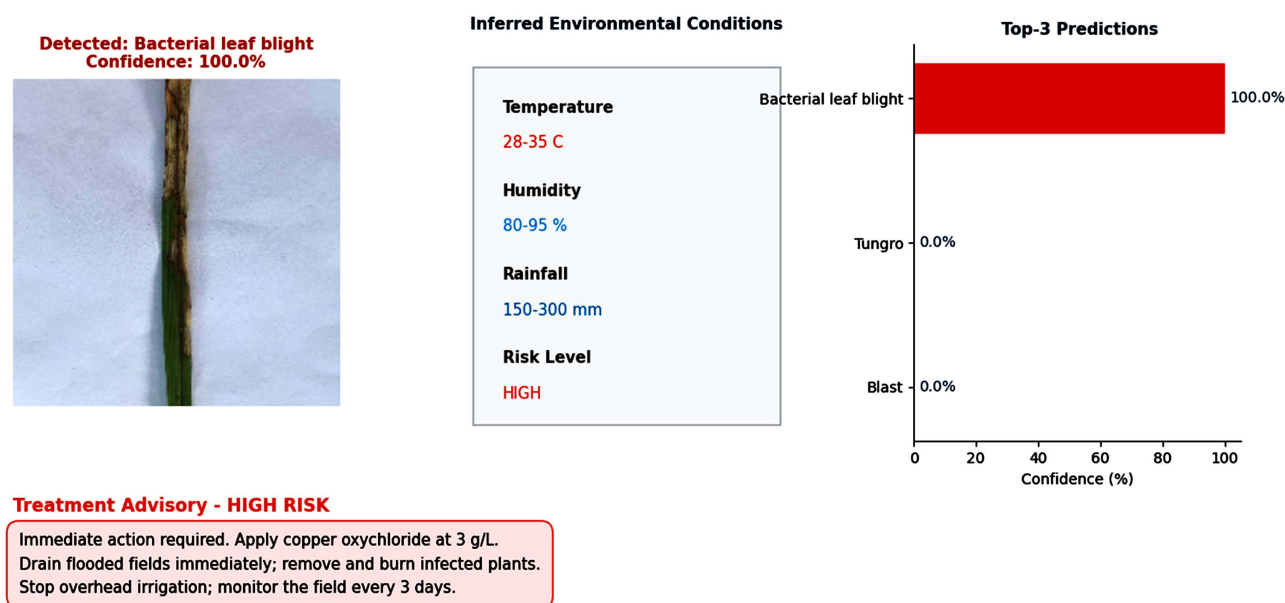


Figure 6. Combined diagnosis-and-advisory report produced by the deployed interface for a Bacterial leaf blight image. Left: the uploaded leaf with the detected class and its confidence. Centre: the inferred environmental conditions for that disease-temperature, humidity and rainfall-and the resulting risk level. Right: the top three class confidences. Below: the treatment advisory issued at that risk level.

4. Results and Discussion

4.1. Classification Performance

Training accuracy rises quickly during Phase 1, climbing from about 0.57 at epoch 1 to roughly 0.87 by epoch 5 and 0.91 by epoch 10, then settling into a 0.93 - 0.98 band from epoch 15 onwards. Validation accuracy rises faster still and tracks consistently above it, reaching about 0.94 by epoch 8 and peaking near 0.99 between epochs 17 and 20 before holding in the 0.97 - 0.98 range for the remainder of Phase

1, with a single dip to about 0.945 at epoch 22. Validation accuracy exceeding training accuracy throughout is expected in this configuration, because training batches are augmented and pass through active dropout layers while validation batches are unaugmented and evaluated with dropout disabled. The loss curves fall in step over the same period: training loss drops from 1.44 at epoch 1 to about 0.30 by epoch 6 and reaches its minimum of roughly 0.065 at epoch 30, while validation loss falls from 0.92 to its minimum of about 0.045 around epoch 18 and then drifts gently upward to roughly 0.08 by the end of Phase 1. Unfreezing the backbone at epoch 32 causes a brief, expected disruption: training accuracy drops sharply to about 0.875 at epoch 33 before recovering to fluctuate in the 0.90 - 0.95 range for the rest of training, and training loss spikes from 0.075 to roughly 0.39 as the newly trainable layers adjust, thereafter settling around 0.18 - 0.26. The validation curves are barely affected by the transition itself, holding near 0.97 immediately after the unfreeze and easing to about 0.94 - 0.96 by epoch 48, with validation loss continuing its slow upward drift from 0.07 to about 0.11 (**Figure 7**). Training stopped at epoch 48 under EarlyStopping. That the validation curves absorb the unfreeze without destabilising suggests the conservative 20-layer fine-tuning depth and the small Phase 2 learning rate achieved their intended purpose; the mild upward drift in validation loss across both phases, against flat validation accuracy, is the expected signature of a model becoming slightly more confident in its errors on a dataset this size. Because Phase 1 reached the higher peak validation accuracy of the two phases, the checkpoint-selection step described in Section 3.4 resolves in its favour.

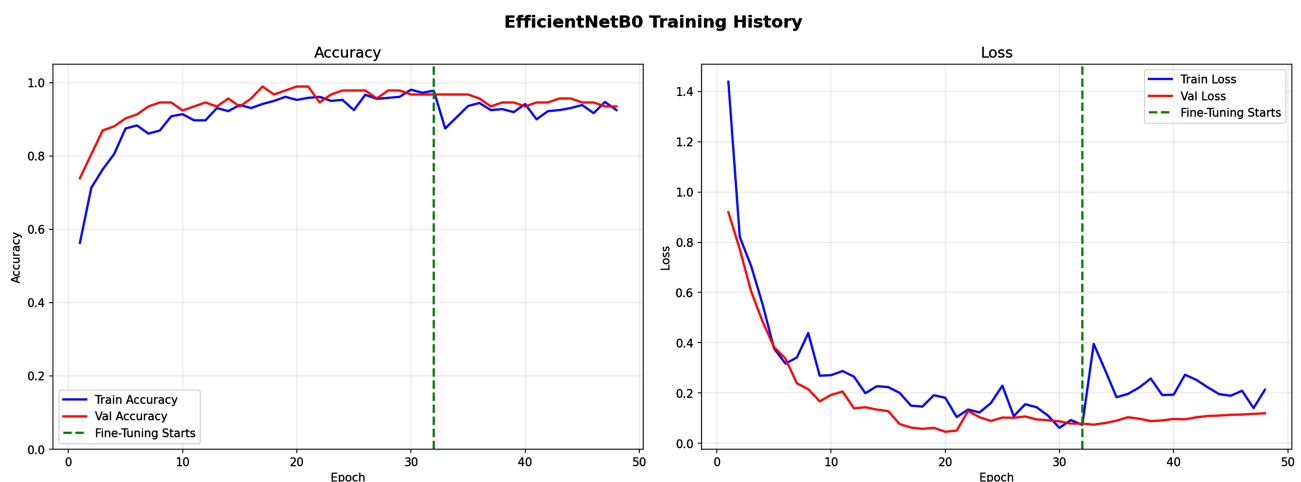


Figure 7. Training and validation accuracy and loss across both training phases; the dashed line marks the start of Phase 2 fine-tuning (epoch 32). Training ended at epoch 48 under EarlyStopping.

Table 6 reports the resulting accuracy, precision, recall and F1-score on the held-out test split. Validation-split metrics are not tabulated: the validation set guided model-checkpoint selection during training, so its scores are not an unbiased estimate of generalisation. The test-split numbers reported here are the sin-

gle, unbiased result of running the selected model once on data that had never influenced training or checkpoint selection.

Table 6. Accuracy, precision, recall, and F1-score on the held-out test split.

Metric	Held-Out Test Split (n = 80)
Accuracy	96.25%
Precision (weighted)	96.97%
Recall (weighted)	96.25%
F1-score (weighted)	96.40%

On the held-out test split, the model reaches 96.25% accuracy with a weighted F1-score of 96.40%. **Figure 8**'s confusion matrix shows strong class-wise performance across all five diseases. Blast is recognised perfectly (12 of 12 correct), and Brown spot, the largest class, is recognised almost as well (30 of 32 correct, two misclassified as Leaf smut). Leaf smut and Tungro, the smallest classes, are both recognised with 100% recall despite having only 6 and 12 test images respectively, though Leaf smut's precision is lower (75.00%) because the two misclassified Brown spot images land there. Bacterial leaf blight shows a small amount of confusion with Tungro (1 of 18 images), which is also why Tungro's precision (92.31%) sits slightly below its recall. Overall, these results indicate that the class-weighting strategy (Section 3.4) succeeded in preventing the model from favouring the majority Brown spot class at the expense of the smaller classes.

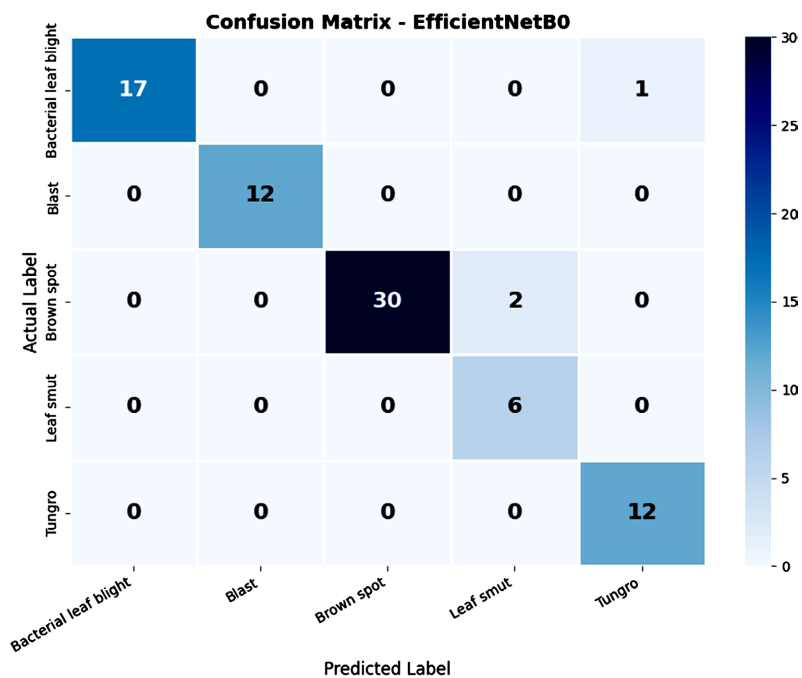


Figure 8. Confusion matrix for the EfficientNetB0 classifier on the 80-image held-out test split.

4.2. Comparison with Literature

For context alongside this work's 96.25% test accuracy, **Table 7** lists the accuracy figures reported by comparable transfer-learning studies from the literature review (Section 2), several of them using Bangladeshi data. These figures come from separate studies on their own datasets and evaluation protocols, so they are presented only as an approximate external point of reference rather than as a directly controlled comparison.

Table 7. Reported accuracy from comparable literature.

Study	Model	Reported Accuracy
RiceLeafBD [1]	EfficientNetV2 (transfer learning)	91.5%
Habiba <i>et al.</i> [12]	EfficientNetB3 (transfer learning)	91.0%
Khasim <i>et al.</i> [9]	Decision Tree (classical ML)	>97%
MobileNetV3/EfficientNet-B0 study [16]	MobileNetV3-Large	95.83%
TRiP (same 5-class taxonomy) [15]	SENet (transfer learning)	95.73%

4.3. Comparative Study against Other Backbone Architectures

Section 3.3 selected EfficientNetB0 over VGG16, ResNet50, and AlexNet on qualitative, parameter-efficiency grounds. To examine that choice empirically, VGG16, ResNet50, and AlexNet were each fine-tuned on the same 531-image, five-class dataset, initialised with ImageNet-pretrained weights and given a new classification head sized for five classes, and all four models were then evaluated on the same held-out 80-image test split under identical preprocessing. **Table 8** reports each model's accuracy, precision, recall, and F1-score alongside its parameter count.

Table 8. Comparative study against other backbone architectures.

Model	Params (M)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
EfficientNetB0 (this work)	4.0	96.25	96.97	96.25	96.40
VGG16	134.3	91.25	92.40	91.25	90.04
ResNet50	23.5	90.00	90.30	90.00	89.82
AlexNet	57.0	91.25	90.89	91.25	90.99

Note: all four models were evaluated on the same held-out 80-image test split, with identical preprocessing and no augmentation. Correct predictions: EfficientNetB0 77 of 80, VGG16 73 of 80, ResNet50 72 of 80, AlexNet 73 of 80. Precision, recall and F1-score are weighted averages across the five classes.

EfficientNetB0 scores highest on every metric while using by far the fewest parameters: 4.0 million against VGG16's 134.3 million, AlexNet's 57.0 million and ResNet50's 23.5 million. Because all four models were scored on the same 80 im-

ages, the margin is directly interpretable: EfficientNetB0 classified 77 of 80 correctly against 73 for VGG16, 73 for AlexNet and 72 for ResNet50, a lead of four to five images. This supports the accuracy-per-parameter argument made qualitatively in Section 3.3-the smallest of the four models is also the most accurate, not merely the cheapest to train-though on a test set of this size a four-image lead should be read as a consistent advantage rather than a precise effect size. The three baselines are closely matched on accuracy, with VGG16 and AlexNet tied at 91.25% and ResNet50 marginally behind at 90.00%, but they are not interchangeable. Ranking them by F1-score reverses the top two: AlexNet leads at 90.99%, ahead of VGG16 at 90.04%, even though VGG16 records the highest weighted precision of any baseline at 92.40%. The per-class results explain the reversal. VGG16 recognises the three largest classes perfectly-18 of 18 Bacterial leaf blight, 12 of 12 Blast and 32 of 32 Brown spot-but recovers only 2 of 6 Leaf smut and 9 of 12 Tungro images, so its errors fall entirely on the two smallest classes and its high precision is purchased by under-predicting them. AlexNet distributes its errors more evenly, and ResNet50 sits between the two, recovering 4 of 6 Leaf smut images. Leaf smut is the clearest point of separation across the whole comparison: EfficientNetB0 recovers all 6, ResNet50 4, AlexNet 3 and VGG16 only 2. This is the failure mode Section 3.4 identified as the reason for class weighting, and it is visible here as the difference between models that reach a similar aggregate accuracy on the back of the majority Brown spot class and a model that also handles the rarest class.

One limitation of this comparison should still be stated plainly. All four models were trained and evaluated once, with no repeated runs under different random seeds, and the test split contains only 80 images. Differences of one or two images between the baselines are therefore within the range that a different initialisation could plausibly reverse, and the ordering among VGG16, ResNet50 and AlexNet should be read as indicative. EfficientNetB0's lead over all three is larger and consistent across all four metrics, but it too rests on a single evaluation of a small test set.

4.4. Discussion of Hyperparameter Choices and Dataset Limitations

Every hyperparameter choice described in Section 3 was made with the underfitting/overfitting balance explicitly in mind. The compact Dense(128)→Dense(64) head reduces the parameter count that has to be learned from only a few hundred training images, lowering overfitting risk without meaningfully reducing capacity, since the pretrained EfficientNetB0 backbone already does most of the representational work. The stronger augmentation settings and the conservative 20-layer fine-tuning depth both trade a small amount of potential peak accuracy for a meaningfully lower risk of the network memorising the training images outright, which matters more on a dataset this size than squeezing out the last percentage point of training accuracy. Class weighting addresses a different failure mode:

without it, a model could reach seemingly reasonable overall accuracy while performing poorly on Leaf smut specifically, simply by leaning on the four larger classes-which is why per-class precision and recall, not accuracy alone, are essential for verifying that class weighting achieved its intended effect.

Because the dataset used in this work is relatively small, especially for classes such as Leaf smut (six test images), the resulting accuracy figures for those classes should be treated with some caution: a small test set makes it harder to be fully confident that measured performance will hold up on new, unseen images. This is a limitation of the underlying dataset rather than of the modelling choices made, and is addressed further in the future work discussion below.

5. Conclusions

This work developed a rice leaf disease detection system tailored to the agricultural context of Bangladesh, pairing a deep-learning image classifier with a separate, weather-aware risk advisory tool. The core of the system is a fine-tuned EfficientNetB0 model, chosen for its balance between accuracy and computational efficiency on a small, class-imbalanced dataset of 531 images. On the held-out test split, the trained classifier reached 96.25% accuracy with a weighted F1-score of 96.40% across the five disease classes, showing that transfer learning with a compact classification head can perform well even on a modest, real-world dataset, and scoring above VGG16, ResNet50 and AlexNet on the same held-out test split while using the fewest parameters. Alongside the classifier, an independent rule-based expert system evaluates current temperature, humidity and rainfall against agronomically grounded thresholds for each disease, producing a risk level and treatment advisory that complements the CNN's visual diagnosis. Together, these two components turn a raw leaf image into an actionable recommendation rather than a bare disease label.

The main contributions of this work are: an accurate small-dataset classifier reaching 96.25% test accuracy and a 96.40% weighted F1-score despite significant class imbalance; an independent weather-risk advisory system extended to five disease-specific rule groups that translate a temperature, humidity and rainfall reading into a HIGH/MODERATE/LOW risk level and a concrete treatment advisory, specified in full in **Table 4**; a combined diagnosis-and-advisory pipeline that packages the CNN's prediction and the weather-risk assessment into a single interactive, reproducible report; and a set of small-data training choices-class weighting, a compact classification head, and a conservative fine-tuning depth-deliberately selected to suit a modest dataset size rather than settings meant for a much larger one.

The work has several limitations. The image dataset totals only 531 images across five classes, with as few as 6 - 12 images per class in the test split, which limits how confidently the model's performance can be expected to generalise to rice varieties, field conditions, or regions of Bangladesh not represented in the data. The rule-based advisory module currently takes temperature, humidity and

rainfall as manually entered values rather than a live feed, so its usefulness in practice depends on the accuracy and timeliness of that manual input. Finally, the image classifier and the weather-risk system operate independently and are combined only after each produces its own output; the CNN does not take weather readings as an input feature, so the system does not yet learn jointly from combined visual and environmental patterns.

Future work could incorporate a Long Short-Term Memory network over a history of weather readings to forecast disease risk ahead of an outbreak rather than assessing only a single point-in-time reading; move toward true multimodal fusion, training a single model on both leaf images and weather readings jointly to capture interactions the current two-stage pipeline cannot; develop a mobile application to make the system accessible to farmers directly in the field; expand the dataset across additional regions, seasons and rice varieties to improve robustness, particularly for the smaller classes; fix the rainfall accumulation window explicitly as part of that feed; and connect the advisory module to a live weather-data source such as Bangladesh's BAMIS platform or the Bangladesh Meteorological Department's forecasts, in place of manually entered values, moving the system closer to the deployed early-warning tools already used for wheat blast in Bangladesh [14].

Data and Code Availability

The rice leaf image dataset analysed in this study is available at [https://github.com/mahmudul-30/thesis_work]. The complete training and inference pipeline, including data preparation, model training, evaluation, and the rule-based weather-risk advisory module, is available at [https://github.com/mahmudul-30/thesis_task]. The trained model weights and the class-name mapping file used to generate the results in Section 4 are included in the same repository.

Acknowledgements

This paper is based on the authors' undergraduate thesis (CSE-480) submitted to the Department of Computer Science and Engineering, Jahangirnagar University, Savar, Dhaka, Bangladesh, in August 2026. The authors thank their thesis supervisor for guidance throughout this work.

Author Contributions

Conceptualization, N.A.R. and M.M.H.; methodology, M.M.H.; software, S.A.M.; validation, N.A.R., M.M.H., and S.A.M.; formal analysis, M.M.H.; investigation, N.A.R.; resources, M.M.H.; data curation, S.A.M.; writing—original draft preparation, M.M.H.; writing—review and editing, N.A.R.; visualization, S.A.M.; supervision, M.M.H.; project administration, N.A.R.; funding acquisition, N.A.R. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Rimi, S.A., Chowdhury, M.D.J.U., Abdullah, R., Ahmed, I., Mim, M.A. and Rahman, M.S. (2025) Empowering Agricultural Insights: RiceLeafBD—A Novel Dataset and Optimal Model Selection for Rice Leaf Disease Diagnosis through Transfer Learning Technique. arXiv: 2501.08912. <https://doi.org/10.48550/arXiv.2501.08912>
- [2] Ahmed, K., Shahidi, T.R., Irfanul Alam, S.M. and Momen, S. (2019) Rice Leaf Disease Detection Using Machine Learning Techniques. arXiv: 2501.08912.
- [3] Matin, M.M.H., Khatun, A., Moazzam, M.G. and Uddin, M.S. (2020) An Efficient Disease Detection Technique of Rice Leaf Using AlexNet. *Journal of Computer and Communications*, **8**, 49-57. <https://doi.org/10.4236/jcc.2020.812005>
- [4] Saha, S. (2023) Automated Rice Leaf Disease Recognition System Using Different Textural and Intensity Moments Based Features. Master's Thesis, Khulna University of Engineering and Technology. <https://doi.org/10.13140/RG.2.2.28064.97280>
- [5] Lim, J.A., Yaacob, J.S., Mohd Rasli, S.R.A., Eyahmalay, J.E., El Enshasy, H.A. and Zakaria, M.R.S. (2023) Mitigating the Repercussions of Climate Change on Diseases Affecting Important Crop Commodities in Southeast Asia, for Food Security and Environmental Sustainability—A Review. *Frontiers in Sustainable Food Systems*, **6**, Article 1030540. <https://doi.org/10.3389/fsufs.2022.1030540>
- [6] Wang, Y., Rajkumar Dhamodharan, U.S., Sarwar, N., Almalki, F.A., Naith, Q.H., Sathiyaraj, R., *et al.* (2024) A Hybrid Approach for Rice Crop Disease Detection in Agricultural IoT System. *Discover Sustainability*, **5**, Article No. 99. <https://doi.org/10.1007/s43621-024-00285-4>
- [7] Kim, Y., Roh, J. and Kim, H.Y. (2017) Early Forecasting of Rice Blast Disease Using Long Short-Term Memory Recurrent Neural Networks. *Sustainability*, **10**, Article 34. <https://doi.org/10.3390/su10010034>
- [8] Bari, B.S., Islam, M.N., Rashid, M., Hasan, M.J., Razman, M.A.M., Musa, R.M., *et al.* (2021) A Real-Time Approach of Diagnosing Rice Leaf Disease Using Deep Learning-Based Faster R-CNN Framework. *PeerJ Computer Science*, **7**, e432. <https://doi.org/10.7717/peerj-cs.432>
- [9] Khasim, S., Rahat, I.S., Ghosh, H., Shaik, K. and Panda, S.K. (2023) Using Deep Learning and Machine Learning: Real-Time Discernment and Diagnostics of Rice-Leaf Diseases in Bangladesh. *EAI Endorsed Transactions on Internet of Things*, **10**, 1-8. <https://doi.org/10.4108/eetiot.4579>
- [10] Pinki, F.T., Khatun, N. and Islam, S.M.M. (2017) Content Based Paddy Leaf Disease Recognition and Remedy Prediction Using Support Vector Machine. 2017 20th International Conference of Computer and Information Technology (ICCIT), Singapore, 22-24 December 2017, 1-5. <https://doi.org/10.1109/iccitechn.2017.8281764>
- [11] Islam, A., Islam, R., Haque, S.M.R., Islam, S.M.M. and Khan, M.A.I. (2021) Rice Leaf Disease Recognition Using Local Threshold Based Segmentation and Deep CNN. *International Journal of Intelligent Systems and Applications*, **13**, 35-45. <https://doi.org/10.5815/ijisa.2021.05.04>
- [12] Habiba, U., Sadrin, R.I., Banu, A., Mamun, M.S.H., Hossain, R. and Fatema-Tuj-Johora (2026) Enhanced Rice Disease Recognition Using Transfer Learning. *Proceedings of the International Conference on Intelligent Data Analysis and Applications (IDAA 2025)*, Dhaka, 12-13 December 2025, 755-771.

- https://doi.org/10.2991/978-94-6239-664-7_52
- [13] Mousumi, M.A., Paparrizos, S., Ahmed, M.Z., Kumar, U., Uddin, M.E. and Ludwig, F. (2023) Common Sources and Needs of Weather Information for Rice Disease Forecasting and Management in Coastal Bangladesh. *NJAS: Impact in Agricultural and Life Sciences*, **95**, Article ID: 2191794. <https://doi.org/10.1080/27685241.2023.2191794>
 - [14] Krupnik, T.J., Cunha Fernandes, J.M., Vargas, F., Del Ponte, E.M., Hossain, K., Kamal, M., *et al.* (2025) A Weather-Forecast Driven Early Warning System for Wheat Blast Disease: User-Centered Design, Validation, and Scaling in Bangladesh and Brazil. *Climate Services*, **39**, Article ID: 100589. <https://doi.org/10.1016/j.cliser.2025.100589>
 - [15] Yuan, P., Xia, Y., Tian, Y. and Xu, H. (2024) Trip: A Transfer Learning Based Rice Disease Phenotype Recognition Platform Using Senet and Microservices. *Frontiers in Plant Science*, **14**, Article 1255015. <https://doi.org/10.3389/fpls.2023.1255015>
 - [16] Abiyyu, A.N. and Rahardi, M. (2026) Comparison of Transfer Learning Models MobileNetV3-Large and EfficientNet-B0 for Rice Leaf Disease Classification. *Journal of Applied Informatics and Computing*, **10**, 818-828. <https://doi.org/10.30871/jaic.v10i1.12033>
 - [17] Agus, F., Ihsan, M., Marisa Khairina, D. and Candra, K.P. (2024) ESforRPD2: Expert System for Rice Plant Disease Diagnosis. *F1000Research*, **7**, Article 1902. <https://doi.org/10.12688/f1000research.16657.3>
 - [18] Jearanaiwongkul, W., Anutariya, C., Racharak, T. and Andres, F. (2021) An Ontology-Based Expert System for Rice Disease Identification and Control Recommendation. *Applied Sciences*, **11**, Article 10450. <https://doi.org/10.3390/app112110450>
 - [19] Tan, M. and Le, Q.V. (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Honolulu, 23-29 July 2023, 6105-6114.
 - [20] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 26 June 26-1 July, 770-778. <https://doi.org/10.1109/cvpr.2016.90>