

An Android-Based Expert-in-the-Loop System for Date Palm Disease Diagnosis Using Calibrated Confidence Routing and Active Dataset Enrichment

Muntadher Kareem^{1*}, Raed Majeed¹

¹Department of Computer Science, College of Computer Science and Information Technology,
University of Sumer, Thi-Qar, Iraq

* Corresponding Author: muntadherkareem@uos.edu.iq

Phone number: 07823521526

Abstract

Date palm disease diagnosis in the field remains dependent on expert availability and subjective visual assessment. This paper presents an Android-based diagnostic system that integrates on-device deep learning inference with selective cloud escalation and structured expert feedback. A ShuffleNetV2 $\times 1.0$ student model (~ 5 MB), trained through knowledge distillation from an optimized ConvNeXt-Tiny teacher ($T = 2$, $\alpha = 0.7$), performs nine-class leaf disease classification directly on the mobile device via TensorFlow Lite. To address the overconfidence problem inherent in compact neural networks, post-hoc temperature scaling ($T_{cal} = 1.3976$) is applied to calibrate the softmax outputs before routing. A systematically optimized threshold ($\tau = 0.93$), selected through a 50-point sweep on calibrated confidence scores, governs whether each prediction is accepted locally or forwarded to the cloud-hosted teacher model on Google Cloud Run. The application provides two role-based interfaces: a Viewer interface for field workers offering GPS-tagged diagnosis, analytics, and data export; and an Expert interface enabling agricultural specialists to review uncertain cases and contribute validated labels to a structured Firebase dataset for future retraining. A three-tier caching mechanism ensures offline operability. Evaluation on 464 held-out test images yielded 99.14% system accuracy (Macro F1 = 0.9789, Balanced Accuracy = 0.9733), with 93.75% of cases resolved at the edge. The calibrated routing intercepted 8 of the 9 edge-model errors and corrected 6 through cloud re-inference, reducing total misclassifications from 9 to 4.

Keywords: date palm disease; expert-in-the-loop; confidence calibration; temperature scaling; knowledge distillation; TensorFlow Lite; edge-cloud routing; Firebase; Android; smart agriculture

نظام تشخيص لأمراض نخيل التمر قائم على نظام أندرويد مع إشراك الخبير في الحلقة،
باستخدام توجيه قائم على الثقة المُعايرة وإثراء نشط لمجموعة البيانات

منتظر كريم^{1*}، راند مجيد¹

¹قسم علوم الحاسوب، كلية علوم الحاسوب وتكنولوجيا المعلومات، جامعة سومر، ذي قار، العراق

الخلاصة

لا يزال تشخيص أمراض نخيل التمر في الحقل يعتمد على توفر الخبراء وعلى التقييم البصري الذاتي. تقدم هذه الورقة نظاماً تشخيصياً قائماً على نظام أندرويد يدمج الاستدلال العميق على الجهاز مع التصعيد الانتقائي إلى السحابة وتغذية راجعة منظمة من الخبراء. يقوم نموذج الطالب ShuffleNetV2 $\times 1.0$ (بحجم يقارب 5 ميغابايت)، والمدرب باستخدام تقطير المعرفة من نموذج المعلم ConvNeXt-Tiny المُحسن، بتنفيذ تصنيف أمراض الأوراق إلى تسع فئات مباشرة على الجهاز المحمول باستخدام TensorFlow Lite. ولمعالجة مشكلة الثقة المفرطة الكامنة في الشبكات العصبية الصغيرة، تم تطبيق معايرة لاحقة باستخدام مقياس الحرارة ($T_{cal} = 1.3976$) لضبط مخرجات Softmax قبل اتخاذ قرار التوجيه. كما تم اعتماد عتبة مُحسنة بشكل منهجي ($\tau = 0.93$)، تم اختيارها عبر اختبار 50 قيمة مختلفة على درجات الثقة المُعايرة، لتحديد ما إذا كان يتم قبول

التنبؤ محليًا (offline) أو إرساله إلى نموذج المعلم المستضاف سحابيًا عبر Google Cloud Run. يوفر التطبيق واجهتين قائمتين على الأدوار: واجهة "المستخدم" (Viewer) "العاملين في الحقل"، تتضمن تشخيصًا مرفقًا بإحداثيات GPS وتحليلات وتصدير بيانات؛ وواجهة "الخبير" (Expert) "التي تمكن المختصين الزراعيين من مراجعة الحالات غير المؤكدة وإضافة تسميات موثقة إلى مجموعة بيانات Firebase منظمة لاستخدامها في إعادة التدريب مستقبلاً. كما تضمن آلية تخزين مؤقت ثلاثية المستويات إمكانية العمل دون اتصال بالإنترنت. أظهرت نتائج التقييم باستخدام 464 صورة اختبارية مستقلة دقة نظام بلغت 99.14% (Macro F1 = 0.9789، Balanced Accuracy = 0.9733)، حيث تم حل 93.75% من الحالات على مستوى الحافة (Edge). وقد نجح نظام التوجيه المعايير في اعتراض 8 من أصل 9 أخطاء صادرة عن نموذج الحافة، وتم تصحيح 6 منها عبر إعادة الاستدلال السحابي، مما خفض إجمالي عدد الأخطاء من 9 إلى 4.

1. Introduction

As an economically important crop, date palm (*Phoenix dactylifera* L.) supports rural populations in the Middle East and North Africa, including the food security of the region [1-3]. The species also has a variety of fungal, physiological and insect-inflicted leaf disorders that exhibit overlapping symptoms, making it challenging to identify them accurately in the field even for experienced agronomists [4][5]. Traditional diagnosis relies on regular check-ups by a professional practitioner trained in the field of diagnosis - a method that is slow, expensive and difficult to scale.

Convolutional neural networks have shown high accuracy with plant diseases classification benchmarks [6-8], and knowledge distillation (KD) has additionally allowed the use of small models on resource-limited devices [9][10]. However, most published systems consider the diagnostic pipeline to be a one-pass classifier that has no feedback loop. This perspective ignores various realities of operation: no model can be resistant to ambiguous inputs that require human intervention; the training data obtained at a small set of farms cannot be representative of all field symptoms; and real-world deployment needs more than raw classification, such as location tracking, offline tolerance, and structured data management. Another issue is model calibration - compact distilled networks are more likely to give overconfident results that are not correlated with correctness probability [11], unwanted when confidence scores are used to make routing choices.

There is a theoretical foundation of human-in-the-loop and active learning paradigms [12][13] that can be used in integrating expert feedback, but such notions have not been implemented in mobile agriculture systems frequently. Current crop disease applications rely fully on cloud connections [14], do not involve expert verification [14][15], or do not have structured data pipelines for retraining.

The paper introduces PalmDiag, an Android-powered diagnostic framework of date palm leaf diseases that combines on-the-field inference, routing of confidence based on a calibration, and expert-based dataset augmentation in a single platform. We use a ShuffleNetV2 $\times 1.0$ student model [17] that is distilled on an optimized ConvNeXt-Tiny teacher [18] ($T = 2$, $\alpha = 0.7$) and run on TensorFlow Lite to do local nine-class classification. The softmax outputs are calibrated by post-hoc temperature scaling ($T_{cal} = 1.3976$) and routing choice is controlled by a threshold ($\tau = 0.93$) which is determined by systematic

sweep. Firebase offers the record-store to store the records and a Validated_Expert_Dataset directory containing labels validated by experts to be re-trained later.

The novelty of this work is the combination, not previously reported for date palm diagnosis, of three capabilities in one field application: routing between the edge and the cloud using temperature-calibrated confidence instead of raw softmax scores; an in-app expert-review loop that saves validated labels into a class-structured dataset for retraining; and domain-tuned distillation that produces a 5 MB model running fully offline. Earlier mobile systems offer none of these together.

Contributions are: (1) a dual-interface mobile application that offers field diagnosis with GPS tagging, analytics, and data export with expert review and labelling, (2) a calibrated confidence routing mechanism grounded in temperature-scaled probabilities rather than raw softmax outputs, (3) an expert-validated dataset enrichment pipeline on Firebase Cloud Storage organized by class for direct retraining use, (4) a three level caching for offline queue and conditional re-routing, and (5) system-level evaluation on 464 images demonstrating 99.14% accuracy with a reduction from 9 edge-only errors to 4 through calibrated routing.

2. Related Work

2.1 Mobile Deep Learning for Plant Disease Diagnosis

Mohanty et al. [19] were one of the first to prove that CNN-based classification can work on leaf image and the later reviews [6-8] prove the validity of deep learning as the prevailing paradigm in automated plant disease detection. Mobile deployment MobileNetV3 [20] and EfficientNet-Lite [21] are lightweight models that have been deployed to smartphones without an active internet connection, with server-dependent architectures also being investigated. ShuffleNetV2 [17] represents a positive trade-off between accuracy and latency through the substitution of a usual convolution with channel-split and channel-shuffle operations and is thus well adapted to real-time mobile usage. However, the majority of published mobile systems concentrate only on classification accuracy and do not consider the operational side of the deployment such as how to deal with uncertain predictions, how field data is processed or how the model can be refined with time passing by.

2.2 Knowledge Distillation for Edge Deployment

Knowledge distillation, coined by Hinton et al. [9], learns a small student network that tries to recreate the softened output distribution of a bigger teacher. Huang et al. [10], and others, have applied this technique to plant disease classification, showing that distilled models can be as accurate as teachers and do so at a fraction of the computational rate. One of the most frequently used practical considerations is that the distillation hyperparameters, i.e. the softening temperature T and the loss interpolation weight α , are usually configured to literature defaults without task specific fine-tuning. We systematically grid searched 15 combinations of T and α in this work, and found that $T = 2$ and 0.7 were the best parameters to use in our nine-class date palm task, which is inconsistent with the common defaults of $T =$

4-20. The comparatively small optimal temperature indicates that in smaller, domain-specific data sets, it can be more helpful to retain sharper inter-class differences in the soft labels than the highly-smooth distributions popular in large-scale benchmarks.

2.3 Confidence Calibration and Edge-Cloud Routing

Guo et al. [11] showed that the modern deep networks are often poorly calibrated, giving confidence values that either over- or underestimate the true likelihood of correctness. A post-hoc calibration technique, which learns a single temperature parameter on a validation set, has become the de facto lightweight solution. In edge, cloud offloading architectures, confidence scores are used as the routing decision signal [22]. Uncalibrated scores will cause overconfident edge predictions to fall under the safety of re-evaluating the cloud, whereas well-calibrated routing will use the cases most likely to gain advantage of a stronger model selectively. In spite of its proven usefulness in safety-critical tasks, calibration has seldom been explored in deployed agricultural edge-AI systems. Our system uses temperature scaling, $T_{cal} = 1.3976$, trained on the validation partition, and uses a systematic 50-point search to find the routing threshold.

2.4 Expert-in-the-Loop Learning

Active learning [12] and human-in-the-loop frameworks [13] are not new concepts in machine learning, but have not been applied to deployed agricultural diagnostic tools. Yu et al. [22] have surveyed cloud-edge-device collaborative computing in smart agriculture, which has the potential but little implementation of feedback-based model improvement. We implement this idea in our system by making the expert reviewing process a part of the mobile application: the cases in question are marked as uncertain or escalated and sent straight to the specialized review, and the confirmed labels are logged into a structured Firebase storage folder (Validated_Expert_Dataset) by their class, creating a constantly growing dataset that can be periodically retrained.

Table 1 contrasts PalmDiag with representative prior systems. Earlier mobile and edge approaches target raw accuracy and do not model uncertainty, calibrate confidence, or let experts enrich the dataset; PalmDiag combines all of these with offline operation.

Table 1- Comparison of PalmDiag with representative prior mobile/edge plant-disease systems.

<i>Study</i>	<i>Platform</i>	<i>Calibrated routing</i>	<i>Confidence calibration</i>	<i>Expert-in-the-loop enrichment</i>	<i>Offline operation</i>
<i>Mohanty et al. [19]</i>	Server / desktop	No	No	No	No
<i>MobileNetV3 / EfficientNet-Lite [20], [21]</i>	Mobile (on-device)	No	No	No	Partial
<i>Huang et al. [10] (knowledge distillation)</i>	Edge (distilled)	No	No	No	No
<i>Yu et al. [22] (survey)</i>	Cloud-edge-device	Conceptual	No	No	n/a
PalmDiag (this work)	Android edge + cloud	Yes	Yes	Yes	Yes

3. System Design and Implementation

3.1 Overall Architecture

The system is structured in three communicating layers. The mobile edge layer operates on an Android platform and can capture images, perform on-device preprocessing, TFLite inference, calibration of confidence, and the routing decision. The optimized ConvNeXt-Tiny teacher model is deployed as a containerized FastAPI service on Google Cloud Run (region: Europe West 1, 2 vCPUs, 2 GiB RAM). The Firebase Realtime Database is used as a structured record storage layer in the backend and Firebase Cloud Storage is used to archive images and the expert-validated dataset. The three layers have a confidence-based routing protocol: when the calibrated edge confidence is equal or greater than the threshold $\tau = 0.93$, the prediction is accepted locally and a 320×320 thumbnail is uploaded to Firebase; otherwise, the full-resolution image is sent to the cloud API to be inferred by the teacher-model.

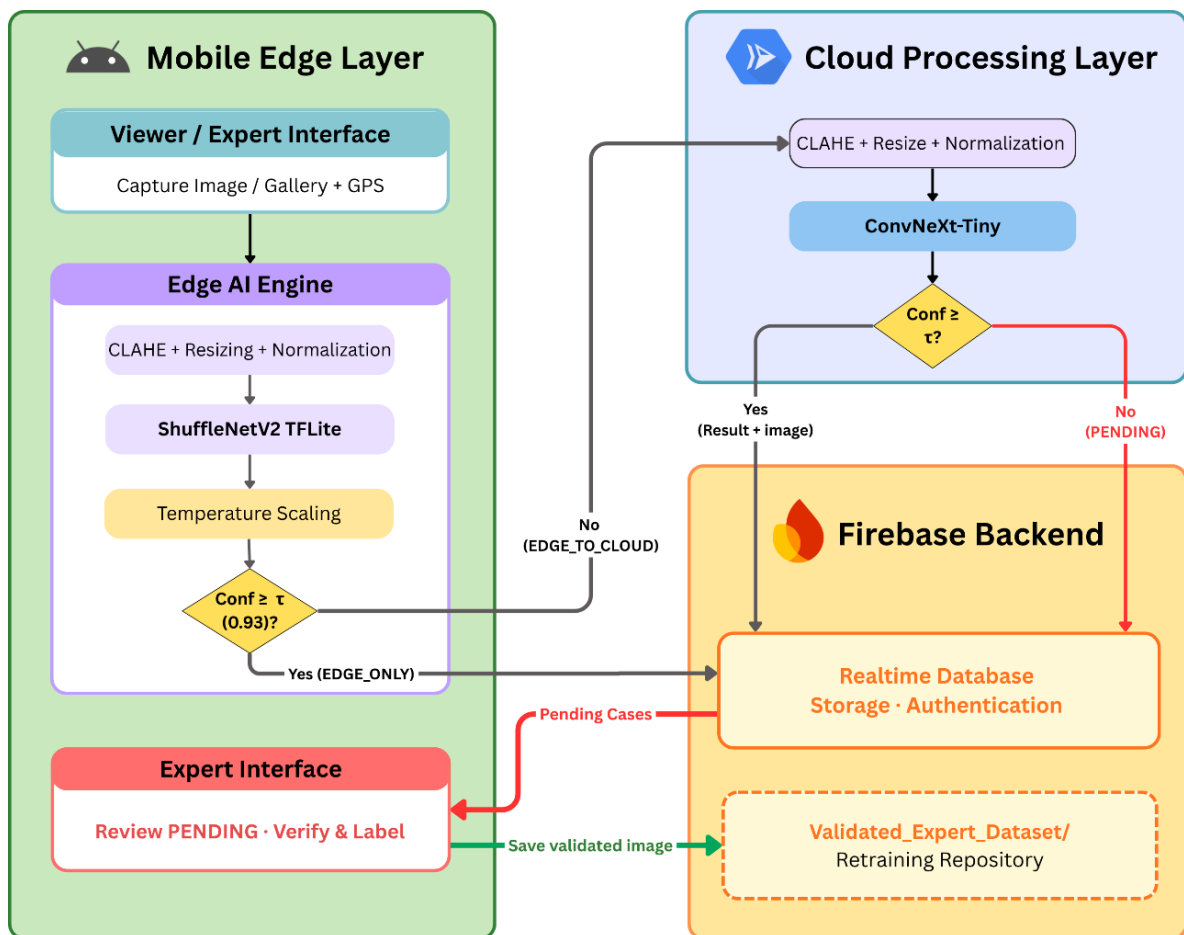


Figure -1 System architecture diagram.

3.2 Technology Stack

The application is natively developed in Kotlin using Jetpack Compose with Material Design 3 for the user interface and follows the Model-View-ViewModel (MVVM) pattern with StateFlow for reactive state management. Navigation is handled by Jetpack Navigation

Compose, image loading by Coil (AsyncImage), and HTTP communication with the cloud API by OkHttp3. The on-device inference engine relies on TensorFlow Lite 2.14 configured with 4 CPU threads. Geolocation is obtained through Google Play Services Location combined with AndroidX ExifInterface for EXIF GPS extraction. The application supports English and Arabic with RTL layout support, and provides light and dark themes. Table 2 summarizes the complete technology stack.

Table - 2 Technology stack of the PalmDiag Android application.

<i>Component</i>	<i>Technology</i>
<i>Language</i>	Kotlin
<i>UI Framework</i>	Jetpack Compose (Material Design 3)
<i>Architecture Pattern</i>	MVVM (ViewModel + StateFlow)
<i>Navigation</i>	Jetpack Navigation Compose
<i>ML Runtime</i>	TensorFlow Lite 2.14 (4 CPU threads)
<i>Image Loading</i>	Coil (AsyncImage)
<i>Maps</i>	Google Maps Compose
<i>HTTP Client</i>	OkHttp3
<i>Location</i>	Google Play Services Location + EXIF extraction
<i>PDF Generation</i>	Android Canvas-based PDF document
<i>Authentication</i>	Firebase Authentication (Email/Password)
<i>Database</i>	Firebase Realtime Database
<i>File Storage</i>	Firebase Cloud Storage
<i>Cloud Inference</i>	Google Cloud Run (FastAPI, europe-west1)
<i>Localization</i>	English, Arabic (RTL support)

3.3 Model Pipeline

The diagnostic model was produced through a multi-stage teacher-student pipeline. In the first stage, five ImageNet-pretrained CNN architectures were evaluated on a nine-class date palm leaf disease benchmark [4] containing 3,089 processed images. ConvNeXt-Tiny [18] attained the highest Macro F1 score (0.9616) and was selected as the teacher. In the second stage, six optimization strategies were applied to the selected teacher; class-imbalance-aware weighted cross-entropy loss achieved the strongest result (Macro F1 = 0.9789, Balanced Accuracy = 0.9733, Standard Accuracy = 99.14%). In the third stage, three lightweight student architectures were compared as distillation targets; ShuffleNetV2 $\times 1.0$ [17] yielded the highest baseline Macro F1 (0.9302) and was chosen. Finally, a grid search across 15 combinations of temperature $T \in \{2, 4, 6, 8, 10\}$ and soft-label weight $\alpha \in \{0.3, 0.5, 0.7\}$ identified $T = 2$ and $\alpha = 0.7$ as the optimal KD hyperparameters. Full distillation with these settings produced a student with Macro F1 = 0.9500, Balanced Accuracy = 0.9414, and Standard Accuracy = 0.9806. The resulting model (~5 MB) was exported to TFLite format and bundled as an asset in the Android application.

On-device preprocessing mirrors the training pipeline to avoid train-deploy mismatch. All captured images are processed with CLAHE in the L-channel of LAB color space (clip limit = 2.0, tile grid = 8×8) and then bicubic resized to 224×224 pixels and normalized using ImageNet statistics (mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]). The

preprocessed tensor is passed through the TFLite interpreter, which outputs a 9-element logit vector subsequently transformed by the softmax function.

3.4 Confidence-Based Hybrid Routing

Small-scale distilled models are vulnerable to making overconfident predictions, despite having inputs incorrectly classified [11]. To alleviate this, we use post-hoc temperature scaling: a single scalar parameter $Tcal$ is trained by minimizing the negative log-likelihood on the held-out validation set with L-BFGS minimization. The fitted value $Tcal = 1.3976$ rescales the raw logits, scaled by the softmax operation, which has the effect of spreading out the output distribution, and making the probabilities closer to the actual correctness rates.

The routing threshold τ was found by a systematic search on 50 evenly spaced values in the range [0.50, 0.99] of the calibrated confidence scores on the full test set. We chose the operating point of $\tau = 0.93$ as where the system accuracy levels off at 99.14% and where edge autonomy is traded off with the cost of cloud escalation. At this level 93.75 percent of images are resolved locally and 6.25 percent are sent to the cloud. The routing follows the following logic: The raw logits are divided by $Tcal$, followed by the softmax after on-device inference. In case the maximum probability that results is equal or greater than τ , the prediction is accepted (EDGE_ONLY); a 320x320 thumbnail and a metadata record are uploaded to Firebase. When the calibrated confidence is less than τ and the device is connected to the network, it will send the complete-resolution image to the Cloud Run endpoint and perform teacher-model inference (EDGE_TO_CLOUD). In case the cloud model also provides a confidence below the threshold, the case will be marked with reviewStatus = PENDING to be reviewed by experts. Without connectivity, low-confidence outcomes are stored to a local offline queue to be processed by the cloud. System accuracy also stays at 99.14% for every threshold from 0.90 to 0.99 (Macro F1 = 0.9789 for τ of 0.93 or higher), so the result barely depends on the exact value; we still note in Section 5 that tuning τ on a separate validation split is preferable.

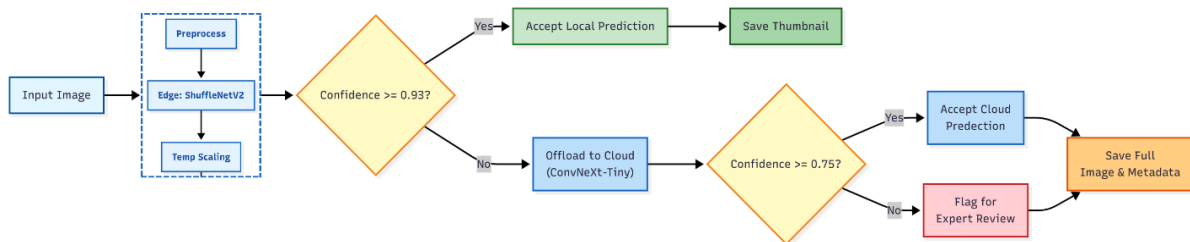


Figure -2 Routing decision flowchart

3.5 Viewer Interface

The Viewer interface is targeted at field workers and offers, a Dashboard with diagnostic statistics, weather information, and AI engine status, capture of images via camera or gallery with a quality guide, automatic GPS-tagging through EXIF extraction with a fallback to device-location, diagnostic results with predicted class, confidence, top-3

alternatives, severity, and treatment recommendations, a filterable Records List (All, Pending, Edge AI, Cloud AI, Expert, Rejected), Analytics charts covering disease distribution, daily timelines, and seasonal trends; an interactive Map View with disease-typed markers and clustering, full-text search, a Treatment Guide with symptom catalogs and management advice, data export in CSV, JSON, and PDF formats, and Settings for theme, language, and cache management.

3.6 Expert Interface and Active Dataset Enrichment

The Expert interface includes all Viewer functions plus an Expert Review tab with a badge counter showing the number of cases awaiting review. The workflow of the review goes as below. The specialist opens a pending case and reviews the full-resolution image, the AI prediction, and the top-3 confidence scores. The specialist then chooses the proper disease classification in a list of nine categories, and can optionally add some textual comments. On confirmation, the system updates reviewStatus to LABELED, and sets expertLabel to the class chosen by the expert with confidence = 1.0 (human ground truth), and moves the original image to Firebase Cloud Storage in the path Validated_Expert_Dataset/{disease_name}/{record_id}.jpg. This is a directory hierarchy with one folder each corresponding to the disease class, which forms a supervised training set that can be retrained directly on top of the original benchmark without further curation. The professional can also decline an image (reviewStatus = REJECTED) in case it is not informative enough or not of sufficient quality. The Expert interface is distinguished with a distinctive color scheme (Deep Teal, Indigo, and Gold accents) and permanent Expert Mode ribbon.

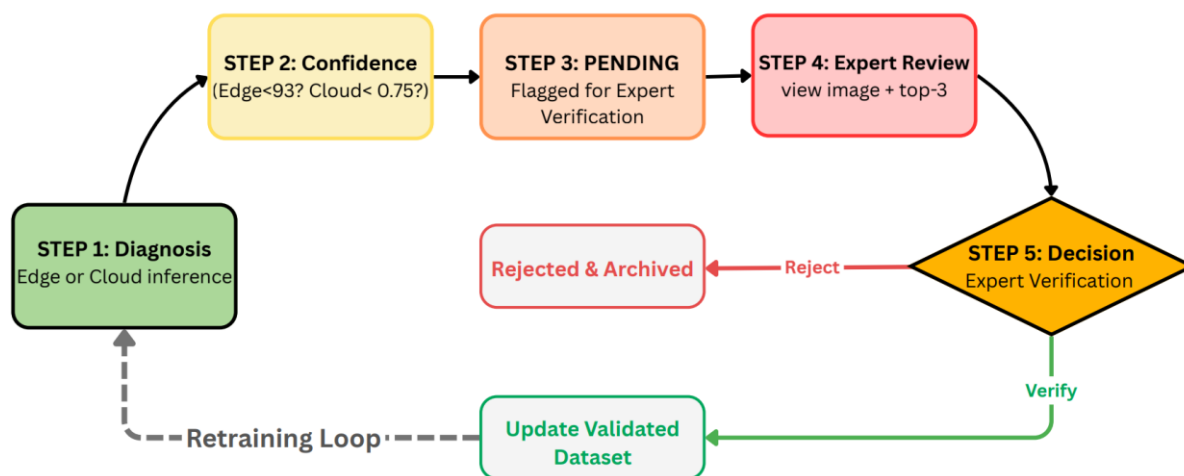


Figure -3 Expert review process

3.7 Diagnosis Record Data Model

Each diagnosis creates a record stored at diagnoses/{recordId} in Firebase Realtime Database. The schema is: recordId (UUID), timestampUtcMs, source (EDGE, CLOUD, or EXPERT_REVIEW), inferencePath (EDGE_ONLY or EDGE_TO_CLOUD), offloadReason, predictedClass, confidence, calibratedConfidence, gps (latitude and

longitude), image paths (storagePath, thumbPath, downloadUrl), reviewStatus (NOT_NEEDED, PENDING, LABELED, or REJECTED), expertLabel, and expertNotes. Records that need expert attention are written to a need_review/{recordId} branch for efficient retrieval. The Firebase Cloud storage directory is illustrated in Table 3.

Table 3- Firebase Storage structure.

<i>Directory Path</i>	<i>Content and Purpose</i>
<i>images/{disease}/{id}.jpg</i>	Full-resolution images uploaded through the cloud inference path
<i>thumbs/{id}.jpg</i>	320×320 thumbnails for efficient mobile loading and bandwidth conservation
<i>Validated_Expert_Dataset/{class}/{id}.jpg</i>	Expert-verified images organized by confirmed disease class for supervised retraining

3.8 Offline Resilience and Three-Level Caching

In the agricultural environments, reliable performance when connected intermittently is necessary. The application is using a three-tier caching strategy. The first level is a DiagnosisRecord object in-memory cache, which is maintained across screens within one session. The second layer stores the records in SharedPreferences as strings of JSON that are readable at the application restart. Firebase Realtime Database is the third tier and is the authoritative source of data when the device is on-line. Reads use the cascade Memory to Disk to Firebase and Writes go down the pyramid back to disk and memory. In offline mode, completed diagnoses (high-confidence edge results and low-confidence pending cases both) are stored to a local queue directory paired as a pair of files with recordId.json and recordId.jpg. The count of outstanding queues is shown as a sync banner on dashboard. Three events automatically cause queue processing: starting an application when a connection is detected, on a user-initiated pull-to-refresh, and when the network state changes. As part of synchronization, entries with offloadReason = LOW-CONFIDENCE are re-reported to the Cloud API with enhanced classification, instead of just uploaded, to ensure deferred predictions have access to teacher-model inferences.

4. Experimental Evaluation

4.1 Dataset and Experimental Setup

The analysis is based on the date palm leaf disease dataset of Namoun et al. [4], a collection of 3,089 processed images of nine classes: Potassium Deficiency (831), Magnesium Deficiency (566), Manganese Deficiency (415), Rachis Blight (402), Healthy (389), Fusarium Wilt (210), Black Scorch (106), Parlatoria Blanchardi (101), and Leaf Spots (69). The original raw images (608 in total) were taken on the date farms of ten different locations in Madinah region of Saudi Arabia using smartphones and a SLR camera and were filtered, cropped, and augmented through the use of the geometrical transformations. The processed data was stratified with stratified random sampling using a fixed seed of 42 to divide it into training (70%), validation (15%), and test (15%) items, with a test set of 464

images. Experiments at the system level described in this section were run on this held-out test partition.

The dataset is markedly imbalanced: the largest class (Potassium Deficiency, 831 images) has about twelve times the samples of the smallest (Leaf Spots, 69), with Parlatoria Blanchardi (101) and Black Scorch (106) also scarce. We address this with class-weighted cross-entropy during teacher training and by reporting Macro F1 and Balanced Accuracy, not accuracy alone; the stratified split also keeps every class present in each partition. At run time, calibrated routing escalates more minority-class cases to the cloud (Section 4.5) and the expert loop grows these classes over time. The split was made at the image level on the provider-augmented data; the augmentation was done by the dataset authors [4], and our CLAHE step was applied to all partitions equally. Because an image-level split cannot guarantee that augmented copies of one image stay in the same partition, a source- or farm-level split is noted as a limitation in Section 5.

4.2 Model Performance

Table 4- Model pipeline results across the teacher-student distillation stages.

<i>Configuration</i>	<i>Macro F1</i>	<i>Bal. Acc.</i>	<i>Std. Acc.</i>	<i>NLL</i>
<i>ConvNeXt-Tiny (Weighted CE)</i>	0.9789	0.9733	0.9914	0.0785
<i>ShuffleNetV2 Baseline</i>	0.9302	0.9200	0.9720	0.0984
<i>ShuffleNetV2 + KD ($T=2, \alpha=0.7$)</i>	0.9500	0.9414	0.9806	0.0904
Δ (KD vs. Baseline)	+0.0198	+0.0214	+0.0086	-0.0080

Table 4 shows the performance of the stages of the model pipeline. ConvNeXt-Tiny teacher, which was optimized using the class-weighted cross-entropy, and which was used to correct the imbalanced distribution of classes, showed a Macro F1 of 0.9789 and was used as the cloud inference model. Knowledge distillation enhanced the ShuffleNetV2 student on all metrics when compared to the standalone baseline: Macro F1 improved by 0.9302 to 0.9500, Balanced Accuracy by 0.9200 to 0.9414, and Standard Accuracy by 0.9720 to 0.9806. Such gains can be explained by the soft targets that the teacher gave in the training and which can portray information on similarity between classes not contained in hard labels.

Table 5 reports a cumulative ablation. Knowledge distillation raises the student from 0.9302 to 0.9500 Macro F1. Temperature scaling keeps the predictions, and so the accuracy, unchanged but halves the ECE (0.0142 to 0.0076), which is what makes the confidence threshold reliable. Calibrated edge-cloud routing gives the largest gain, reaching 0.9789 Macro F1 and 99.14% accuracy by sending the most uncertain cases to the teacher. The expert loop then handles the four residual high-confidence errors that no threshold can catch.

Table 5- Cumulative ablation of the PalmDiag components on the 464-image test set.

<i>Configuration (cumulative)</i>	<i>Macro F1</i>	<i>Balanced Acc.</i>	<i>Standard Acc.</i>	<i>ECE</i>
<i>Edge student, no distillation</i>	0.9302	0.9200	0.9720	n/a
+ <i>Knowledge distillation ($T = 2, \alpha = 0.7$)</i>	0.9500	0.9414	0.9806	0.0142
+ <i>Temperature scaling ($T_{cal} = 1.3976$)</i>	0.9500	0.9414	0.9806	0.0076
+ Calibrated edge-cloud routing ($\tau = 0.93$)	0.9789	0.9733	0.9914	n/a

4.3 Calibration Quality

Because routing depends on confidence, we assess calibration directly. Figure 4 shows the reliability diagrams before and after temperature scaling and Table 6 the metrics. The raw model is over-confident: its mean confidence (0.9916) exceeds its accuracy (0.9806), and the ECE is 0.0142. Fitting one temperature ($T_{cal} = 1.3976$) on the validation set lowers the mean confidence to 0.9843, halves the ECE to 0.0076, and lowers the top-label Brier score from 0.0157 to 0.0148, so the scores used for routing are far more trustworthy than raw softmax outputs.

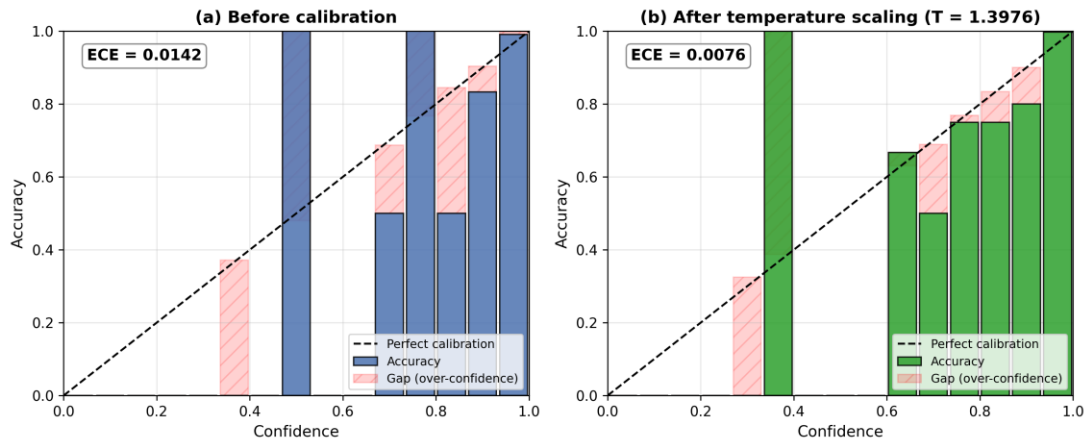


Figure -4 Reliability diagrams of the edge model before (a) and after (b) temperature scaling; the calibrated model lies closer to perfect calibration and roughly halves the ECE.

Table 6- Calibration metrics of the edge model before and after temperature scaling (test set, 15 bins; lower is better for ECE and Brier).

Metric	Before calibration	After temperature scaling
Expected Calibration Error (ECE)	0.0142	0.0076
Brier score (top-label)	0.0157	0.0148
Mean predicted confidence	0.9916	0.9843
Edge accuracy (reference)	0.9806	0.9806

4.4 System-Level Results

Table 7- System-level evaluation summary (464 test images, $\tau = 0.93$)

Metric	Value
Overall Accuracy	99.14% (460 / 464)
Macro F1-Score	0.9789
Macro Precision	0.9873
Balanced Accuracy	0.9733
Edge-Resolved Cases	93.75% (435 / 464)
Cloud-Escalated Cases	6.25% (29 / 464)
Edge Errors Intercepted	8 / 9 (88.9%)
Cloud-Corrected Errors	6 / 8 (75.0%)
Residual System Errors	4 (vs. 9 edge-only)
Mean Edge Latency	96.47 ms
Mean Cloud Round-Trip Latency	1057.56 ms
Total Cloud Bandwidth	773.68 KB (29 images)

Table 7 summarizes the system-level performance. The hybrid edge-cloud pipeline correctly classified 460 of 464 test images, yielding an accuracy of 99.14%. Of the 464 cases, 435 (93.75%) were resolved at the edge and 29 (6.25%) were forwarded to the cloud. The mean latency for edge-resolved cases was 96.47 ms, while cases that required cloud inference incurred a mean round-trip latency of 1057.56 ms, including network transfer and teacher-model processing. The total data transmitted to the cloud across all 29 escalated images was approximately 774 KB, a modest bandwidth footprint attributable to the JPEG compression of full-resolution leaf images.

4.5 Routing Effectiveness and Error Analysis

The edge model alone produced 9 misclassifications out of 464 (accuracy = 98.06%). The calibrated routing intercepted 8 of these 9 errors by detecting sub-threshold calibrated confidence and forwarding them to the cloud, where the teacher correctly reclassified 6, yielding a net reduction from 9 to 4 system errors. The single unintercepted error involved a Leaf Spots image misclassified as Magnesium Deficiency with a calibrated confidence of 0.9980, far above the threshold and therefore invisible to any confidence-based gate. This represents a fundamental limitation which is when a model is wrong and highly confident no threshold can intervene.

Figure 5 contrasts the confusion matrices of the edge model alone (a) and the full system (b): the nine edge errors fall to four after routing. The one error routing cannot catch (the Leaf Spots sample predicted as Magnesium Deficiency at 0.9980 confidence, noted above) is the high-confidence failure mode that a confidence gate cannot detect, which is why the expert loop forms a necessary third tier.

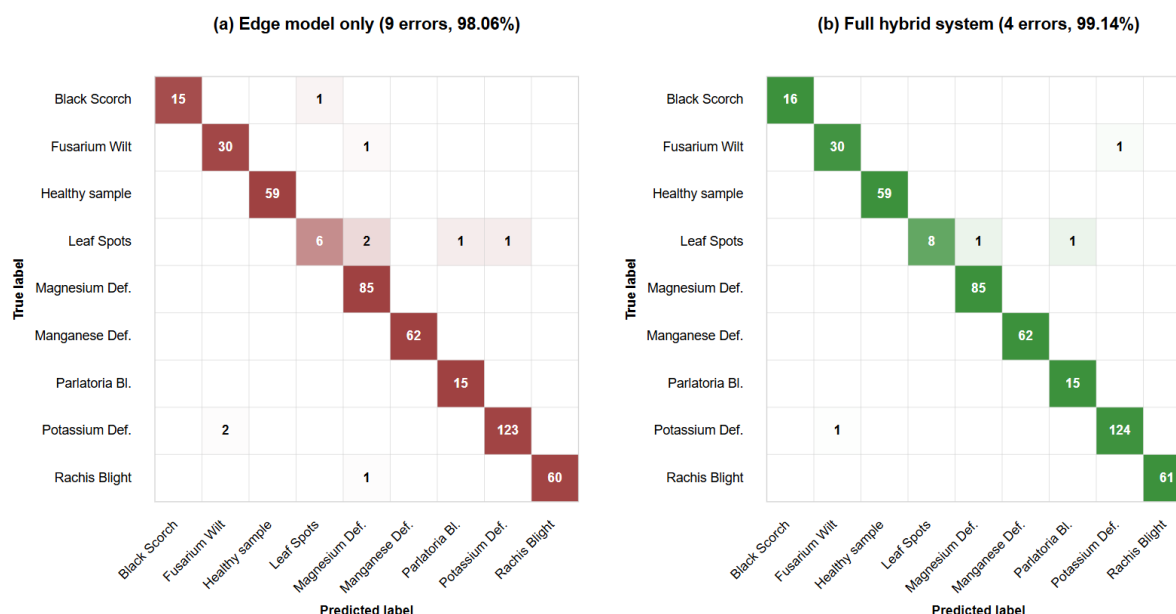


Figure -5 Confusion matrices of the edge model only (a, nine errors) and the full hybrid system (b, four errors) on the 464-image test set.

The four residual errors fall in two patterns. In three instances, the routing properly escalated the prediction, and the cloud teacher additionally created an erroneous label. The fourth case involved an overconfident error of the edge that avoided routing. By these failures, the expert review element is inspired: once the two automated models fail to label the image, a human specialist may still analyze the image and provide the appropriate label with the Expert interface.

The routing activity focused on under-represented and visually ambiguous classes. Leaf Spots had 6 of 10 test samples (60%) escalated, Fusarium Wilt 5 of 31 (16.1%), and Magnesium Deficiency 10 of 85 (11.8%). Classes exhibiting unique characteristics (Healthy, Manganese Deficiency, and Parlatoria Blanchardi) had zero escalations, indicating that calibrated routing only focuses on truly uncertain predictions.

Table 8- Per-class routing distribution and system F1 scores.

Class	N	Edge	Cloud	Prec.	Recall	F1
Black Scorch	16	15	1	1.0000	1.0000	1.0000
Fusarium Wilt	31	26	5	0.9677	0.9677	0.9677
Healthy	59	59	0	1.0000	1.0000	1.0000
Leaf Spots	10	4	6	1.0000	0.8000	0.8889
Magnesium Def.	85	75	10	0.9884	1.0000	0.9942
Manganese Def.	62	62	0	1.0000	1.0000	1.0000
Parlatoria Bl.	15	15	0	0.9375	1.0000	0.9677
Potassium Def.	125	119	6	0.9920	0.9920	0.9920
Rachis Blight	61	60	1	1.0000	1.0000	1.0000

The most difficult class is Leaf Spots ($F1 = 0.8889$), which can be explained by two factors: a very small sample size of tests (only 10 test samples and 69 samples in total) and a visual similarity with the symptoms of nutrient deficiency. The hybrid routing enhanced Leaf Spots recall at the edge alone of 0.60 to 0.80 at the system level by correcting three of four edge misclassifications by routing them to the cloud. This not only emphasizes the special importance of calibrated routing of minority classes, but also the significance of the expert enrichment pipeline: direct solicitation of more Leaf Spots images via the Validated_Expert_Dataset mechanism is the most obvious way to special purpose this class.

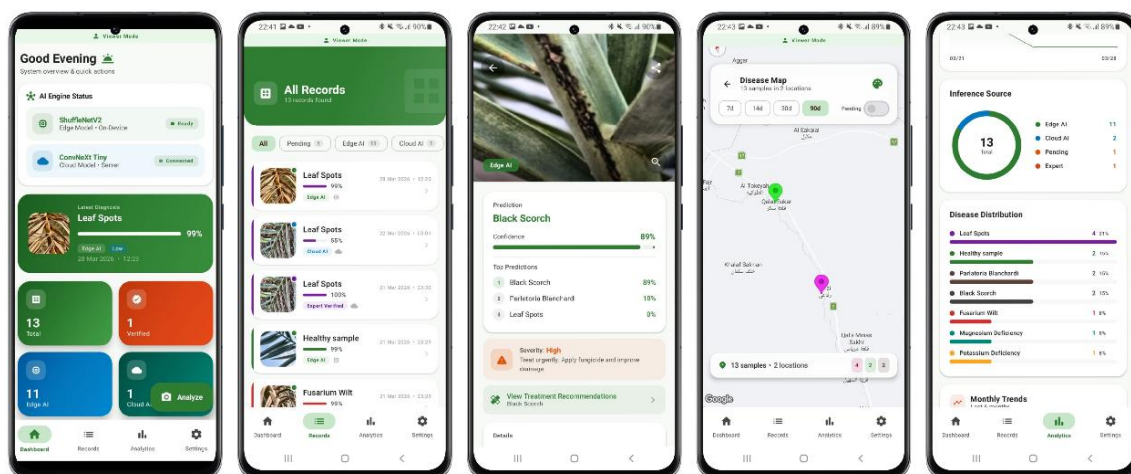


Figure 6. PalmDiag Viewer interface: dashboard, records list, diagnosis result, map view, and analytics.

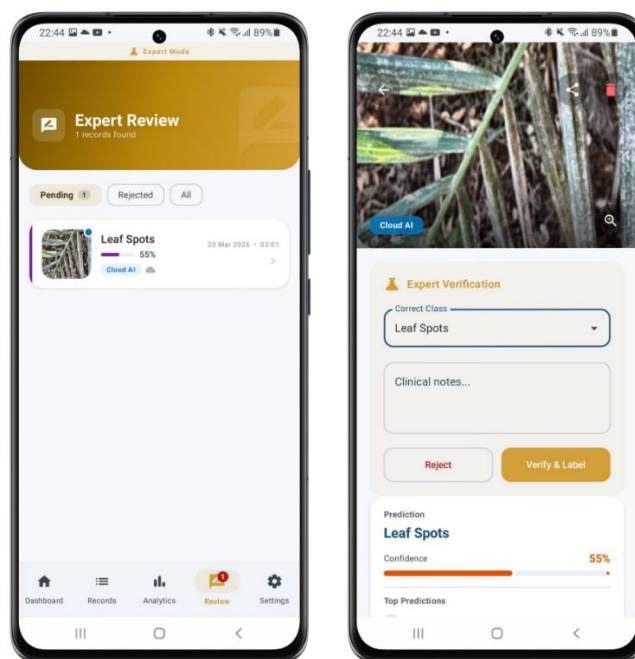


Figure 7- PalmDiag Expert interface: the review queue (left) and the expert verification screen (right).

5. Discussion

Results confirm that our system achieves strong diagnostic accuracy (99.14%) with operational capabilities that are not presented by classifier-only methods.

The error-reduction capability is based on the calibrated routing mechanism. ShuffleNetV2, in the absence of calibration, gives near-maximum softmax scores to false predictions [11]. The learned temperature ($T_{cal} = 1.3976 > 1.0$) smooths out the output distribution, increasing the difference between confident-correct and confident-wrong cases. This allowed the 0.93 threshold to capture 88.9% of edge errors and maintain 93.75% of predictions local. The trade-off is explicit: lowering τ catches more errors at the cost of increased cloud traffic; raising it preserves edge autonomy at the risk of missing misclassifications. The 50-point threshold sweep gives practitioners a performance curve from which to choose the operating point that fits their deployment constraints.

The dual-interface design bridges an existing gap in plant diagnosis apps in mobile. By embedding diagnosis and expert review in one application, the system creates a feedback loop: uncertain cases surface for review, expert corrections are recorded with provenance, and validated images accumulate in a class-organized directory ready for retraining. This is especially true considering the fact that the benchmark dataset [4] was only used on 10 farms and only 69 Leaf Spots images are included in this class- the one that the model performs the worst.

Several limitations can be identified. First, the benchmark was split at the image level over data the providers had already augmented, so augmented copies of one image may fall in different partitions and inflate the scores; a source- or farm-level split (not possible here, since per-image provenance is not released with the dataset) would be more conservative.

Second, the threshold τ was tuned on the evaluation set; the flat accuracy plateau from 0.90 to 0.99 makes this robust, but tuning τ on a separate validation split is preferable. Third, the data come from one region of Saudi Arabia, so external validation, for example on Iraqi farms, is needed to confirm generalizability. Fourth, a confidence gate cannot stop high-confidence wrong predictions; the expert loop reduces but does not remove this, and a field usability study with agronomists is still needed. These points set the roadmap for future work.

6. Conclusions

In this paper, PalmDiag, an Android-based date palm disease diagnosis system, has been introduced to integrate on-device deep learning inference, calibrated confidence-based cloud routing with structured expert feedback into a single mobile application. The combination of post-hoc temperature scaling ($T_{cal} = 1.3976$) and a systematically optimized routing threshold ($\tau = 0.93$) allowed the system to intercept most edge-model errors and perform local inference on 93.75% of inputs. The system was able to hit 99.14% in 464 held-out test images with nine disease classes (Macro F1 = 0.9789, Balanced Accuracy = 0.9733), and dropped the total misclassifications to 4 (hybrid system) compared to 9 (edge-only). The dual Viewer-Expert interface architecture provides an ever-expanding expert-verified dataset into the standard workflow of the application, providing a viable route to data-driven model enhancement across multiple deployment cycles. The three-tier caching architecture and offline queue ensure that the system remains functional in the connectivity-limited environments typical of agricultural field work.

Future work will pursue four directions: (1) formal usability studies with agricultural experts and extension workers to evaluate the diagnostic workflow under field conditions; (2) an automated retraining pipeline triggered when the volume of expert-validated images reaches a configurable threshold; (3) temporal disease progression monitoring through repeated GPS-tagged assessments of individual palm trees; and (4) cross-regional validation using datasets from Iraqi date palm farms to assess the generalizability of the trained models beyond the Saudi Arabian benchmark.

Ethical Statement

This article does not include any research involving humans or animals conducted by the authors. The dataset used was publicly available and properly cited.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgments

The authors are grateful to Namoun et al. [4] for making the date palm leaf disease dataset available publicly. The authors take full responsibility for the content of this publication.

References

- [1] The Ministry of Environment, Water and Agriculture (MEWA), "FAO Approves Saudi Arabia's Proposal to Declare 2027 the International Year of Date Palm," 2020. [Online]. Available: <https://mewa.gov.sa/en/MediaCenter/News/Pages/News201220.aspx/>. [Accessed: Apr. 1, 2026].
- [2] N. Al-Habsi, "Date Palm (*Phoenix dactylifera* L.) Fruit: Strategic Crop for Food Security, Nutritional Benefits, and Postharvest Quality," *Sustainability*, vol. 17, p. 7491, 2025.
- [3] A. Zaid and P. F. de Wet, "Botanical and Systematic Description of the Date Palm," in *Date Palm Cultivation*, FAO Plant Production and Protection Paper No. 156 Rev. 1, Rome: FAO, 2002.
- [4] A. Namoun, A. B. Alkhodre, A. A. Abi Sen, Y. Alsaawy, and H. Almoamari, "Dataset of infected date palm leaves for palm tree disease detection and classification," *Data in Brief*, vol. 57, p. 110933, 2024. <https://doi.org/10.1016/j.dib.2024.110933>.
- [5] M. El Bouhssini and J. R. H. Faleiro, *Date Palm Pests and Diseases Integrated Management Guide*. Beirut, Lebanon: International Center for Agricultural Research in the Dry Areas (ICARDA), 2018.
- [6] B. Tugrul, E. Elfatimi, and R. Eryigit, "Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review," *Agriculture*, vol. 12, no. 8, p. 1192, 2022.
- [7] M. Shoaib et al., "An Advanced Deep Learning Models-Based Plant Disease Detection: A Review of Recent Research," *Frontiers in Plant Science*, vol. 14, p. 1158933, 2023.
- [8] I. Pacal, I. Kunduracioglu, M. H. Alma, M. Deveci, S. Kadry, J. Nedoma, V. Slany, and R. Martinek, "A systematic review of deep learning techniques for plant diseases," *Artificial Intelligence Review*, vol. 57, p. 304, 2024. <https://doi.org/10.1007/s10462-024-10944-7>.
- [9] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, 2015.
- [10] Q. Huang, X. Wu, Q. Wang, X. Dong, Y. Qin, X. Wu, Y. Gao, and G. Hao, "Knowledge Distillation Facilitates the Lightweight and Efficient Plant Diseases Detection Model," *Plant Phenomics*, vol. 5, p. 0062, 2023. <https://doi.org/10.34133/plantphenomics.0062>
- [11] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in *Proc. Int. Conf. Machine Learning (ICML)*, Sydney, Australia, 2017, pp. 1321-1330.
- [12] B. Settles, *Active Learning Literature Survey*. Madison, WI, USA: University of Wisconsin-Madison, Computer Sciences Technical Report 1648, 2009.
- [13] R. Monarch, *Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-Centered AI*. Shelter Island, NY, USA: Manning, 2021.
- [14] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming," *IEEE Access*, vol. 7, pp. 156237-156271, 2019.
- [15] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep Learning in Agriculture: A Survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70-90, 2018.
- [16] P. Sharma, Y. P. S. Berwal, and W. Ghai, "Performance Analysis of Deep Learning CNN Models for Disease Detection in Plants Using Image Segmentation," *Information Processing in Agriculture*,

vol. 7, no. 4, pp. 566-574, 2020.

- [17] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," in *Proc. European Conf. Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 122-138.
- [18] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 11966-11976.
- [19] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.
- [20] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Seoul, Republic of Korea, 2019, pp. 1314-1324.
- [21] R. Liu, "Higher Accuracy on Vision Models with EfficientNet-Lite," TensorFlow Blog, Mar. 16, 2020. [Online]. Available: <https://blog.tensorflow.org/2020/03/higher-accuracy-on-vision-models-with-efficientnet-lite.html>.
- [22] P. Yu, F. Teng, W. Zhu, C. Shen, Z. Chen, and J. Song, "Cloud-Edge-Device Collaborative Computing in Smart Agriculture: Architectures, Applications, and Future Perspectives," *Frontiers in Plant Science*, vol. 16, p. 1668545, 2025.