International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Based Pest Detection and Classification for Multi-Crop Farms Using Enhanced YOLOv8: A Review and Proposed Framework

Mrs. Minakshi A. Gite[1], Vaishnavi Madhukar Dhondge[2], Mayuri Manoj Ahire[3], Sakshi Dipak Daund[4], Mayuri Santosh Dere[5]

[1]Project Guide, Department of Information Technology, Matoshri College of Engineering and Research Centre, Eklahare, Nashik, Maharashtra, India

[2,3,4,5]Students, Department of Information Technology, Matoshri College of Engineering and Research Centre, Eklahare, Nashik, Maharashtra, India

Corresponding author e-mail: sakshidaund53@gmail.com

Abstract— Pests and plant diseases reduce crop yields every year, and smallholder farmers who grow several crops in one field are affected the most. Most existing computer-vision tools serve a single crop, depend on stable internet access, and report only a class label without any explanation or advice. This paper reviews the literature on multi-crop pest and disease detection, with particular attention to YOLO-based detectors, public datasets, multimodal weather and satellite data, explainable artificial intelligence, and deployment on low-cost devices. The review identifies five recurring gaps: single-crop design, missing severity estimation, image-only inputs, unexplained predictions, and weak support for offline use. In response, the paper proposes a layered framework that combines a multi-crop classifier, an enhanced YOLOv8 pest detector, a disease recognition model, pest counting with severity grading, a weather and vegetation-index risk model, a weighted multimodal risk score, Grad-CAM and SHAP explanations, a retrieval-augmented chatbot, and an early warning alert. An evaluation protocol based on precision, recall, mean average precision, F1-score, and inference speed is also defined. The framework is a design proposal, and experimental validation is planned as future work.

Keywords: Pest detection, YOLOv8, multi-crop farming, explainable AI, NDVI, precision agriculture.

I. INTRODUCTION

Agriculture supplies the food, fibre, and income on which most of the world's population depends, yet every season a large part of the harvest is lost to pests and plant diseases. Pests and pathogens remain among the most important causes of preventable yield loss in major food crops [1]. A recent review of crop disease detection reports that crop diseases alone cost the world roughly USD 220 billion each year, and that more than 800 million people rely on food-security crops such as cassava, maize, beans, rice, and potato for nutrition and livelihood [2]. These figures show why early and reliable detection is not only a technical problem but also an economic and social one.

The difficulty is greatest for smallholder farmers. Many of them grow two or three crops in the same field, so a single plot can host several pests and diseases at the same time. Detection still depends mostly on visual inspection: a farmer or an extension worker walks through the field and examines leaves one by one. This approach is slow, depends on personal experience, and often identifies a problem only after the damage has spread. The same review points out that traditional scouting is time-consuming, error-prone, and limited by the availability of experts [2].

Deep learning has changed what is possible. Early studies showed that convolutional neural networks (CNNs) can recognise diseases from leaf photographs with very high accuracy [3], [4], [5]. However, most of these results were obtained on carefully photographed images with plain backgrounds. Several authors have shown that accuracy often falls when the same models are applied to photographs taken in real fields, where lighting, background, leaf angle, and image quality vary widely [6]. In addition, a classification model only states what is in the image. It does not say where the pest is, how many pests are present, or how serious the attack is.

Object detectors address the first of these limitations. The *You Only Look Once* (YOLO) family predicts the position and class of every object in a single pass through the network, which makes it fast enough for practical use [7]. The family has grown through several generations [8], and its latest widely used releases, such as YOLOv8, are available in small and large variants [9]. These detectors are therefore good candidates for locating small insects on leaves, but pests are tiny and often camouflaged against the foliage, so a standard detector is likely to miss many of them.

A review of the field reveals five recurring gaps. First, most models are built for one crop, although intercropping is common [2]. Second, most studies name a pest or a disease but rarely estimate how severe the infestation is. Third, models use photographs alone and ignore weather and satellite data that can indicate stress before symptoms become visible. Fourth, predictions are rarely accompanied by an explanation, which makes farmers reluctant to trust them. Fifth, many tools need a stable internet connection and give no advice or early warning.

The review is guided by three questions. How well do current detection and classification methods work when crops are mixed and field conditions are uncontrolled? Which additional inputs and outputs, such as severity,

weather, vegetation indices, explanations, and advice, are needed to turn a detection result into a useful decision? Which design choices keep such a system affordable and deployable on smallholder farms?

This paper makes three contributions. (i) It reviews the literature on multi-crop pest and disease detection from the viewpoint of detection models, datasets, multimodal inputs, explainability, and deployment. (ii) It maps each identified gap to a concrete module and proposes a layered framework around an enhanced YOLOv8 detector. (iii) It defines an evaluation protocol that can be used to test the framework in future work. The paper does not report new experimental results; it is a review with a design proposal.

The remainder of the paper is organised as follows. Section II reviews related work. Section III describes the review approach, the proposed framework, and the evaluation protocol. Section IV presents the findings of the review. Section V discusses their implications and limitations, and Section VI concludes the paper.

II. LITERATURE REVIEW

A. Deep Learning for Plant Disease and Pest Recognition

The use of deep learning for plant health began with image classification. Sladojevic et al. trained a CNN to recognise several plant diseases directly from leaf images, without hand-crafted features [3]. Mohanty et al. trained deep networks on the PlantVillage collection of roughly 54,000 images covering 14 crops and 26 diseases, and showed that a computer can recognise a disease from a photograph with very high accuracy [4]. Ferentinos compared several CNN architectures on a large open image collection and reached the same conclusion [5]. Too et al. studied the fine-tuning of pre-trained networks for plant disease identification, which reduces training time and works well when the available data are limited [10].

Mobile deployment has also been explored. Ramcharan et al. built a cassava disease model that runs on an ordinary smartphone, which brought diagnosis from the laboratory into the field [11]. Broader surveys summarise how CNNs have been applied in agriculture [12] and how detection, classification, and segmentation methods have been used for plant diseases and pests [13]. Fuentes et al. applied detection networks to tomato plants and showed that diseases and pests can be recognised in real time, which was an early step from classification toward localisation [14].

B. The Gap Between Laboratory and Field

Barbedo analysed the main challenges of recognising plant diseases from visible-range images and identified background clutter, variable illumination, symptom variability, and the presence of several disorders on one leaf as persistent obstacles [15]. A later study examined the factors that influence deep learning for plant disease recognition and stressed that the composition of the training data strongly affects how well a model works on new images [6]. The review by Nalwanga et al. adds that models trained in one region may fail in another because soil, climate, crop varieties, and farming practices differ [2]. Together, these studies explain why high accuracy on a benchmark does not guarantee reliable behaviour on a farm.

C. YOLO-Based Detection

The original YOLO detector reframed object detection as a single regression problem and achieved real-time speed [7]. A comprehensive review traces the evolution of the family from YOLOv1 to YOLOv8 and beyond, describing the changes in backbones, necks, and prediction heads [8]. YOLOv8 uses an anchor-free prediction scheme, a decoupled head that separates classification from box regression, and a cross-stage module that improves gradient flow; it is released in five sizes, from the nano variant to the extra-large variant [9]. The small variants are of particular interest for agriculture because they can run on modest hardware.

Several studies have adapted YOLOv8 for pests. Wei et al. proposed AEC-YOLOv8n, which improves the small YOLOv8n model for the large IP102 pest dataset by adding two feature-enhancement modules and a weighted way of combining features from different levels, and they report a higher mean average precision than the original model [16]. Zhang et al. proposed YOLOv8-Extend for small pests that resemble the crop background. They tested it on an edge device with TensorRT acceleration and reported a speed of about 57.6 frames per second together with a gain of 4.6% in mean average precision over the original YOLOv8 [17]. Both studies concentrate on detection accuracy or speed. They do not address crop identification, severity estimation, weather data, or advice for farmers.

During training, YOLOv8 optimises a combined objective with terms for box regression, classification, and distribution focal loss [9]. The practical consequence is that the quality of the boxes and the quality of the class labels both influence the final score, which is why annotation quality, discussed in Section II-D, is so important for pest detection.

Three design ideas recur in the improvement of detectors for small objects. The first is attention, for example the convolutional block attention module (CBAM), which re-weights features along the channel and spatial dimensions so that the network concentrates on informative regions [18]. The second is multi-scale feature fusion, in which shallow and deep features are combined through a feature pyramid [19]. The third is weighted fusion, in which each input scale receives a learnable importance, as in the bidirectional feature pyramid network of EfficientDet [20]. These ideas motivate the enhancements proposed in Section III.

D. Public Datasets

Data are the foundation of every model. PlantVillage provides a large collection of leaf images taken under controlled conditions [21]. PlantDoc consists of images collected under less controlled conditions and was created

to support plant disease detection in more realistic settings [22]. CCMT contains images of cashew, cassava, maize, and tomato collected on farms in Ghana, and it is one of the few multi-crop collections [23]. IP102 is a large benchmark for insect pest recognition with 102 pest categories [24]. Table I summarises these collections, and Fig. 1 compares their sizes.

TABLE I. OPEN DATASETS CONSIDERED IN THIS WORK (COUNTS AS LISTED IN [2])

Dataset	Content	Images
PlantVillage [21]	14 crops; leaf diseases	54,306
PlantDoc [22]	13 crops; field-style leaf images	2,569
CCMT [23]	Cashew, cassava, maize, tomato	24,881
IP102 [24]	102 insect pest classes	75,222

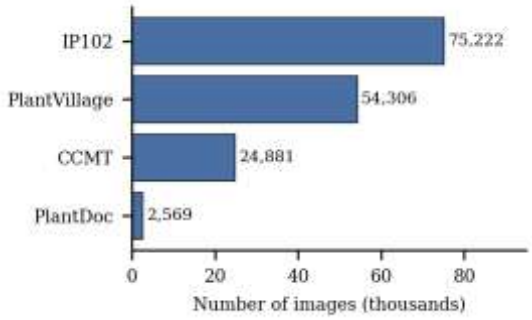


Fig. 1. Number of images in the open datasets considered.

Two observations follow from these collections. Datasets of leaf diseases are far more common than datasets of pests, and datasets that cover many crops at once are rare [2]. A model intended for mixed-crop farms therefore has to combine several sources, and this raises practical questions of label consistency and class balance.

Detection also requires bounding-box labels, which are much more expensive to produce than image-level labels. A pest dataset may therefore offer many images for classification but a smaller number with boxes. Annotation quality matters as well, because loose boxes, missed small pests, and inconsistent class names all reduce the reliability of reported metrics. For a combined multi-crop collection, a common label scheme, a consistent box format, and a record of the source of every image are needed so that results can be traced and the subsets can be separated without leakage.

E. Multimodal Data, Weather, and Vegetation Indices

A photograph shows only what is visible at the moment it is taken. Nalwanga et al. compare several data types, including ordinary RGB, spectral, thermal, and fluorescence images, and conclude that RGB images are cheap and widely available but can detect only visible symptoms, whereas spectral and fluorescence data can reveal stress earlier but require costly sensors [2]. The same review notes that weather data can complement images by adding context, although collecting them reliably requires sensors or data services, and that multimodal fusion can be performed at the data, feature, or decision level [2].

Satellite imagery offers a low-cost source of plant-health information. The normalised difference vegetation index (NDVI) combines the red and near-infrared reflectance of a surface and was introduced for monitoring vegetation from satellite data [25]. Healthy vegetation reflects strongly in the near-infrared band, so a falling NDVI value can signal weak or stressed plants. Weather variables such as temperature, humidity, and rainfall are linked to the spread of many fungal diseases [2]. Tabular inputs of this kind are usually handled well by ensemble learners such as Random Forest [26] and gradient-boosted trees such as XGBoost [27], which is why they are natural choices for a weather and NDVI risk model.

F. Explainable Artificial Intelligence

Farmers and extension officers are more likely to act on a result when they can see the reason for it. Grad-CAM produces a heat map that highlights the image regions that influenced the decision of a CNN [28]. It helps to check whether a model looked at the pest or the diseased tissue instead of the background, although it explains a prediction without proving that the prediction is correct. SHAP assigns each input feature a contribution value based on a game-theoretic idea, and it is well suited to tabular inputs, where it can show how much a factor such as humidity pushed a prediction up or down [29]. The two methods are complementary: one is designed for images and the other for tabular data.

G. Edge Deployment and Conversational Support

Cloud-based inference requires a stable connection, which is often unavailable in rural areas. Lightweight architectures such as MobileNets [30] and EfficientNet [31] reduce computation, and TinyML techniques allow models to run on very low-power devices [32]. The review by Nalwanga et al. lists smartphones, single-board computers such as the Raspberry Pi, and embedded accelerators as practical deployment targets, together with their memory, power, and cost [2]. The same review identifies the combination of vision models with large language models as an emerging direction for multilingual, conversational support, and it notes that model size and offline use remain open problems [2]. Retrieval-augmented generation (RAG) is a promising way to keep such answers grounded, because the language model draws on a store of curated documents instead of relying only on its internal knowledge [33].

Hybrid architectures that combine cloud, edge, and mobile layers are a promising compromise, because they allow real-time diagnosis on the device while model updates and retraining take place in the cloud [2]. Such a design suits the framework proposed here, in which image analysis can run locally but weather, vegetation, and conversational services depend on connectivity.

H. Pest Counting and Severity Assessment

Detecting a pest is only the first step of crop protection, because the decision to act depends on how many pests are present and how much damage they are likely to cause. Integrated pest management traditionally relies on action thresholds, in which a control measure is recommended only when the pest population or the damage reaches a level that justifies the cost of treatment. Object detectors fit this idea well, because every detected box can be counted. Counting nevertheless has difficulties: pests may overlap, hide beneath leaves, or appear at very different sizes, so the count from a single photograph can be an underestimate. In the detection studies discussed above, evaluation focuses on accuracy and speed rather than on counts or severity grades [16], [17]. A practical system therefore needs a simple and transparent rule that turns counts into levels, and it should state clearly that the levels are indicative and depend on the crop and the growth stage.

I. Summary of Representative Studies

Table II summarises the studies that shaped this work and the gap that each leaves open.

TABLE II. REPRESENTATIVE STUDIES AND THE GAPS THEY LEAVE

Study	Focus	Remaining gap
Nalwanga et al. [2]	Review of multi-crop and edge methods	Few field tests; limited datasets
Mohanty et al. [4]	CNNs on PlantVillage images	Controlled images; weak field transfer
Ramcharan et al. [11]	Smartphone cassava model	Single crop; uncontrolled conditions
Wei et al. [16]	AEC-YOLOv8n on IP102	Detection only; no severity or advice
Zhang et al. [17]	YOLOv8-Extend for small pests	Detection only; no weather data
Domingues et al. [34]	Survey on detection and prediction	No edge deployment
Mensah et al. [23]	CCMT multi-crop dataset	One region; four crops

III. METHODOLOGY

A. Review Approach

This work is a narrative review. The starting point was the multi-crop review by Nalwanga et al. [2], whose reference list was used to identify relevant work on datasets, deployment, and multimodal sensing. The set was then extended with studies on YOLO-based pest detection, explainable AI, tabular risk models, and retrieval-augmented generation. Peer-reviewed journal and conference papers, together with widely used datasets and software, were preferred. Studies were retained when they dealt with at least one of the following topics: multi-crop recognition, pest or disease detection, real-time or edge-capable models, multimodal inputs, explainability, or farmer-facing advice. The retained studies were compared on method, contribution, and remaining limitation, as shown in Table II.

B. Layered Framework

The proposed framework follows the steps that a farm expert would take: identify the crop, look for the pest and the disease, judge how serious the problem is, check the weather and the health of the plants, and finally give advice. It is organised in three layers, shown in Fig. 2. The input layer collects a leaf or pest photograph, local weather data, and the red and near-infrared bands of satellite imagery. The AI processing layer analyses these inputs. The decision and advisory layer converts the model outputs into a risk level, an explanation, advice, and an alert.

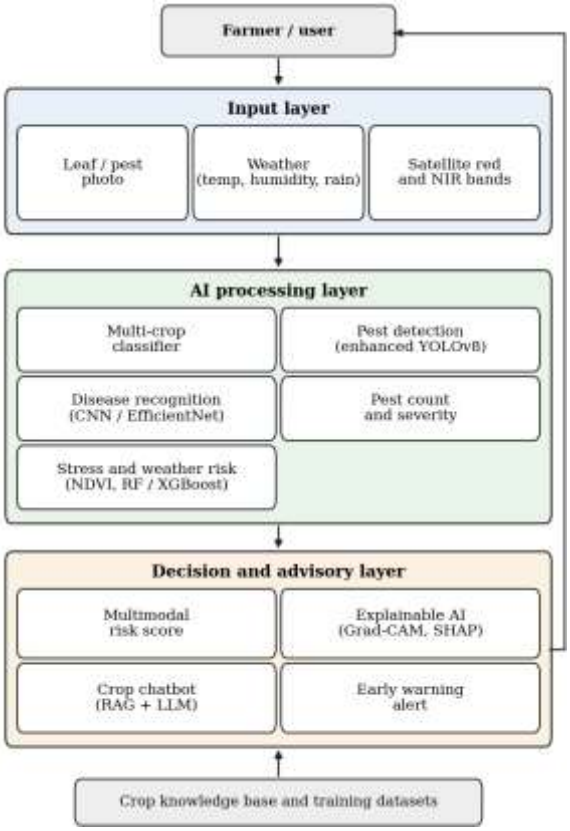


Fig. 2. Layered architecture of the proposed framework.

The photograph is resized, normalised, and augmented before it reaches the models. The weather values are collected for the farm location. The satellite bands are used to compute the vegetation index described in Section III-E. A crop knowledge base and the training datasets support the processing layer. The result, the risk level, and the alert are returned to the farmer.

C. Multi-Crop Classification and Enhanced YOLOv8 Detection

The first processing step is a multi-crop classifier that identifies which crop appears in the photograph. This step is intended to ensure that the pest and disease checks that follow are appropriate for that crop, which addresses the single-crop limitation discussed in Section I.

Pest detection uses YOLOv8 as the baseline [9]. The detector predicts a bounding box, a pest class, and a confidence score for every pest in the image. Three enhancements are planned to improve the detection of tiny insects against busy leaf backgrounds. A CBAM attention module [18] would help the network focus on small pests.

An additional prediction layer operating on high-resolution feature maps would give small objects a better chance of being detected. Weighted feature fusion, in the spirit of [16], [20], would allow shallow and deep features to contribute with learned importance. The compact variants YOLOv8n and YOLOv8s would be preferred so that the model remains suitable for phones and edge boards [17]. Fig. 3 shows the detection flow.

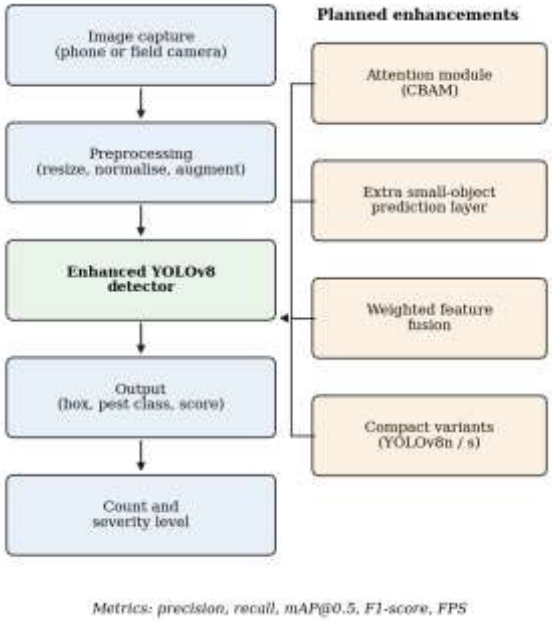


Fig. 3. Enhanced YOLOv8 detection flow and planned enhancements.

The quality of a predicted box is measured by its intersection over union (IoU) with the annotated box:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \qquad (1)$$

where A is the predicted box and B is the ground-truth box. A prediction is counted as correct when its IoU exceeds a chosen threshold, for example 0.5.

D. Disease Recognition, Pest Counting, and Severity

A separate CNN or EfficientNet model with transfer learning [31], [10] reads the leaf image and names the disease. This module works alongside the pest detector so that a photograph can report both pests and symptoms. The number of pests detected in each image is then counted and compared with set limits to grade the attack as low, medium, or high. The limits are not fixed in this paper because they depend on the crop and on the field data; they will be tuned during implementation. Counting converts the detector's output from a label into a measure of how serious the problem is, which most existing studies do not provide.

E. Weather and NDVI Risk Model

The vegetation index is computed from the red and near-infrared reflectance of the satellite bands [25]:

$$NDVI = \frac{\rho_{NIR} - \rho_{Red}}{\rho_{NIR} + \rho_{Red}} \qquad (2)$$

where ρ_{NIR} and ρ_{Red} denote the reflectance in the near-infrared and red bands. Values near zero or below indicate bare soil, water, or very weak vegetation, and higher values indicate denser, healthier plants. Freely available multispectral satellites such as Sentinel-2 provide these bands at a spatial resolution of 10 m, which is adequate for many farms but may be coarse for very small plots.

A Random Forest [26] or XGBoost [27] model would be trained on temperature, humidity, rainfall, and NDVI values to estimate crop stress and disease risk. The aim is to provide a warning even before spots appear on the leaves, which is not possible from photographs alone.

F. Multimodal Risk Score and Early Warning

The image result, the weather risk, and the NDVI stress are combined into a single score:

$$S = w_1 \cdot I + w_2 \cdot W + w_3 \cdot V \qquad (3)$$

where I, W, and V are the image-based, weather-based, and vegetation-based components, each scaled to the range 0 to 1, and w_1, w_2, and w_3 are their weights. The weights are to be learned with logistic regression on labelled field outcomes and then normalised. The score is mapped to three levels, as shown in Table III and Fig. 4. When the score crosses the high-risk limit, the system sends an early warning alert so that the farmer can act before the damage spreads. The numerical limits will be tuned on real data during implementation.

TABLE III. RISK LEVELS AND RECOMMENDED ACTIONS

Level	Meaning	Action
Low	Normal conditions	Keep routine monitoring
Medium	Rising risk	Inspect the field closely
High	Severe risk	Send alert to the farmer

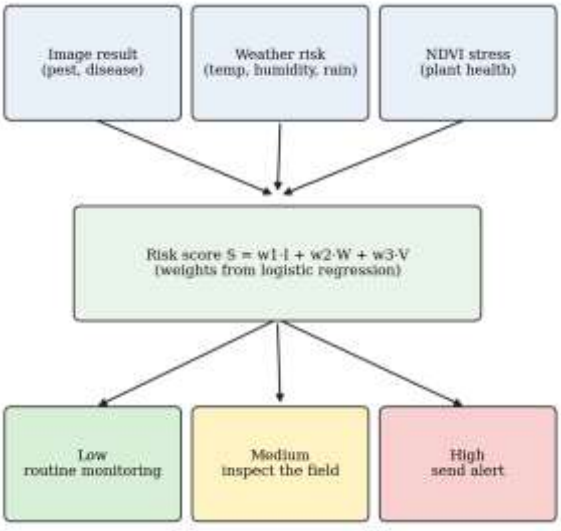


Fig. 4. Multimodal risk score and early warning levels.

G. Explainability and Chatbot Advice

Grad-CAM would be applied to the image models so that a farmer can see which part of the leaf influenced the result [28]. SHAP would be applied to the weather and NDVI model so that the contribution of each input to the risk score can be shown [29]. A chatbot based on retrieval-augmented generation [33] would answer questions about treatment and care in plain language. It would retrieve passages from a curated store of crop-care notes and combine them with the detected pest, disease, and risk

level before generating an answer. Grounding the answers in curated notes is intended to reduce the chance that the language model gives unsupported advice.

H. Evaluation Protocol

The detection model will be evaluated on a held-out test set that is not used for training or tuning. Precision, recall, and the F1-score are defined from the numbers of true positives (TP), false positives (FP), and false negatives (FN):

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

Detection quality across classes is summarised by the mean average precision (mAP), computed at an IoU threshold of 0.5 and denoted mAP@0.5. For each class, the average precision (AP) is the area under the precision–recall curve, and the mean is taken over the N classes:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (7)$$

Inference speed will be measured in frames per second on a laptop, on a cloud GPU, and, where possible, on an edge board. The enhanced model will be compared with an unmodified YOLOv8 baseline of the same size so that the effect of each enhancement can be isolated through ablation. The image models will also be tested on field-style images, for example from PlantDoc [22] and from farm photographs if they can be collected, to measure the laboratory-to-field gap discussed in Section II-B.

I. Data Preparation and Training Plan

Images from the shortlisted datasets would be merged into one multi-crop collection after the labels are harmonised into a single scheme. Duplicate and corrupted files would be removed. The collection would then be divided into training, validation, and test subsets. Images that come from the same plant, the same photographing session, or the same source would be kept in the same subset, because placing near-identical images in both the training and the test subsets inflates the measured accuracy. Augmentation such as flipping, rotation, scaling, and changes in brightness and contrast would be applied to the training subset only, so that the model becomes less sensitive to the lighting and orientation changes found in real fields [2].

The models would start from weights pre-trained on a large general image collection and would then be fine-tuned on the pest and disease data, a strategy that speeds up convergence and works well when data are limited [10]. Hyperparameters would be chosen using the validation subset only, and the test subset would be used once for the final report. The number of images per class would be examined, and class weighting or targeted augmentation would be considered if some pests turn out to be rare. The weather and NDVI records would be aligned with the date and location of each labelled observation so that the risk model learns from matching inputs.

J. Deployment and User Interaction

A simple interface built with Streamlit or Flask would let a farmer upload a photograph and see the crop name, the boxed pests, the disease, the severity level, the risk level, the explanation, and the chatbot on one screen. Training would be carried out on a laptop or on a cloud GPU such as Google Colab. After training, a compact variant would be exported and tested on a smartphone and, if possible, on a single-board computer such as the Raspberry Pi or an embedded GPU board such as the Jetson, which the base review lists among practical edge platforms [2]. Because the weather and satellite services need internet access, the complete framework is expected to run in a hybrid mode: image analysis could run locally, whereas weather, NDVI, and the chatbot would need a connection or cached data. A fully offline application is therefore left as future work.

K. Flow of Interaction

The interaction between the user and the system follows a short sequence. The farmer uploads an image through the interface, and the image is passed to the detection modules, which return the crop, the pests, and the disease. The risk engine then requests the weather and NDVI values, computes the risk score, and returns it together with the explanation. The interface shows the result and, when the risk is high, the alert. If the farmer asks a question, the query and the detected context are sent to the chatbot, which returns advice in plain language. Three kinds of actors are expected: farmers, who upload images and receive results; agricultural experts or administrators, who manage datasets, update models, and review performance and the crop-care notes; and external weather and satellite services, which supply the environmental data.

IV. RESULTS

This section presents the findings of the review. No experimental performance figures are reported, because the framework has not yet been trained and tested.

A. Findings from the Literature

The review shows that image classification of plant diseases is mature, but that practical diagnosis on mixed-crop farms is not yet solved. Four findings stand out. First, strong models exist for individual crops and for benchmark collections, but their accuracy on field images is less certain [4], [6]. Second, YOLOv8-based pest detectors have improved accuracy and speed for small pests, but the reported work is limited to detection [16], [17]. Third, multimodal data such as weather and satellite-derived indices are recognised as useful, yet they are rarely combined with photographs in a single farmer-facing system [2]. Fourth, explanations, severity grading, and conversational advice are usually absent from detection studies, even though they determine whether a result can be used.

B. Mapping Gaps to Proposed Modules

Table IV connects each gap identified in the literature to the module of the proposed framework that is intended to address it.

TABLE IV. IDENTIFIED GAPS AND PROPOSED RESPONSES

Gap in the literature	Proposed response
Models serve a single crop	Multi-crop classifier selects the correct checks
Severity is rarely estimated	Pest counting with low, medium, and high levels
Image-only inputs	Weather and NDVI risk model
Unexplained predictions	Grad-CAM for images; SHAP for tabular inputs
Cloud dependence; no advice	Compact YOLOv8 variants; RAG chatbot; alert

C. Comparison of Sensing Modalities

The choice of inputs strongly influences what a system can detect and where it can run. Table V summarises the main sensing modalities discussed in the review by Nalwanga et al. [2] from three viewpoints: acquisition cost, the ability to reveal stress before symptoms are visible, and suitability for edge devices.

TABLE V. COMPARISON OF SENSING MODALITIES, SUMMARISED FROM [2]

Modality	Cost	Pre-symptom	Edge fit
RGB images	Low	No	Excellent
Spectral	Very high	Yes	Poor to moderate
Thermal	Medium	Partly	Good
Fluorescence	High	Yes	Fair
LiDAR	Very high	No	Poor
IoT sensors	Variable	Partly	Excellent

The table shows a clear tension. The modalities that detect stress before symptoms appear are the most expensive and the hardest to deploy on small devices, while the cheapest and most deployable modalities see only visible symptoms. This tension explains the design choice of the proposed framework, which combines ordinary photographs with inexpensive environmental information instead of relying on specialised sensors.

D. Comparison with Existing Farmer-Facing Tools

Mobile applications such as Plantix, Nuru, and Agrio already bring diagnosis to smartphones, and they offer convenience and, in some cases, community knowledge sharing. The review by Nalwanga et al. notes, however, that such applications generally need internet connectivity, have restricted coverage of crops and regions, and may lose accuracy under diverse environmental conditions [2]. The proposed framework is intended to complement them by naming the crop first, by grading severity, by adding environmental risk, and by explaining its answers.

E. Status of the Proposed Work

At the time of writing, the literature survey and the identification of gaps are complete, and the architecture with its modules has been designed. Candidate datasets have been shortlisted, and the next steps are to clean and merge them, train a baseline YOLOv8 model, add the planned enhancements, and compare the resulting mean average precision with the baseline. These steps are listed here to make clear that the framework is a proposal and that its performance remains to be demonstrated.

V. DISCUSSION

A. Interpretation

The main message of the review is that accuracy on a single task is not enough. A farmer needs to know the crop, the pest, the severity, the reason for the result, and the next step, and the literature rarely provides all of these in one system. The proposed framework follows this chain of needs. Each module is based on an established technique, and the contribution lies in their integration for mixed-crop farms and in the emphasis on severity, explanation, and early warning.

The design also reflects a trade-off that appears throughout the literature. Advanced sensors such as hyperspectral or thermal cameras detect stress earlier, but they are expensive and complex, whereas phone cameras and open satellite data are accessible but less sensitive [2]. By using photographs, weather data, and freely available satellite bands, the framework favours accessibility and accepts that very early, asymptomatic detection will be limited.

Another implication concerns the way detection research is reported. A paper that reports only mean average precision on a benchmark tells a farmer little about whether the tool will help in the field. Reporting severity accuracy, behaviour on field-style images, speed on the target devices, and the usefulness of the explanations would give a fuller picture of real-world value, and the evaluation protocol in Section III-H has been written with this in mind.

B. Implications

For farmers, a single tool that reports a risk level and a plain-language next step is more useful than a raw class name. For extension officers, the explanations offer a way to check the model's reasoning before advising a farmer. For researchers, the framework suggests an ablation-based way to test whether attention, an extra small-object layer, and weighted fusion each improve detection on tiny pests.

C. Limitations and Risks

Several limitations should be kept in mind. The first concerns data. Public collections differ in image quality, labelling style, and geographic origin, and a model trained on them may not transfer to a new region [2]. Merging datasets requires careful label harmonisation, and class imbalance may bias the model toward common pests.

The second concerns the satellite and weather inputs. A satellite pixel of 10 m may mix several crops or bare soil on a small plot, which weakens the NDVI signal, and cloud cover can interrupt observations [2]. The weights of the risk score depend on labelled outcomes, which are costly to collect and may be hard to obtain at first.

The third concerns the chatbot. A language model can produce confident but incorrect statements, and retrieval reduces but does not remove this risk. Advice on pesticides or treatment should therefore be reviewed by agricultural

experts before it enters the knowledge base. Language barriers are also important, since many farmers prefer local languages [2].

The fourth concerns privacy and scope. Farm images may reveal sensitive information about land and neighbouring plots, so data handling policies are needed [2]. The framework works on photographs and open data, and laboratory tests are outside its scope.

A fifth consideration is the cost of different errors. In pest management, a missed infestation is usually more harmful than a false alarm, because a missed pest can spread unnoticed while a false alarm costs only an unnecessary inspection. The review therefore supports tuning the detector and the alert limit toward high recall, while monitoring precision so that frequent false alarms do not make farmers ignore the warnings [2]. Reporting both measures, rather than accuracy alone, is essential for this reason.

Finally, reproducibility deserves attention. Dataset sources, label schemes, subset assignments, training settings, and software versions should be recorded so that other researchers can repeat the experiments. Using an open implementation such as Ultralytics YOLO [9] and public datasets makes this easier, provided that the exact versions are stated.

D. Future Work

The immediate next step is implementation and evaluation according to the protocol in Section III-H. After that, a fully offline mobile application based on compact models would address connectivity limits, and quantisation or pruning could reduce the model size further [32]. Other extensions include more crops, local-language support, drone imagery, and spectral or thermal sensing. Field trials with farmers are essential, because laboratory-style testing cannot show whether the tool is understandable and useful in practice.

VI. CONCLUSION

This paper reviewed the literature on multi-crop pest and disease detection and identified five gaps: single-crop design, missing severity estimation, image-only inputs, unexplained predictions, and limited support for offline use and advice. In response, it proposed a layered framework that combines a multi-crop classifier, an enhanced YOLOv8 pest detector, disease recognition, pest counting with severity grading, a weather and NDVI risk model, a weighted multimodal risk score, Grad-CAM and SHAP explanations, a retrieval-augmented chatbot, and an early warning alert. The paper also defined an evaluation protocol based on precision, recall, mAP@0.5, F1-score, and speed.

The framework is a design proposal, and its performance has not yet been measured. The next stage of the project is to train and test the models, tune the risk limits on real data, and compare the enhanced detector with a baseline. If the framework performs as intended, it could give small and mixed-crop farms one practical tool that finds pests, rates their seriousness, explains its answers, and warns early.

ACKNOWLEDGMENT

The authors thank the Department of Information Technology, Matoshri College of Engineering and Research Centre, Eklahare, Nashik, for providing the facilities and support for this work.

REFERENCES

[1] S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, “The global burden of pathogens and pests on major food crops,” Nature Ecology & Evolution, vol. 3, no. 3, pp. 430–439, 2019.

[2] R. Nalwanga, L. Spreeuwers, E. Talavera, and G. Owomugisha, “Multi-crop disease detection in computer vision for resource-constrained farms—A review,” IEEE Access, vol. 13, pp. 217265–217284, 2025.

[3] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep neural networks based recognition of plant diseases by leaf image classification,” Computational Intelligence and Neuroscience, vol. 2016, Art. no. 3289801, 2016.

[4] 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.

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

[6] J. G. A. Barbedo, “Factors influencing the use of deep learning for plant disease recognition,” Biosystems Engineering, vol. 172, pp. 84–91, 2018.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

[8] J. Terven and D.-M. Córdova-Esparza, “A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” Machine Learning and Knowledge Extraction, vol. 5, no. 4, pp. 1680–1716, 2023.

[9] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLO,” version 8.0.0, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[10] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, “A comparative study of fine-tuning deep learning models for plant disease identification,” Computers and Electronics in Agriculture, vol. 161, pp. 272–279, 2019.

[11] A. Ramcharan et al., “A mobile-based deep learning model for cassava disease diagnosis,” Frontiers in Plant Science, vol. 10, p. 272, 2019.

[12] A. Kamilaris and F. X. Prenafeta-Boldú, “A review of the use of convolutional neural networks in agriculture,” Journal of Agricultural Science, vol. 156, no. 3, pp. 312–322, 2018.

[13] J. Liu and X. Wang, “Plant diseases and pests detection based on deep learning: A review,” Plant Methods, vol. 17, Art. no. 22, 2021.

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

[15] J. G. A. Barbedo, “A review on the main challenges in automatic plant disease identification based on visible range images,” Biosystems Engineering, vol. 144, pp. 52–60, 2016.

[16] J. Wei et al., “Improving the accuracy of agricultural pest identification: Application of AEC-YOLOv8n to large-scale pest datasets,” Agronomy, vol. 14, no. 8, p. 1640, 2024.

[17] R. Zhang, X. Bai, and J. Fan, “Crop pest target detection algorithm in complex scenes: YOLOv8-Extend,” Smart Agriculture, vol. 6, no. 2, pp. 49–61, 2024.

[18] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in Proc. European Conf. Computer Vision (ECCV), 2018, pp. 3–19.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936–944.

[20] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10781–10790.

[21] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv:1511.08060, 2015.

[22] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in Proc. 7th ACM IKDD CoDS and 25th COMAD, 2020, pp. 249–253.

[23] P. K. Mensah et al., "CCMT: Dataset for crop pest and disease detection," Data in Brief, vol. 49, Art. no. 109306, 2023.

[24] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, "IP102: A large-scale benchmark dataset for insect pest recognition," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8787–8796.

[25] J. W. Rouse, R. H. Haas, J. A. Schell, and D. W. Deering, "Monitoring vegetation systems in the Great Plains with ERTS," in Proc. 3rd Earth Resources Technology Satellite-1 Symp., NASA SP-351, vol. 1, 1974, pp. 309–317.

[26] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[27] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[28] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 618–626.

[29] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 4765–4774.

[30] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.

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

[32] N. N. Alajlan and D. M. Ibrahim, "TinyML: Enabling of inference deep learning models on ultra-low-power IoT edge devices for AI applications," Micromachines, vol. 13, no. 6, p. 851, 2022.

[33] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 9459–9474.

[34] T. Domingues, T. Brandão, and J. C. Ferreira, "Machine learning for detection and prediction of crop diseases and pests: A comprehensive survey," Agriculture, vol. 12, no. 9, p. 1350, 2022.