Journal of Cereal Research
Volume 18 (1): 1-24
Homepage: http://epubs.icar.org.in/ejournal/index.php/JWR

*Review Article*

# Edge - AI for Transforming Plant Disease Identification and Monitoring with Major Focus on Cereal Crops - An Overview

Tamanna Chauhan[1,2], Ankita Kailas Kurhade[1,2], Sivadinesh Chinnadurai[1,2], Ashok Beniwal[1,2], Chandra Nath Mishra[1], Satish Kumar[1*], Om Prakash Ahlawat[1] and Ratan Tiwari[1]

[1]*ICAR- Indian Institute of Wheat and Barley Research, Karnal-132001, Haryana*
[2]*ICAR- Indian Agricultural Research Institute, New Delhi-110012*

***Article history:***
Received: 27 Dec., 2025
Revised: 14 Feb., 2026
Accepted: 05 Mar., 2026

***Citation:***
Chauhan T, AK Kurhade, S Chinnadurai, A Beniwal, CN Mishra, S Kumar, OP Ahlawat and R Tiwari. 2026. Edge - AI for Transforming Plant Disease Identification and Monitoring with Major Focus On Cereal Crops - An Overview. *Journal of Cereal Research* ***18 (1):*** *1-24.* http://doi.org/10.25174/2582-2675/2026/174597

****Corresponding author:***
*E-mail: kumarsatish227@gmail.com*



## Abstract

**Plant diseases represent a persistent threat to global agricultural productivity, with annual crop losses of 20–40% threatening food security and farmer livelihoods worldwide. Particularly in the early stages of infection, when intervention is most effective, traditional visual inspection techniques continue to be subjective, laborious, and unreliable. Although automated, image-based disease categorization has been made possible by recent developments in artificial intelligence (AI), machine learning (ML), and deep learning (DL), the majority of solutions rely on cloud infrastructure, which is a significant drawback in rural agricultural areas with poor connectivity. A viable substitute is Edge AI, which applies AI inference directly to nearby devices (such as smartphones, drones, and microcontrollers) and provides real-time analysis, lower latency, improved data privacy and offline capabilities. Significant obstacles still exist, nevertheless, in the areas of computing effectiveness, model generalization, and real-world application across numerous agricultural contexts. The current level of Edge AI for plant disease detection is critically examined in this article, with a focus on cereal crops, which are essential to the world's food security. In order to facilitate deployment on devices with limited resources, we thoroughly examine the Edge AI frameworks (TensorFlow Lite, OpenVINO, ONNX Runtime, Edge Impulse), hardware architectures (Google Edge TPU, NVIDIA Jetson, Intel Movidius, Raspberry Pi), and optimization strategies (quantization, pruning, knowledge distillation). We find important gaps through a thorough analysis of applications specific to cereals, including a lack of standardized evaluation procedures, dataset homogeneity that jeopardizes model robustness, limited field validation under variable environmental conditions, and inadequate integration of temporal and environmental data. Although Edge AI shows technological viability, lightweight systems can achieve >90% accuracy on controlled datasets, it is still difficult to translate to diverse field circumstances. To achieve the potential of Edge AI in precision agriculture, we conclude that: (1) domain-adaptive models trained on a variety of multi-location field datasets must be developed; (2) sensor systems and evaluation benchmarks must be standardized; (3) multimodal sensing (spectral, temporal, and environmental) must be integrated to improve diagnostic reliability; (4) few-shot learning and transfer learning must be explored to reduce data requirements; and (5) emerging technologies, such as large language models (LLMs) for farmer-accessible interfaces and federated learning for privacy-preserving model improvement, must be investigated. The detection of Fusarium head blight in wheat, blast surveillance in rice, and rust disease monitoring systems are priority topics for cereal crops in particular. These diseases have a significant economic effect and intricate symptomologies that require advanced edge-based solutions.**

**Keywords:** Edge AI; plant disease detection; deep learning; precision agriculture; TensorFlow Lite; TinyML; IoT

## 1. Introduction

One of the biggest obstacles to worldwide agricultural output is still plant diseases, which compromise crop quality and quantity and jeopardize food security. The FAO claims that pests and plant diseases reduce worldwide crop yields by 20–40% annually, which has a negative impact on farmer livelihoods and economic stability (Savary *et al.,* 2019). Early and precise disease identification is crucial for minimizing significant losses and guaranteeing sustainable food supply in nations like India, where agriculture feeds over half of the population and accounts for about 18% of the country's GDP (Government of India, 2022). Conventional illness detection techniques, which mostly rely on visual inspection, are subjective, time-consuming, and frequently unreliable, particularly in the early stages of the disease when symptoms are mild or mistaken for abiotic stress (Bock *et al.,* 2010). These limitations highlight the need for automated, scalable, and precise diagnostic tools for modern agriculture.

By enabling automated image-based categorization with high accuracy, recent advances in artificial intelligence (AI), machine learning (ML), and deep learning (DL) have transformed the diagnoses of plant diseases (Mohanty *et al.,* 2016; Ferentinos *et al.,* 2018). But most of these AI systems depend on cloud computing, which necessitates reliable internet access, which is a significant problem in rural farming areas of Asia and Africa (Kamilaris *et al.,* 2018). Edge AI, which enables AI models to operate directly on local devices like smartphones, drones, microcontrollers, and field-deployed sensors, has emerged as a revolutionary approach to get around this restriction. Edge AI guarantees low latency, real-time inference, decreased bandwidth consumption, improved data privacy, and robustness in low-connectivity conditions by processing data locally rather than sending it to distant servers (Chen and Ran 2019).

Furthermore, it is now possible to implement high-performance plant disease detection models on energy-efficient edge platforms thanks to developments in lightweight deep-learning architectures (such as MobileNet, EfficientNet and ShuffleNet), model compression, quantization, and specialized hardware accelerators like Google Edge TPU, NVIDIA Jetson, and Intel Movidius (Howard *et al.,* 2019; Wang *et al.,* 2025). These advancements offer a solid technical basis for developing field-ready, real-time, and farmer-friendly precision agriculture diagnostic systems.

While numerous surveys have explored AI, ML and DL for plant disease detection–emphasis on cloud-based models, dataset benchmarks, and accuracy improvements (Kamilaris and Prenafeta-Boldú, 2018; Ferentinos, 2018; Jackulin *et al.,* 2022), they mostly overlook Edge AI's deployment realities in resource-constrained, low-connectivity agricultural settings. A thorough analysis is necessary to summarize technological developments, assess existing frameworks and hardware, and suggest future research routes given the quick development of Edge AI and its expanding application to agriculture. Existing reviews concentrate on centralized architectures, model training on lab datasets such as Plant Village, and generalization issues while ignoring on-device optimization pipelines, hardware-software co-design (e.g., TensorFlow Lite with Jetson Nano), or real-time inference trade-offs such as quantization and pruning. This review fills these gaps by providing the comprehensive synthesis of Edge AI frameworks, hardware accelerators, lightweight model strategies, and field-ready pipelines designed for offline plant disease monitoring across diverse crops, enabling scalable precision agriculture in rural India and beyond.

## 2. Background

In recent years, technological advancements have revolutionized various aspects of agriculture. Artificial Intelligence and machine learning have emerged as powerful tools in automated disease detection. Farmers can use mobile devices to capture images of diseased plants, and AI models can analyze and classify them within seconds. However a major limitation of AI based solutions is their reliance on cloud computing, which necessitates an internet connection. Many rural farmers lack stable connectivity, making cloud-based solutions impractical. (Nyakuri *et al.,* 2025)

Edge AI presents a viable solution to this challenge by allowing AI computations to be performed locally on edge devices, such as smartphones, drones and sensors (Chang *et al.,* 2021). It refers to the deployment of AI algorithm and models directly on local edge devices like sensors or IoT devices, enabling real time data processing and analysis without relying on cloud server. It combines edge computing with AI to perform different tasks directly, allowing on site decision making and nullifying the need to constantly transmit data to a central location. Early disease detection helps minimize pesticide use, reducing environmental impact and promoting eco-friendly farming methods (Green *et al.,* 2023).

It is central to the success of smart agriculture, a framework that involves deploying AI algorithms directly on edge devices within the agricultural environment.

## 3. Edge AI Frameworks

In the domain of Edge AI, software frameworks play a critical role in enabling the deployment of machine learning models directly on edge devices for real time interference. The two primary tools of TensorFlow Lite, an open-source deep learning framework created by Google for inference on embedded devices, are Converter and Interpreter. TensorFlow Interpreter is a library that runs the code on embedded devices, but it only supports a subset of TensorFlow operations and cannot be used for model training. TensorFlow Converter converts the TensorFlow code into a specific format, reduces the size of the model, and optimizes the code for minimal accuracy loss. Another key framework is Intel's OpenVINO Toolkit, which accelerates inference performance on Intel hardware, including CPUs, integrated GPUs and VPUs. OpenVINO converts pre-trained models into an Intermediate Representation (IR), optimizing them for low-latency execution on Intel edge platforms.

Edge Impulse offers a complete MLOps pipeline for TinyML, covering data acquisition, training, optimization, and deployment, particularly for micro controllers and ultra-low-power edge devices. Its architecture and application were formally documented by Banbury (2021) in the MLSys conference, highlighting its significance for embedded AI development. ONNX (Open Neural Network Exchange) and its runtime further extend the interoperability of AI models by allowing models trained in different frameworks (e.g., PyTorch. 2020; TensorFlow. 2020) to be deployed across various platforms. Developed jointly by Microsoft and Facebook in 2017, ONNX Runtime enables accelerated execution and has been shown to outperform native backends in certain edge deployments through memory optimization and operator fusion (Table 1).

Additionally, the growing TinyML ecosystem–comprising frameworks like TensorFlow Lite Micro, Tiny Engine, and uTVM–supports inference on microcontrollers with extremely limited computational resources. A comprehensive overview of TinyML for edge AI was presented by Soro (2021), while reformable TinyML approaches were surveyed by Rajapakse *et al.* (2023) emphasizing model compression, energy efficiency, and edge adaptability. These frameworks support real-time inference by leveraging model quantization, pruning, and hardware-aware compilation strategies. Collectively, these Edge AI software frameworks are pivotal in enabling low-power, high-speed, and offline AI processing, facilitating real-time applications in agriculture (Figure 1).

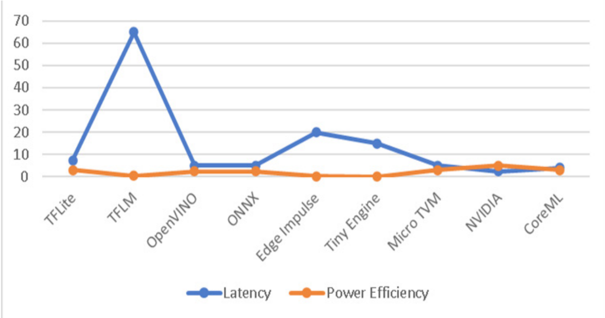


Fig 1: Comparative latency and Power Efficiency of Major AI frameworks

## 4. Edge Computing Hardware and Systems:

Hardware used in real time plant disease detection, especially in Edge AI systems deployed in agricultural fields (e.g., on drones, mobile phones, sensors). These are specific or general-purpose computing system that run the machine learning models to get the inference on device. Various edge AI devices are classified on the basis of their general architecture.

***4.1 Application specific Integrated Circuits (ASICs)*** could be preferred some specific cases in order to reduce high power consumption (Capra *et al.,* 2019). ASICs can satisfy the demands of edge computing patterns for AI algorithms due to their tiny size, low power consumption, and strong security performance. To reduce latency and energy consumption, the DianNao series of DNN accelerators makes use of effective memory access (Chen *et al.,* 2016). Google created the Edge Tensor Processing Unit (TPU) to run machine learning models at the edge. Edge TPUs perform well in terms of reduced power consumption and a smaller physical footprint (Chang *et al.,* 2021). The primary disadvantage is the loss of architectural flexibility, which is addressed by the use of semi-specific processors known as application specific integrated processors

**Table 1:** Comparative Efficiency of Major Edge AI Frameworks

| Framework | Efficiency (Latency) | Efficiency (Model Size Reduction) | Efficiency (Memory Footprint) | Power Efficiency | Throughput (FPS) | Strengths | References |
|---|---|---|---|---|---|---|---|
| TensorFlow Lite (TFLite) | 5–20 ms (MobileNet on ARM CPU) | 30–70% (via quantization) | 3–20 MB | Low (1–3 W on mobile) | 30–60 FPS (Edge TPU) | Highly optimized for mobile & Edge TPU | David *et al.*, 2021; Abadi *et al.*, 2016 |
| TensorFlow Lite Micro (TFLM) | 20–150 ms on MCUs | 50–80% | 20–250 kB RAM | Very low (<0.5 W) | 1–10 FPS | Best for microcontrollers, ultra-low memory | Banbury *et al.*, 2021 |
| OpenVINO Toolkit | 1–10 ms (INT8 on Intel CPU/VPU) | 40–70% | 10–50 MB | Low (1–5 W on VPU) | 30–120 FPS | Fastest CPU/VPU optimization engine | Li *et al.*, 2022 |
| ONNX Runtime | 5–15 ms (ARM/Intel CPUs) | 20–50% (operator fusion) | 5–25 MB | Low–Moderate (1–5 W | 20–60 FPS | Highly portable, multi-backend | Nocker *et al.*, 2025 |
| Edge Impulse Runtime | 10–50 ms (MCU/Cortex-M) | 50–85% | 10–150 kB RAM | Very low (<0.3 W) | 1–8 FPS | Best end-to-end TinyML pipeline | Hymel *et al.*, 2022 |
| TinyEngine / MCUNet | 10–40 ms | 60–90% | <100 kB Flash, <256 kB RAM | Ultra-low (<0.1 W) | 2–10 FPS | Highest efficiency for IoT-class MCUs | Lin *et al.*, 2023 |
| TVM / MicroTVM | 5–15 ms (ARM cores) | 30–60% | 2–10 MB | Low (1–3 W) | 20–40 FPS | Best compiler-level optimization & autotuning | Chen *et al.*, 2018 |
| NVIDIA TensorRT | 1–5 ms (FP16/INT8 on Jetson) | 50–70% | 20–200 MB | Moderate (5–15 W) | 60–200 FPS | Fastest GPU inference engine | Shafi *et al.*, 2021 |
| CoreML | 2–10 ms (A14 neural engine) | 30–60% | 5–20 MB | Low (<3 W) | 30–120 FPS | Extremely optimized for iOS | Sahin *et al.*, 2020 |
| TensorFlow Lite (TFLite) | 5–20 ms (MobileNet on ARM CPU) | 30–70% (via quantization) | 3–20 MB | Low (1–3W on mobile) | 30–60 FPS (Edge TPU) | Highly optimized for mobile & Edge TPU | David *et al.*, 2021; Abadi *et al.*, 2016 |
| TensorFlow Lite Micro (TFLM) | 20–150 ms on MCUs | 50–80% | 20–250 kB RAM | Very low (<0.5 W) | 1–10 FPS | Best for microcontrollers, ultra-low memory | Banbury *et al.*, 2021 |
| OpenVINO Toolkit | 1–10 ms (INT8 on Intel CPU/VPU) | 40–70% | 10–50 MB | Low (1–5W on VPU) | 30–120 FPS | Fastest CPU/VPU optimization engine | Li *et al.*, 2022 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ONNX Runtime | 5–15 ms (ARM/ Intel CPUs) | 20–50% (operator fusion) | 5–25 MB | Low–Moderate (1–5 W | 20–60 FPS | Highly portable, multi-backend | Nocker *et al.*, 2025 |
| Edge Impulse Runtime | 10–50 ms (MCU/ Cortex-M) | 50–85% | 10–150 kB RAM | Very low (<0.3 W) | 1–8 FPS | Best end-to-end TinyML pipeline | Hymel *et al.*, 2022 |
| TinyEngine / MCUNet | 10–40 ms | 60–90% | <100 kB Flash, <256 kB RAM | Ultra-low (<0.1 W) | 2–10 FPS | Highest efficiency for IoT-class MCUs | Lin *et al.*, 2023 |
| TVM / MicroTVM | 5–15 ms (ARM cores) | 30–60% | 2–10 MB | Low (1–3 W) | 20–40 FPS | Best compiler-level optimization & autotuning | Chen *et al.*, 2018 |
| NVIDIA TensorRT | 1–5 ms (FP16/ INT8 on Jetson) | 50–70% | 20–200 MB | Moderate (5–15 W) | 60–200 FPS | Fastest GPU inference engine | Shafi *et al.*, 2021 |
| CoreML | 2–10 ms (A14 neural engine) | 30–60% | 5–20 MB | Low (<3 W) | 30–120 FPS | Extremely optimized for iOS | Sahin *et al.*, 2020 |

(ASIPs), which are made to use specialized hardware to speed up recurring application-specific operations (like convolution or bit-wise operations). The overall processor is not as specifically tailored on the whole application as in ASICs in order to remain flexible and future-ready (Capra *et al.,* 2019). It supports CNN models like Mobile Net, Efficient Net Lite, YOLO-tiny.

***4.2 Central Processing Unit (CPU)*** can run light weight or moderately complex plant disease detection by image processing using CNN model to get the inference. Devices like Raspberry Pi (Darak *et al.,* 2025), Jetson Nano (Iftikhar *et al.,* 2024), smartphones uses CPU (Richey *et al.,* 2020).

***4.3 Graphic Processing Unit (GPU)*** is based on fundamental feature of data parallelism i.e., ideal for deep neural networks and increase the real throughput, reaching a faster speed than CPU. This characteristic make GPU more suitable for implementing AI algorithms(Chang *et al.,* 2021). Designing an edge device with a GPU is a superior choice because it is not cloud-based and can function without the internet. The NVIDIA Jetson Nano and Jetson Xavier NX are compact, low latency, and very power-efficient embedded AI computer devices.

***4.4 Field Programmable Gate Array (FPGAs)*** are very adaptable, low-power, parallel computing resources, and extremely secure (Chang *et al.,* 2021). For tasks like processing images, analysing sensor data or running AI models, it enables the creation of custom-built logic. These FPGAs are suitable for fixed functions such as image filter or CNN based disease diagnosis on leaves.

## 5. Workflow of Disease Detection in Plants

### *5.1 Image Acquisition*

The initial step in the image analysis process is image acquisition. Plant disease detection begins with high quality image capture using smartphones, UAVs or RGB sensors. Traditional computer vision pipelines for plant disease detection typically rely on high resolution image capture followed by cloud based processing It can be defined as the visual character of an object can be represented as a digitally encoded one. In simple terms, it can be defined as an image captured using a camera. Nowadays digital image is extended to a mobile phone it makes the process of image acquisition a user-friendly. Photographs, printed paper, and photographic film are the media used for it. However, edge constraints fundamentally alter this architecture by necessitating on-device preprocessing to reduce data transmission overhead and enable real-time inference in connectivity-limited agricultural environments (Ushadevi *et al.,* 2020). Edge deployed systems implement dynamic image resolution adjustments based on available computational resources and battery status unlike traditional pipelines that capture maximum resolution by default (Ahmed *et al.,* 2021).

Rather than transmitting full images to cloud servers, edge pipelines perform lightweight segmentation and Region-of-Interest extraction directly on smartphones or UAVs to isolate disease leaf regions, significantly reducing data volume and latency (Oliver *et al.,* 2018).

### *5.2 Image Preprocessing on Device*

There are two forms of image pre-processing: digital image processing and analog image processing. The primary procedure involved is the elimination of undesirable elements from the picture. Unwanted aspects in the image are eliminated using additional algorithms. 1. Image acquisition is one of the main phases in image pre-processing. 2. Normalization of Images 3. Improving the Image 4. Partitioning 5. Morphology. Picture segmentation is the process of dividing a picture into individual pixels and their comparable characteristics. It mostly aids in the process of interpreting images (Oliver *et al.,* 2018). It creates a high-level image from the low-level one. Reliability in the segmentation process is crucial to the success of image analysis. Both contextual and non-contextual factors are involved in the segmentation process. The segmentation procedure makes use of multiple algorithms. Z During the feature selection process, a duplicate of the original feature is retained. These two procedures primarily deal with eliminating undesired noise from the image and selecting the features that are required just for the image analysis process. New sets of features are generated in the feature extraction phase. During the feature extraction phase, the attribute procedure is transformed. This procedure improves the process's efficiency and speed.

The data are divided into several classes throughout the classification phase. Determining which class a new observation belongs to if it enters the process (Ferentinos *et al.,* 2018a). For the classification process, a number of classification methods are available that also produce accurate classification results.

### *5.3 Model Inference using Edge AI Frameworks*

Model inference at the edge relies on specialized Edge AI frameworks that optimize deep-learning models for low-power hardware while maintaining acceptable prediction accuracy. Frameworks such as TensorFlow Lite, PyTorch Mobile, and ONNX Runtime enable split inference, where early convolutional layers execute on -device for rapid feature extraction, while computationally intensive classification layers can optionally offload to edge servers when connectivity permits (Ahmed *et al.,* 2021; Singh *et al.,* 2023). This tiered approach contrasts sharply with traditional monolithic pipelines where the entire model runs in a single location. Optimized runtimes like NVIDIA TensorRT, OpenVINO, and Qualcomm SNPE further accelerate inference by exploiting GPU, VPU, and DSP architectures, achieving faster throughput and lower energy consumption—critical for field conditions where connectivity is limited (Qazi *et al.,* 2022; Shankar *et al.,* 2024). These frameworks ensure that plant disease detection models run efficiently on edge devices with reduced computational load, minimal memory footprint, and high inference speed, making them suitable for real-time monitoring in agriculture. Traditional pipelines use framework- agnostic operations while edge pipelines sacrifice portability for performance through hardware-specific optimization. Edge frameworks implement adaptive inference that switches between model variants (e.g., MobileNetV3-Small vs. MobileNetV3-Large) based on real-time battery level, thermal state, and latency requirements—a capability absent in traditional static pipelines. Therefore, inference shifts from a single pass, fixed architecture process to a dynamic, resource aware execution strategy with continuous monitoring and adaptation.

### *5.4. Model Optimization Techniques for Edge AI Inference*

Model optimization is essential for deploying deep-learning models on resource-constrained edge devices used in field-based plant disease detection. Techniques such as model quantization, including post-training quantization and quantization-aware training, reduce model size and accelerate inference by converting 32-bit floating-point weights into 8-bit, enabling 4x memory reduction and 2-4x inference speedup (Jacob *et al.,* 2018; Wan *et al.,* 2020). Structured pruning removes redundant neurons or filters, creating irregular sparse networks that necessitate custom sparse convolution kernels and modified backbones(Han *et al.,* 2015). Thereby traditional dense matrix operations atre replaced with sparse tensor operations, requiring new dataflow patterns and memory access strategies optimized for sparse data.Knowledge distillation further compresses models by training a smaller student model to mimic a larger, high-performing teacher network, achieving faster inference suitable for mobile and embedded platforms (Hinton *et al.,* 2015). Framework-driven optimizers such as TensorFlow Lite Converter, PyTorch Mobile's quantization toolkit, and ONNX Runtime apply graph level optimizations that merge sequential operations into single optimized kernels (Ahmed *et al.,* 2021). Then these computational graph structure itself is transformed, with traditional multilayer sequences collapsed into fused operations..

### *5.5. Decision Output*

Disease Identification and Severity Estimation Once the model completes inference on the edge device, the output is converted into actionable information for farmers or field technicians. Traditional pipelines output raw predictions (class labels, confidence scores) for post processing on powerful servers. Edge constraints require on- device decision logic and actionable output generation. Edge pipelines integrate. Edge pipelines integrate lightweight post- processing modules that compute disease severity (percentage leaf area affected, lesion count, severity grades) directly on device using efficient image analysis algorithms (Barbedo *et al.,* 2019; Ahamed *et al.,* 2024).). Unlike traditional pipelines that defer decision-making to external systems, edge pipelines embed rule-based or lightweight ML-based recommendation engines that map disease-severity pairs to actionable interventions (pesticide selection, treatment thresholds) using on-device lookup tables or compact decision trees (Mattihalli *et al.,* 2021; Darmawan *et al.,* 2024; Vijayakumar *et al.,* 2025). BeagleBone Black systems deploy pre-stored disease databases with automatic sprinkler actuation based on severity classification (initial vs. final stage), operating without cloud dependency (Mattihalli *et al.,* 2021). Raspberry Pi 4B deployments provide biological agent-based recommendations for identified threats without internet connectivity using compressed student models (0.12 GFLOPs, 38.9 ms inference) (Darmawan *et al.,* 2024). Edge pipelines implement adaptive output complexity: detailed reports with visualizations when resources permit, minimal text-based alerts under severe battery/memory constraints.

Edge constraints do not merely reduce model size—they fundamentally restructure the entire computer vision pipeline architecture, introducing new stages (adaptive preprocessing, runtime resource monitoring), reformulating existing operations (quantized/sparse computation), and enabling distributed execution strategies (tiered inference, dynamic model switching). These alterations represent a paradigm shift from traditional cloud-centric, resource-abundant pipelines to resource-aware, autonomous, and adaptive edge-native architectures.

## 6. Architecture in Edge AI

Architecture in Edge AI means the overall design and structure of how AI models, hardware, and software are organized ad work at the edge. Architecture consists of three layers namely hardware architecture, software architecture, and system architecture. Hardware and software architecture explained in above session. The system architecture consists of data acquisition, preprocessing, inference at edge, action/ output session (Hao *et al.,* 2021). Deep learning is the forefront of the development of machine learning which has achieved excellent results in many ways like natural language processing and computer vision. The most common deep learning method used for edge AI is Convolutional Neural Network (CNN) (Junaidi *et al.,* 2025). It is especially effective for processing data with a grid-like structure, such as images (2D pixels), videos, or even time series. The components of CNN include input layer, convolutional layer, pooling layer, fully connected layer, and output layer (Paymode *et al.,* 2022).

### *The major components*

1. More intricate and important visual features can be automatically extracted by the convolutional layer. The primary purpose of convolutional layers is to extract distinctive information from images. The extraction of input features is made easier by the regular application of convolutional layers (Chen *et al.,* 2020). The features extraction (Hi) among several layers in CNN is computed using the formula below.

$H_i=\varphi(H_{i-1}W_i+b_i)$

Where, Hi - Feature map, Wi –Weight, bi is offset and φ – Rectified Linear Unit (RELU)

2. Because of the convolutional network's large processing capability, the pooling layer lowers the number of data parameters. It minimizes the computational resources required for image processing while also reducing the size of convolved features in dimension. Max pooling and average pooling are the two types of pooling. While average pooling yields the average value of the picture section, max pooling yields the maximum value of the images.
3. Every neuron in the fully connected layer is linked to every neuron in the layer before it. It is also called as dense layer. It is used to combine all extracted features from previous layers ad perform final classification or regression. For this usually SoftMax function is used (Paymode *et al.,* 2022).

The majority of recent studies concentrate on using CNN models to categorize photos of pests. Identifying and locating each pest in photos of the natural environment is more important than classifying pests. Computer vision tasks like object detection could be handled by the CNN model's feature extractors in conjunction with meta-architecture (W *et al.,* 2021). Object identification techniques include single shot multi-box detectors (SSD), region convolutional neural networks (RCNN) (Saleem *et al.,* 2020), faster region convolutional networks (Faster RCNN) (Xie *et al.,* 2020), and you only look once (YOLO) (Zhang *et al.,* 2022). Due to the removal of the region proposal and subsequent pixel or feature resampling, the SSD model is simpler. This deep learning model is referred to as a single-shot multi-box detector since it incorporates all computation into a single network. The use of feature maps for category score, box offset prediction for the collection of default bounding boxes, and small convolution filters like 4x4 and 8x8 are the main characteristics of SSD. Compared to SSD, the object detection process operates at two distinct stages in the Faster RCNN architecture. Using feature extractors rather than an external technique like Edge boxes, the images are processed to produce region suggestions at the region proposal network (RPN) stage. For every intermediate convolutional layer, class-specific suggestions are predicted using these features. Then, the generated anchor boxes are used at the second step of detecting the characteristics of the same immediate layer of an image (Saleem *et al.,* 2020). YOLO is one of the classic one-stage object detection algorithms. The basic concept of YOLO is to convert the object detection to regression to reach higher efficiency and accuracy. Its key method is to divide the input picture into SxS grids. Each grid is responsible for detecting the targets whose centre points falls in the grid (Zhang *et al.,* 2022). Four mainstream backbone architectures for detecting deep learning targets are VGG 16, ResNet101, MobileNet, EfficientNet-B0 (Patil *et al.,* 2022).

## 7. Applications of Edge AI in Crop Disease Detection

Plant diseases pose a major threat to global agricultural productivity and food security by reducing both crop yield and quality. Infections can spread quickly throughout fields and cause large economic losses if they are not detected and managed on time (Mutka and Bart 2015). Conventional manual disease evaluation techniques, including grid counting and visual scoring, are laborious, arbitrary, and prone to mistakes (Cruz *et al.,* 2019). Furthermore, excessive use of chemical pesticides raises production costs, threatens human health, and harms the environment, even if they persist as a common means of control (Patil *et al.,* 2022). Early and accurate disease detection is therefore essential for implementing focused treatments and reducing the use of pesticides. New technologies such as the Internet of Things (IoT) and artificial intelligence (AI) have made it possible to solve this problem. In resource-constrained environments, Edge AI, which carries out AI computations directly on low-power local devices like the Raspberry Pi or NVIDIA Jetson Nano, provides an efficient method for real-time plant disease detection by reducing latency, network dependency, and cost (Gatla *et al.,* 2024).

Crop disease detection has been transformed by the application of Edge AI in agriculture, which allows for localised, real-time analysis and decision-making. Edge AI processes data directly on lightweight devices like smartphones, Raspberry Pi units, NVIDIA Jetson platforms, or ARM microcontrollers, in contrast to traditional cloud-based methods that have concerns with communication delays, bandwidth restrictions, and privacy issues. It is especially appropriate for rural and resource-constrained farming contexts because of its decentralisation, which guarantees quicker responses, less reliance on internet connectivity, and enhanced scalability (James *et al.,* 2024). However, a significant methodological flaw in the current work should be emphasized. Results that are reported often depend on heterogeneous experimental conditions, evaluation measures, and datasets. While some studies use field-collected imagery captured under variable lighting, backgrounds, and disease severity, others use controlled datasets like Plant Village. Metrics include accuracy, latency, or power consumption, and hardware platforms vary from microcontrollers to GPUs. As a result, accuracy values shouldn't be taken as definitive rankings, and direct performance comparisons between studies are problematic. Instead, they show the model's potential in particular experimental conditions. This disclaimer is necessary to ensure equitable interpretation and prevent results from being over generalized.

## *7.1 Edge AI Applications in Cereal Crop Disease Detection*

Global food security is based on cereal crops, especially wheat, rice, maize, and barley. These crops are highly susceptible to bacterial, viral, and fungal diseases that require timely diagnosis. For targeted, crop-specific disease surveillance, edge AI has thus been thoroughly investigated, with solutions designed for classification, detection, and severity assessment tasks.

### *7.1.1 Wheat Disease Detection*

Wheat diseases such as yellow rust, stem rust, leaf rust, and Fusarium head blight (FHB) are among the most economically damaging constraints to wheat production worldwide. Depending on symptom morphology and deployment requirements, edge AI systems for wheat have concentrated on both whole-image classification and object-detection frameworks. Initial study by Lu *et al.* (2017) showed that deep multi-instance learning could be used to diagnose wheat diseases offline using smartphones, proving that meaningful inference could be made even in the absence of internet connectivity. Lightweight convolutional neural networks (CNNs) for embedded hardware deployment were extensively assessed in later studies. Ahsan *et al.* (2024), for instance, benchmarked six CNN architectures on Jetson Nano hardware and demonstrated that small models, like ResNet-18, offered advantageous trade-offs between computational cost, inference time, and accuracy. It's interesting to note that Ui Haq *et al.* (2022) found that for particular rust datasets, classical Random Forest classifiers performed better than deep CNNs. This suggests that on restricted edge devices, computationally economical older machine learning methods may sometimes be preferable. More appropriate object-detection architectures have been found for FHB detection, which necessitates lesion localization on wheat spikes. For Jetson devices, SCS-YOLO was proposed by Gao *et al.* (2025) for real-time field inference. A lightweight YOLOv4 variant was introduced by Hong *et al.* (2022), who used MobileNet-based designs in place of heavy backbones to reduce model size by about 80% while preserving robust detection performance. YOLOv8s-CGF, developed by Yang *et al.* (2024), significantly improved lightweight design by integrating GhostConv and C-FasterNet modules to lower parameters without compromising accuracy. FHBDSR-Net was developed by Wu *et al.* (2025) to measure the diseased spikelets rate, allowing automated phenotyping for resistance breeding. While Shafi *et al.* (2023) implemented ResNet-based classifiers directly on edge devices for farmer usage, Khalid *et al.* (2024) also demonstrated portable embedded systems for yellow rust severity classification utilizing improved GhostNet-like architectures and multispectral sensing.

Three technical insights are highlighted by these studies taken together: (i) classification models work well for diffuse foliar diseases like rusts, (ii) object detection works better for localized symptoms like FHB, and (iii) practical deployment requires balancing accuracy, inference speed, and memory footprint. In order to bridge the gap between laboratory and field conditions, transfer learning, augmentation, and domain adaptability remain crucial.

### *7.1.2 Rice Disease Detection*

Yield stability is seriously threatened by rice diseases like blast and bacterial blight, especially in systems where smallholders predominate. For rice, edge AI system have therefore prioritized portability, ultra-low power consumption, and affordability.

MobileNet-based architectures have been optimized for microcontrollers and small edge boards in a number of studies. Nugroho *et al.* (2024) showed that with carefully pruned MobileNetV2 models, ARM Cortex-M devices could achieve excellent classification accuracy. In order to achieve high performance with minimal computing demands, Zhang *et al.* (2024) suggested a multi-modal system (ISMSFuse) that fused RGB image features with inexpensive spectrum data utilizing a CNN–SVM pipeline. TinyML techniques were further pushed onto microcontroller-class hardware for continuous monitoring by Xan *et al.* (2024). For farmers, smartphone-based technologies provide an accessible solution. Joshi *et al.* (2022) developed RiceBioS, which classified blast and bacterial blight offline and provided real-time feedback. To facilitate multi-scale lesion identification, lightweight YOLOv8 versions and attention-based CNNs have also been tailored for Jetson platforms (Wang *et al.,* 2025; Rana *et al.,* 2025). On Raspberry Pi devices, comparative analyses of MobileNet-V4 and related lightweight models revealed highly accurate real-time inference (Nanda *et al.,* 2025). Additional innovations include FPGA-based accelerators that lower memory and power consumption (Zheng *et al.,* 2025) and augmented-reality glasses and federated learning for in-field blast severity evaluation (Pan *et al.,* 2025). Proactive disease management is further supported by IoT-integrated systems that combine edge inference and environmental sensing (Kumar *et al.,* 2024; Arya and Mishra, 2024).

These advancements show that diagnostics for rice diseases may be included into inexpensive, farmer-friendly devices, allowing for decentralized, continuous monitoring even in areas with limited connectivity.

### *7.1.3 Maize Disease Detection*

In the Edge AI literature, maize received relatively less attention, yet there are a number of potential applications. Smartphone apps based on TensorFlow Lite have made offline multi-crop diagnostics possible, including maize diseases (GC *et al.,* 2024). For on-site detection and farmer interaction, real-time frameworks utilizing explainable AI and lightweight object detectors have been developed (Santhi *et al.,* 2025). Grad-CAM visualizations demonstrated great accuracy and interpretability for field-validated mobile systems that included transformer-enhanced YOLO architectures and MobileNetV2 (Nakatumba-Nabende and Murindanyi, 2025).

For maize, tinyML techniques are very appealing. Deployment on microcontroller units with low RAM and latency was demonstrated by Gookyi *et al.* (2024), allowing for extremely affordable solutions appropriate for smallholder farmers. The viability of portable, offline detection is further demonstrated by earlier smartphone-based MobileNet apps (Barman *et al.,* 2020) and lightweight EfficientNet or MobileNet variations (Bi *et al.,* 2023; Liu *et al.,* 2020; Rajeena *et al.,* 2023). All of these research point to the possibility of integrating maize disease diagnostics into low-cost edge devices, which would enable scalability in environments with limited resources.

### *7.1.4 Barley Disease Detection*

Recently, deep learning frameworks that are compatible with edge technology have been used to identify barley diseases. Rezaei *et al.* (2024) demonstrated the viability of combining edge processing and aerial imaging for large-scale surveillance by proving that MobileNet-based models installed on smartphones and drones could effectively classify a variety of barley diseases. In order to increase robustness in noisy field conditions, Liu *et al.* (2026) suggested a hybrid deep-feature and weighted-ensemble strategy, which produced notable improvements above single-model baselines. These findings demonstrate the need of ensemble approaches and lightweight designs for accurate in-field diagnosis of barley disease.

### *7.1.5 Synthesis and Future Directions*

Edge AI continuously offers three main advantages for cereals: affordability for rural implementation, decreased need on cloud connectivity, and real-time responsiveness. Together, lightweight CNNs (like the MobileNet/EfficientNet families), hardware-aware techniques like quantization, pruning, and FPGA acceleration, optimized object detectors (such YOLO variations), and TinyML implementations make it possible to diagnose diseases on devices with limited resources. System capabilities are further improved by integration with multispectral imagery, UAVs, and IoT sensors.

However, cross-study comparisons are made more difficult by differences in hardware, evaluation methodologies, ambient circumstances, and datasets. Numerous quoted accuracies may not accurately reflect variability in the real world since they are derived from controlled or laboratory datasets. Standardized field benchmarks, cross-location validation, energy-efficiency reporting, and interpretable models that foster farmer trust should thus be given top priority in future research.

All things considered, Edge AI is a revolutionary method for decentralized crop health monitoring. These methods facilitate prompt, accurate, and sustainable disease management by bringing intelligence directly to the field; this is especially important for cereal-based production systems that maintain global food security.

### *7.2 Edge AI Applications in Horticultural and Other Crop Systems: Comparative Context*

Examining Edge AI implementations in horticulture and other agricultural systems offers a useful comparative and methodological background, even though the previous sections focused on cereal crops. In general, the literature on horticulture disease detection is more developed, with an extensive experimentation on real-world deployment techniques, hardware–software co-design, and model compression. Particularly in the areas of hardware selection, lightweight architecture design, and inference pipeline optimization for resource-constrained situations, these studies provide technical lessons that might be applied to the diagnosis of crop diseases. Crucially, even if crop form and symptomatology vary among species, the technical and computational concepts that underpin effective edge deployment are still widely applicable.

Tomato has emerged as one of the most extensively studied crops in Edge AI research due to vulnerability to various foliar diseases, including blight, curl, septoria, and wilt. Compact CNN architectures like GoogLeNet and MobileNetV2 have been shown by Majeed *et al.* (2024) to achieve classification accuracies of over 98% when implemented on embedded hardware, such as Raspberry Pi boards, Google Coral accelerators, and Jetson devices. The study carried out a thorough cost-performance comparison in addition to reporting accuracy and found that, whereas Jetson systems offered faster throughput at higher cost, Raspberry Pi and Coral TPU offered the best combination between affordability and inference speed.

For cereal systems, where widespread deployment over numerous field nodes necessitates lowering per-unit expenditure, this economic analysis is especially informative. Nyakuri *et al.* (2025) extended this idea of extreme efficiency by introducing Tiny-LiteNet, an incredibly small CNN that only takes up 1.2 MB of cache memory on a Raspberry Pi 5 while still achieving 98.6% accuracy. With their incredibly quick inference and low power consumption, cache-resident models offer distinct benefits for distributed cereal disease monitoring networks that depend on a large number of low-cost, battery-operated edge nodes.

The advantages of combining offline functionality and multi-modal sensing are further demonstrated by experiments on the identification of soybean diseases. In order to identify soybean rust and frogeye leaf spot in real time, Darak *et al.* (2025) integrated environmental sensor inputs, such as temperature, humidity, and leaf wetness, with CNN-based image analysis. Rapid field diagnostics were made possible by lowering the inference latency to less than 50 ms with TensorRT optimization and quantization. For cereal crops, where disease outbreaks are heavily influenced by microclimatic conditions, the combination of environmental factors and visual cues is particularly pertinent. For instance, warm, humid weather during flowering significantly raises the risk of wheat Fusarium head blight.

In addition, Alnuaim *et al.* (2025) prioritized user-friendly interfaces for farmers and focused on mobile, internet-independent diagnosis using a lightweight CNN that achieved 94.05% accuracy. The practical implementation of grain disease detection technologies in areas with erratic connectivity depends on several design principles: simplicity, interpretability, and offline capabilities.

CNN-based models installed on edge devices have demonstrated high accuracy and real-time responsiveness in other crops, including cotton, sunflower, castor, and grapevine (Choudhary *et al.,* 2023; Chirasani *et al.,* 2024; Soni *et al.,* 2023; Lazcano-García *et al.,* 2025). These investigations jointly validate the generalizability of lightweight architectures and hardware-aware optimization methodologies, despite the fact that direct transfer of models between crops is rarely possible due to variations in leaf morphology and symptom expression.

Advanced inference acceleration methods developed outside of cereals are especially relevant. For instance, Lv *et al.* (2025) combined convolution-activation operations, TensorRT-accelerated non-maximum suppression, and memory format alignment with hardware architecture to optimize a YOLOX-Tiny peanut disease detector on Jetson Nano. This resulted in a latency reduction of over half and allowed for near-real-time video detection. Similar to this, Zhang *et al.* (2022) used TensorRT acceleration and reduced redundant detection heads to simplify a YOLOv4-tiny derivative for strawberry detection, almost quadrupling inference speed without sacrificing accuracy. For wheat or rice monitoring systems that need continuous, tractor-mounted, or UAV-based video analysis, these methodical pruning and restructure techniques could be immediately applied.

An additional transferable insight is offered by temporal and sequential modeling. In order to improve diagnostic reliability and enable deployment following pruning and quantization, Dhiman *et al.* (2023) presented a CNN–LSTM fusion framework for citrus disease diagnosis that caught the temporal evolution of symptoms across image sequences. For cereals, where disease dynamics–rapid expansion versus delayed progression–often influence management decisions and treatment urgency, such temporally aware models may be particularly helpful. Therefore, adding time-series data from several field observations should improve edge-based cereal diagnostics' accuracy and agronomic significance.

Legume systems also provide instructive parallels. In order to identify rust and angular leaf spots, Katumba *et al.* (2024) developed BeanWatchNet, which uses quantized YOLOv8 and customized CNN models on smartphones and Raspberry Pi 4B devices for in-field use. These findings point to opportunities for transfer learning, where models pre-trained on large bean rust datasets could be refined for wheat or barley rust detection, potentially lowering the need for data specific to cereals, as rust pathogens in beans and cereals exhibit similar pustule-based symptom patterns. Cross-domain adaptation like this could speed up the creation of novel models of cereal diseases.

Lastly, emerging frameworks such as ultra-low-power computing and TinyML significantly increase scalability across crop types. In order to enable sustainable deployment on highly restricted hardware, De Vita *et al.* (2021) showed that a combination of pruning, quantization, and knowledge distillation lowered CNN memory utilization by 85% without compromising accuracy. On microcontroller-class hardware, comparable TinyML solutions for coffee, mango, and apple diseases have demonstrated dependable performance over prolonged battery or solar power runs (Attri *et al.,* 2025; De Vita *et al.,* 2020). These advancements suggest that it is feasible to disperse fully autonomous, inexpensive sensor nodes with continuous disease surveillance capabilities throughout cereal fields, hence facilitating precision agriculture on a large scale. At the same time, architectural innovations such as lightweight vision transformers like Lite-Agro, EdgePlantNet, and LiteViT, attention-enhanced CNNs, and hybrid CNN–ResNet models show that high accuracy can be maintained even with constrained computational budgets (Dockendorf *et al.,* 2023; Priyatharsini and Kumari, 2024; Mudavat *et al.,* 2025).

When combined, horticultural and non-cereal crop studies highlight a number of transferable principles: apply systematic model compression and acceleration, prioritize hardware-aware lightweight architectures, integrate environmental and temporal signals when available, and create farmer-centric, offline-capable interfaces. These engineering techniques offer a solid framework for improving Edge AI deployment in cereal disease detection systems, despite the fact that disease symptoms and datasets vary among crops.

## 8. Challenges on Edge AI

Edge AI is emerging as an intriguing tool for making quick, data-driven decisions in agriculture, but putting it into reality remains a challenge. Farms generate enormous amounts of data every day through sensors, drones, and smartphones, yet edge devices often lack storage and processing capacity to handle such data efficiently (Zhang *et al.,* 2020). Many AI models are trained on clean, controlled datasets like Plant Village, but they performing poorly under real farm conditions where lighting, climate, soil, and crop morphology differ widely. This gap leads to high false positive rates and reduced reliability in practical application (Paul *et al.,* 2025).

Technical constraints further complicate the adoption of Edge AI. Edge device usually has limited computational power, memory, and energy resources and battery life, which makes it hard to run heavy deep learning models and complex convolutional neural networks (CNN). Partitioning AI tasks between devices, edge nodes, and cloud servers adds another layer of complexity, as it directly impacts speed, energy usage, and model accuracy (Tariq *et al.,* 2025).

Poor network coverage and unstable services in rural areas makes things even harder, especially for application requiring real-time alerts such as irrigation or disease outbreaks. Security and privacy concerns are also critical, as the transmission of sensitive agricultural data through wireless networks exposes systems to risks of being tempered and fraud. While encryption and blockchain-based protections exist, they demand additional computation and storage, which resource-constrained devices cannot always support. Another limitation lies in reliability and stability, since frequent device failures, power shortages, or service conflicts between applications can disrupt critical operations (Meuser *et al.,* 2024).

Most current tools developed are still at the experimental stage, and models developed in one region often fails to perform well in other environment, crops or farming practices, reducing their practical applicability. For farmers, especially small holder farmers in developing countries, the high cost of installing and maintaining IOT devices, servers, and connectivity also a big barrier. From a research perspective, edge AI in agriculture remains under-explored, compared to domain like healthcare or smart cities, with limited smartphone-based tools reaching real-world deployment (Meuser *et al.,* 2024).

## 9. Conclusion and Future Perspectives

### *9.1 Critical Assessment of Current State*

This review reveals that Edge AI has progressed from proof-of-concept demonstrations to increasingly sophisticated disease detection systems across multiple crops. Edge AI holds particular promise for cereal crops, which are the focus of global food security: rice blast detection benefits from ultra-low-power TinyML implementations appropriate for distributed sensor networks, maize disease diagnosis has been successfully implemented on farmer-accessible smartphone platforms, and wheat rust monitoring can utilize lightweight YOLO architectures for real-time field surveillance. These days, complex CNN and transformer models may function within the computational limitations of edge devices owing to hardware advancements (Edge TPU, Jetson platforms, ARM microcontrollers) and optimization approaches (8-bit quantization, structural pruning, knowledge distillation).

However, this optimism is tempered by important constraints. The biggest obstacle is still model generalization: systems developed on controlled datasets (like PlantVillage) often perform poorly in real-world settings with diverse illumination, intricate backgrounds, a range of diseases, and symptoms that vary between cultivars and growth phases. There is a concerning trend in the literature when models come into contact with field heterogeneity, stated accuracies of 95–99% on homogenous test sets frequently drop significantly. This problem is exacerbated for cereal crops by abiotic stress mimicking (e.g., nutrient deficit mirroring disease symptoms) and symptom similarity among diseases (e.g., yellow rust vs. stripe rust in wheat).

Dataset limitations further constrain progress. The majority of research relies on single-season dataset collection, restricted geographic representation, and lab-captured images that don't accurately depict field circumstances. In particular, rice disease collections under represent cultivar diversity, wheat rust databases lack temporal progression data, and publicly accessible databases for barley diseases are limited. Although local disease manifestations differ from those of well-studied populations, this data scarcity is especially problematic in developing nations, where Edge AI might have the biggest impact.

Standardization of evaluation is notably absent. Cross-study comparisons are problematic due to the use of conflicting measurements (accuracy vs. F1-score vs. mAP), a variety of hardware platforms, and non-comparable experimental procedures. Inconsistent reporting of energy consumption, inference delay under realistic workloads, and model resistance to input perturbations make it difficult to choose the right hardware and deployment algorithms.

### *9.2 Future Research Directions for Cereal Crop Disease Detection*

To advance Edge AI from experimental systems to reliable field tools for cereal disease management, we identify the following priority research directions:

#### *9.2.1 Domain Adaptation and Robust Learning*

Multi-location field validation: Develop standardized protocols for assessing models in a range of geographical areas, growing seasons, and cultivars. This calls for datasets covering various Puccinia races and environmental circumstances in order to detect wheat rust.

Few-shot and meta-learning techniques: These methods allow for quick adaptation to novel diseases, cultivars, or geographical areas with little labeled data, which is essential for agricultural systems with limited resources.

Data augmentation and aerial training: By using methodical augmentation techniques based on field conditions, increase resilience to real-world variables (lighting changes, occlusion, and camera quality).

Transfer learning across cereal species: To make use of the few disease-specific datasets available, look at cross-crop knowledge transfer (e.g., rust detection algorithms trained on wheat modified for barley).

### 9.2.2 Multimodal and Temporal Integration

- Environmental context integration: To improve diagnostic specificity, combine visual disease detection with data on temperature, humidity, and leaf wetness. This is especially useful for differentiating between disease and abiotic stress in cereals.
- Modeling the temporal progression of symptoms: Create CNN-LSTM or transformer architectures that examine the evolution of symptoms over time to enable earlier diagnosis and severity forecasting. This is particularly important for diseases like wheat Fusarium head blight that spread quickly.
- Multispectral and hyperspectral sensing: Combine inexpensive multispectral sensors with edge platforms to identify infections before symptoms appear, especially for diseases with distinctive spectral signatures.

### 9.2.3 Hardware and Optimization Innovation

- Ultra-low-power TinyML systems: These sophisticated microcontroller-class implementations (less than 1W of power consumption) allow for autonomous field sensors that run on batteries or solar power for continuous cereal crop monitoring.
- Hardware-software co-design: Instead of modifying general-purpose models, build disease detection algorithms that are especially tailored for the target hardware (such as the ARM Cortex-M or Edge TPU).
- Exploration of neuromorphic computing: Investigate spiking neural networks and event-based vision sensors for extremely high energy efficiency in situations involving continuous monitoring.
- Federated learning for edge networks: Farmers can gain from collective learning without disclosing sensitive farm data, which enables cooperative model improvement across dispersed edge devices while protecting data privacy.

### 9.2.4 Cereal-Specific Technological Priorities

- Wheat rust surveillance networks: Combine ground-level monitoring with UAV-based overhead surveillance to implement dispersed edge sensor systems for early rust epidemic warning.
- Quantification of Fusarium head blight: Create mechanisms for spike-level detection and severity assessment to aid in breeding initiatives and maximize the timing of fungicide applications.
- Rice blast forecasting: Use weather data and edge-based image processing to create predictive disease risk models that allow for proactive management.
- Creation of barley disease datasets: To overcome the existing data shortage and facilitate the development of robust models, create extensive, multi-location datasets for barley diseases.

### 9.2.5 User-Centric Design and Deployment

- Explainable AI for farmer trust: Combine natural language explanations with attention visualization (Grad-CAM) to assist farmers in comprehending and having faith in diagnostic results.
- Large language model integration: Investigate the deployment of lightweight LLM on edge devices to offer context-specific guidance, management suggestions, and conversational interfaces in local languages.
- Cost-benefit analysis frameworks: To inform adoption decisions, provide economic models that quantify the Edge AI value proposition for various crop systems and farm sizes.
- Offline-first architecture: Prioritize designs that operate completely offline, with the possibility of cloud synchronization for aggregated analytics and model updates.

### 9.3 Concluding Perspective

Edge AI is an revolutionary approach of managing plant diseases, especially for cereal crops that are essential to the world's food security. Critical drawbacks of traditional methods and cloud-based AI systems are addressed by Edge AI, which makes real-time, localized diagnosis possible without relying on the cloud. Lightweight deep

**Table 2:** Applications of Edge AI in Crop Disease Detection

| Crop(s) | Disease | Edge Hardware/ Platform | Model / Framework | Key Optimization / Design Strategy | Reported Outcome | Transferable Insight | Reference |
|---|---|---|---|---|---|---|---|
| Wheat | Multi-disease diagnosis | Smartphone (offline) | Deep Multi-Instance Learning | On-device inference, no cloud | Field-feasible offline detection | Demonstrated early feasibility of mobile edge AI | Lu (2017) |
| Wheat | Yellow rust | NVIDIA Jetson Nano | ResNet-18, DenseNet-101, EfficientNet, etc. | Lightweight CNN benchmarking | ~87% accuracy | Smaller CNNs give better speed–accuracy trade-off | Ahsan *et al.,* (2024) |
| Wheat | Rust | Edge devices | Random Forest vs CNN | Classical ML | 97.3% (RF) | Traditional ML may outperform DL on constrained hardware | Ui Haq (2022) |
| Wheat | FHB detection | Jetson Nano | SCS-YOLO | Custom lightweight YOLO | Real-time inference | Object detection for spike-level symptoms | Gao *et al.,* (2025) |
| Wheat | FHB (ear detection) | UAV + edge | YOLOv4-MobileNet | Backbone compression, focal loss | 93.69%, 80% smaller model | UAV-based rapid surveillance | Hong *et al.,* (2022) |
| Wheat | FHB | Mobile/edge | YOLOv8s-CGF | GhostConv, reduced params | Compact + high mAP | Hardware-aware model redesign | Yang *et al.,* (2024) |
| Wheat | Spikelet severity (DSR) | Mobile/edge | FHBDSR-Net | Multi-scale + attention | 93.8% AP | Automated phenotyping tool | Wu *et al.,* (2025) |
| Wheat | Yellow rust severity | Portable embedded | GhostNet-like CNN | Multispectral sensing | ~95% | Multimodal improves robustness | Khalid *et al.,* (2024) |
| Wheat | Rust severity classes | Edge device | ResNet-50 | Background removal + classification | 96% | Direct farmer-deployable device | Shafi *et al.,* (2023) |
| Rice | Blast, blight | ARM Cortex-M MCU | MobileNetV2 | Pruning, TinyML | 97.5% | Ultra-low-power detection | Nugroho (2024) |
| Rice | Bacterial blight | Raspberry Pi | CNN + SVM fusion | Image + spectral sensing | 98.14% | Multimodal fusion improves accuracy | Zhang *et al.,* (2024) |
| Rice | Blast/ blight | Smartphone | RiceBioS CNN | Model shrinking, offline | 93.25% | Farmer-friendly mobile diagnosis | Joshi *et al.,* (2022) |
| Rice | Multi-disease | Jetson Nano | YOLOv8 lightweight | Multi-scale detection | Real-time | Early lesion detection | Wang *et al.,* (2025) |
| Rice | Multi-disease | Raspberry Pi 5 | MobileNet-V4 | Focal loss, augmentation | 97.84% | Efficient CNNs for real-time edge | Nanda *et al.,* (2025) |
| Rice | Leaf disease | FPGA | Quantized MobileNetV2 | 16-bit quantization, pipelining | 95.8%, low power | Energy-efficient hardware acceleration | Zheng *et al.,* (2025) |
| Maize | Leaf diseases | Smartphone | MobileNet / TFLite | Offline inference | ~93–97% | Low-cost apps for smallholders | Barman *et al.,* (2020); GC *et al.,* (2024) |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Maize | Multi-disease | Mobile/edge | Transformer-YOLO + Grad-CAM | Explainable AI | mAP 0.995 | Interpretability builds trust | Nakatumba-Nabende & Murindanyi (2025) |
| Maize | Leaf diseases | MCU (TinyML) | Custom CNN | Memory-efficient design | 94.6% | Ultra-low-cost deployment | Gookyi *et al.*, (2024) |
| Barley | Net blotch, scald | Smartphone + drone | MobileNet | Data augmentation | 93.5% | Drone + edge for large fields | Rezaei *et al.*, (2024) |
| Barley | Multi-disease | Edge | CNN + ensemble | Weighted voting | 98.7% | Ensembles increase robustness | Liu *et al.*, (2026) |
| Tomato | Foliar diseases | Raspberry Pi, Coral TPU, Jetson | GoogLeNet/ MobileNetV2 | Cost–performance analysis | >98% | Coral TPU best price–performance | Majeed (2024) |
| Tomato | Foliar diseases | Raspberry Pi 5 | Tiny-LiteNet | 1.2 MB model | 98.6% | Cache-resident ultra-fast inference | Nyakuri *et al.*, (2025) |
| Soybean | Rust, frogeye | Edge + sensors | CNN + environmental data | TensorRT, quantization | <50 ms latency | Weather + vision improves prediction | Darak (2025) |
| Soybean | Leaf diseases | Mobile | Lightweight CNN | Offline UI-focused design | 94.05% | Farmer-centric design | Alnuaim (2025) |
| Peanut | Leaf diseases | Jetson Nano | YOLOX-Tiny | Conv fusion, EfficientNMS | 28.7 FPS | Video-rate detection optimization | Lv (2025) |
| Strawberry | Detection | Jetson Nano | RTSD-Net | Structural pruning | 25 FPS | Remove redundant layers for speed | Zhang (2022) |
| Citrus | Fruit diseases | Edge | CNN–LSTM | Temporal modeling + pruning | 98.25% | Time-series improves diagnosis | Dhiman (2023) |
| Bean | Rust, ALS | Smartphone + Pi | YOLOv8 / BeanWatchNet | Quantization | 90–87.6% | Transfer learning potential for cereal rust | Katumba *et al.*, (2024) |
| Apple/ Mango/ Coffee | Multiple | MCU/ TinyML | Quantized CNNs | Pruning + distillation | 90–93%+ | Solar/battery autonomous nodes | Attri (2025); De Vita *et al.*, (2020, 2021) |

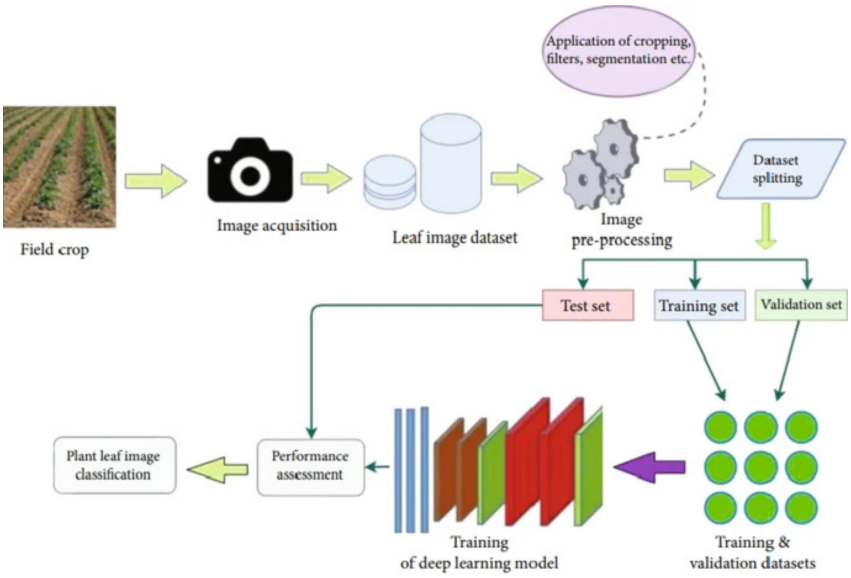


Fig 2: Pipeline of Plant Disease Detection (Tiwari *et al.*, 2021)

learning architectures, effective hardware accelerators, and advanced optimization techniques are examples of current technology capabilities that have shown technically feasible in a variety of cereal disease detection scenarios.

However, resolving basic issues with model robustness, dataset diversity, and practical usability is necessary to move from laboratory demonstration to dependable field implementation. The focus of the research community needs to change from optimizing accuracy on controlled datasets to creating systems that continue to function satisfactorily in the face of real-world variability. This entails developing disease detection technologies that are dependable across cultivars, growth stages, geographical areas, and farming practices, particularly for cereal crops.

Combining cutting-edge technology such as lightweight LLMs, federated learning, multimodal sensing, and few-shot learning presents encouraging paths to getting over present constraints. Adopting user-centric design concepts that put farmer needs first, guarantee affordability, and offer practical insights rather than just technical results is equally crucial. In the end, Edge AI's contribution to sustainable cereal production will be evaluated based on its practical effects rather than its algorithmic complexity: earlier disease detection that allows for targeted interventions; precision application that reduces pesticide use; yield protection that improves food security; and improved farmer livelihoods due to easily accessible and reasonably priced technology. Continued innovation, thorough field validation, and cooperation between the computer science, plant pathology, agricultural engineering, and farming communities are necessary to achieve this vision.

## Author contributions

*Tamanna Chauhan*: Compilation and writing original draft; *Ankita Kailas Kurhade*: Literature survey and drafting the manuscript; *Sivadinesh Chinnadurai*: Data curation and tabulation; *Ashok Beniwal*: Literature survey and figures; *Chandra Nath Mishra*: Editing; *Satish Kumar*: Conceptualization, drafting and editing the final draft; *Om Prakash Ahlawat*: Editing; *Ratan Tiwari*: Editing and supervision

## Conflict of interest statement

On behalf of all authors, the corresponding author states that there is no conflict of interest.

## Ethical Approval

The article doesn't contain any data involving ethical approvals.

## Generative AI or AI/Assisted technologies used in manuscript preparation

No

## References

1. Abadi M, AAgarwal, P Barham, E Brevdo, Z Chen and C Citro. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv. https://doi.org/10.48550/arXiv.1603.04467
2. Abdelmadjid SAAD and AIDI Abdeldjallil. 2024. Optimized Deep Learning Models For Edge Computing: A Comparative Study on Raspberry PI4 For Real-Time Plant Disease Detection. *Proceedings - International Conference on Distributed Computing Systems*, 273-278. http:// doi.org/10.1109/EDiS63605.2024.10783415.
3. Ahamed MF, A Salam, M Nahiduzzaman, M Abdullah-Al-Wadud and SR Islam. 2024. Streamlining plant disease diagnosis with convolutional neural networks and edge devices. *Neural Computing and Applications*, **36**(29): 18445-18477. https://doi.org/10.1007/s00521-024-10152-y
4. Ahmed AA and GH Reddy. 2021. A mobile-based system for detecting plant leaf diseases using deep learning. *Agri Engineering*, **3**(3): 478-493. https://doi.org/10.3390/agriengineering3030032
5. Ahsan A, MS Iqbal, M Ahmar, M Adnan, MA Akbar and A Bermak. 2024. Edge computing based early yellow rust disease detection in wheat plants. Proc. *Int. Conf. Microelectron.* Doha, Qatar 1-6, https://doi.org/10.1109/ICM63406.2024.10815846
6. Alnuaim A, A Altheneyan and AA AlZubi. 2025. Early Leaf Disease Detection of Soybean Plants using Convolution Neural Network Algorithm. *Legume Research*,**48**(6).
7. Arya A and PK Mishra. 2024. MobileNetV2-Incep-M: a hybrid lightweight model for the classification of rice plant diseases. *Multimedia Tools and Applications*, **83**(33): 79117-79144.
8. Attri I, LK Awasthi and TP Sharma. 2025. Tiny ML for plant disease detection: Efficient edge AI

solutions for apple and mango leaves. *Procedia Computer Science*, **258**: 2870-2877. https://doi.org/10.1016/j.procs.2025.04.547

9. Banbury C, C Zhou, I Fedorov, R Matas, U Thakker, D Gope, *et al.,* and P Whatmough. 2021. Micronets: Neural network architectures for deploying tinyml applications on commodity microcontrollers. *Proceedings of the IEEE International Conference on Computer Vision*, **3**: 517-532.

10. Barbedo JGA. 2019. Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, **180**: 96-107. https://doi.org/10.1016/j.biosystemseng.2019.02.002

11. Barman U, D Sahu and GG Barman. 2020. A deep learning based android application to detect the leaf diseases of maize. *In Proceedings of the Sixth International Conference on Mathematics and Computing: ICMC 2020*, 275-286. Singapore: Springer Singapore.

12. Bhattacharyya T, AM Roy, S Ghosh and P Ghosal. 2024. Plant Disease Detection in Smart Agriculture: A Power-aware Edge-AI Implementation on Cortex-A53. *IEEE International Symposium on Smart Electronic Systems (iSES), New Delhi, India* 293-298. IEEE. https://doi.org/10.1109/iSES63344.2024.00069.

13. Bi C, S Xu, N Hu, S Zhang, Z Zhu and H Yu. 2023. Identification method of corn leaf disease based on improved Mobilenetv3 model. *Agronomy,* **13**(2):300.

14. Bock C H, GH Poole, PE Parker andTR Gottwald. 2010. Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. *Critical Reviews inPlant Sciences,***29**(2): 59-107. https://doi.org/10.1080/07352681003617285

15. Capra M, R Peloso, G Masera, M Ruo Roch and M Martina. 2019. Edge computing: A survey on the hardware requirements in the internet of things world. *Future Internet,* **11**(4): 100. https://doi.org/10.3390/fi11040100

16. Chang Z, S Liu, X Xiong, Z Cai and G Tu. 2021. A survey of recent advances in edge-computing-powered artificial intelligence of things. *IEEE Internet of Things Journal*, **8**(18): 13849-13875. https://doi.org/10.1109/JIOT.2021.3088875.

17. Chen J and X Ran. 2019. Deep learning with edge computing: A review. *Proceedings of the IEEE*, **107**(8): 1655-1674.https://doi.org/ 10.1109/JPROC.2019.2921977

18. Chen J, J Chen, D Zhang, Y Sun and YA Nanehkaran. 2020. Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*,**173**: 105393. https://doi.org/10.1016/j.compag.2020.105393

19. Chen T, T Moreau, Z Jiang, L Zheng, E Yan, Shen H, *et al.,* and A Krishnamurthy. 2018. {TVM}: An automated {End-to-End} optimizing compiler for deep learning. *In 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)* 578-594.

20. Chen Y, T Chen, Z Xu, N Sun and O Temam. 2016. Dian Nao family: energy-efficient hardware accelerators for machine learning. *Communications of the ACM*,**59**(11): 105-112. https://doi.org/10.1145/2996864

21. Chirasani SKR, T Prabakaran, S Fairooz, P Munaswamy,M Ashok, G Sravanthi, *et al.,* N Rajeswaran. 2024. Edge Architecture for High-Accuracy Disease Identification in Apple Plants Using Transfer Learning Approach. Traitement du Signal. **41**(3): 1495

22. Choudhary N, V Kukreja, R Sharma, L Gopal and D Rawat. 2023. Cutting-Edge AI for Helianthus Disease Detection: A CNN-KNN Hybrid Model. *4th IEEE Global Conference for Advancement in Technology (GCAT), Bangalore, India,* 1-6. https://doi.org/10.1109/GCAT59970

23. Cruz A, Y Ampatzidis, R Pierro, A Materazzi, A Panattoni, L De Bellis and A Luvisis. 2019. Detection of grapevine yellows symptoms in Vitis vinifera L. with artificial intelligence. *Computers and Electronics in Agriculture,***157**: 63-76. https://doi.org/10.1016/j.compag.2018.12.028

24. Darak M, SRT Marthwada, MD Wangikar, MR Mahamune. 2025. Development of a Real-Time Disease Detection System Using IoT and Edge

AI for Soybean Farming. *Journal of Computational Science*, **18**(05): 1549-3636

25. Darmawan H, M Yuliana, MZS Hadi, and AK Sangaiah. 2024. Deep metric learning with augmented latent fusion and response-based knowledge distillation on edge device for paddy pests and disease identification. *JOIV: International Journal on Informatics Visualization*, **8**(3-2). https://doi.org/10.62527/joiv.8.3-2.3104

26. David R, J Duke, A Jain, V Janapa Reddi, N Jeffries, J Li, *et al.,* and R Rhodes. 2021. Tensorflow lite micro: Embedded machine learning for tinyml systems. *Proceedings of the IEEE International Conference on Computer Vision*, **3**: 800-811.

27. De Vita F, G Nocera, D Bruneo, V Tomaselli, D Giacalone, SK Das. 2020. Quantitative analysis of deep leaf: a plant disease detector on the smart edge. *IEEE International Conference on Smart Computing SMARTCOMP, Bologna, Italy* 49-56. https://doi.org/10.1109/SMARTCOMP50058.2020.00027.

28. De Vita F, G Nocera, D Bruneo, V Tomaselli, D Giacalone, SK Das. 2021. Porting deep neural networks on the edge via dynamic K-means compression: A case study of plant disease detection. *Pervasive and Mobile Computing*, **75**: 101437 https://doi.org/10.1016/j.pmcj.2021.101437

29. Deepti K. 2023. Comparative analysis of machine learning techniques for plant disease detection-data deployment. *Journal of The Institution of Engineers (India): Series B*, **104**(4): 837-849. https://doi.org/10.1007/s40031-023-00897-w

30. Dhiman P, A Kaur, Y Hamid, E Alabdulkreem, H Elmannai, N Ababneh. 2023. Smart disease detection system for citrus fruits using deep learning with edge computing. *Sustainability*, 15(5): 4576. https://doi.org/10.3390/su15054576

31. Dockendorf C, A Mitra, SP Mohanty, E Kougianos. 2023. Lite-agro: exploring light-duty computing platforms for IoAT-edge ai in plant disease identification. IFIP *International Conference on Very Large Scale Integration and System-on-Chip VLSI-SoC,* **684**: 371-380. Cham Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-45882-8_25.

32. Ferentinos KP. 2018. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, **145**: 311-318. https://doi.org/10.1016/j.compag.2018.01.009

33. Gao C, B He, W Guo, Y Qu, Q Wang and W Dong. 2025. SCS-YOLO: A real-time detection model for agricultural diseases–A case study of wheat fusarium head blight. *Computers and Electronics in Agriculture*, **238**: 110794.

34. Gatla A, SP Reddy, D Mandru, S Thouti, J Kavitha, ASE Souissi, *et al.,* and A Flah. 2024. Optimizing Edge AI for Tomato Leaf Disease Identification. Engineering. *Engineering, Technology and Applied Science Research,* **14**(4): 16061-16068. https://doi.org/10.48084/etasr.7802

35. GC H, GU Sanjana, MR Sanjana, BV Shreya. Edible Plant Disease Detection Using Edge AI. GM UNIVERSITY Karnataka, INDIA 2(1): 52.

36. Gokulnath BV and GU Devi. 2020. A survey on plant disease prediction using machine learning and deep learning techniques. *Inteligencia Artificial*, **23**(65): 136-154. https://doi.org/10.4114/intartif.vol23iss65pp136-154

37. Gookyi DAN, FA Wulnye, EAE Arthur, RK Ahiadormey, JO Agyemang, KOBO Agyekum and R Gyaang. 2024. TinyML for smart agriculture: Comparative analysis of TinyML platforms and practical deployment for maize leaf disease identification. *Smart Agricultural Technology*, **8**:100490.

38. Gookyi DAN, FA Wulnye, M Wilson, P Danquah, SA Danso, AA Gariba. 2024. Enabling intelligence on the edge: Leveraging Edge Impulse to deploy multiple deep learning models on edge devices for tomato leaf disease detection. *Agricultural Engineering*, **6**(4): 3563-3585.https://doi.org/10.3390/agriengineering6040203

39. Government of India. 2022. Agricultural Statistics at a Glance. Ministry of Agriculture & Farmers Welfare.

40. Han S, J Pool, J Tran and W Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing* systems, 28.

41. Hao C, J Dotzel, J Xiong, L Benini, Z Zhang and D Chen. 2021. Enabling design methodologies

and future trends for edge ai: Specialization and codesign. *IEEE Design & Test*, **38**(4): 7-26. https:// doi.org/ 10.1109/MDAT.2021.3069952

42. Haq IU, R Mumtaz, M Talha, Z Shafaq and M Owais. 2022. Wheat Rust disease classification using edge-AI. *2nd Proc. 2015 International Conference on Artificial Intelligence*, ICAI, Islamabad, Pakistan 58-63. https:// doi.org/10.1109/ICAI55435.2022.9773489.

43. Heidorn C, F Hannig, D Riedelbauch, C Strohmeyer and J. Tich. 2024. Efficient Deployment of Neural Networks for Thermal Monitoring on AURIX TC3xx Microcontrollers. *International Conference on Vehicle Technology and Intelligent Transport Systems VEHITS – Proceedings,* 64-75. https://doi. org/10.5220/0012563100003702

44. Hinton G, O Vinyals and J Dean. 2015. Distilling the knowledge in a neural network. arXiv. https:// doi.org/10.48550/arXiv.1503.02531

45. Hong Q, L Jiang, Z Zhang, S Ji, C Gu, W Maoand C Tan. 2022. A lightweight model for wheat ear fusarium head blight detection based on RGB images. *Remote Sensing*, **14**(14):3481

46. Howard A, M Sandler, G Chu, LC Chen, B Chen and M Tan. 2019. Searching for mobilenetv3. *Proceedings of the IEEE International Conference on Computer Vision,* 1314-1324. https://doi.org/10.48550/ arXiv.2212.03332

47. Hymel S, C Banbury, D Situnayake, A Elium, C Ward andM Kelcey. 2022. Edge impulse: An mlops platform for tiny machine learning. arXiv

48. Hymel S, C Banbury, D Situnayake, A Elium, C Ward, M Kelcey, *et al.,* and VJ Reddi. 2022. Edge impulse: An mlops platform for tiny machine learning. *arXiv preprint arXiv:2212.03332.*

49. Iftikhar S, HA Khattak, A Saadat, Z Ameer and M Zakarya. 2024. Efficient fruit disease diagnosis on resource-constrained agriculture devices. *Journal Saudi Society Agriculture Sciences*, https://doi. org/10.1016/j.jssas.2024.07.002

50. J, SB. 2024. Edge-AI based Plant Leaf Disease Identification and Prevention for Smart Agriculture. *Grenze International Journal of Engineering & Technology* (GIJET),**10**(2): 6556

51. Jacob B, S Kligys, B Chen, M Zhu, M Tang, A Howard, *et al.,* D Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2704-2713.

52. James C. 2024. Edge AI in Smart Agriculture: Enabling Low-Latency, Data-Driven Crop Management.

53. Joshi P, D Das, V Udutalapally, MK Pradhan and S Misra. 2022. Ricebios: Identification of biotic stress in rice crops using edge-as-a-service. *IEEE Sensors,***22**(5): 4616-4624. https://doi.org/ 10.1109/ JSEN.2022.3143950

54. Junaidi A, SZM Hashim, MS Bin Othman, MM Mohamad, H Alhussian and SJ Abdulkadir. 2025. Deep Learning and Edge Computing in Agriculture. *International Journal on Recent and Innovation Trends in Computing and Communication,***13**: 137464-137490. https://doi.org/ 10.1109/ACCESS.2025.3595621.

55. Kamilaris A and FX Prenafeta-Boldú. 2018. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture,***147**: 70-90. https://doi. org/10.1016/j.compag.2018.02.016

56. Katumba A, WS Okello, S Murindanyi, J Nakatumba-Nabende, M Bomera, BW Mugalu, *et al.,* A Acur. 2024. Leveraging edge computing and deep learning for the real-time identification of bean plant pathologies. *Smart Agricultural Technology,***9**: 100627. https://doi.org/10.1016/j.atech.2024.100627

57. Khalid H. 2024. Portable Edge Computing System Analysis for Real-time Detection and Classification of Wheat Yellow Rust Infection Types Using Embedded AI. In 2024 *International Conference on Engineering & Computing Technologies* (ICECT), 01-04. IEEE. http:// dx.doi.org/10.1109/ICECT61618.2024.105 81008

58. Khan AT, SM Jensen, AR Khan and S Li. 2023. Optimized Deep Learning Models For Edge Computing: A Comparative Study on Raspberry PI4 For Real-Time Plant Disease Detection. *Ieeexplore.Ieee.Org, 14.* https://doi.org/10.3389/ FPLS.2023.1308528/

59. Konstantinos P Ferentinos. 2018. Deep learning models for plant disease detection and diagnosis.

*Computer and Electronics in Agriculture,***145**:311–318. https://doi.org/10.1016/j.compag.2018.01.009

60. Kumar JS, P Rashmita, MS Pushpitha and K Deepika. 2024. SAMS: Smart Agricultural Monitoring System Integrating Edge AI and IoT for Early Detection of Rice Crop Stress. *In 2023 4th International Conference on Intelligent Technologies (CONIT),* 1-5. IEEE.

61. Lazcano-García C, KG García-Resendiz, J Carrillo-Tripp, E Inzunza-Gonzalez, EE García-Guerrero, D Cervantes-Vasquez, *et al.,* OA Aguirre-Castro. 2025. Deep Learning-Based System for Early Symptoms Recognition of Grapevine Red Blotch and Leafroll Diseases and Its Implementation on Edge Computing Devices. *Agricultural Engineering,* **7**(3): 63. https://doi.org/10.3390/agriengineering7030063

62. Li H. 2022. Acceleration of deep learning applications using Intel distribution of Open VINO toolkit . Dissertation, University of Illinois at Urbana-Champaign.

63. Li W, T Zheng, Z Yang, M Li, C Sun and X Yang. 2021. Classification and detection of insects from field images using deep learning for smart pest management. *Ecological Informatics,***66**: 101460. https://doi.org/10.1016/j.ecoinf.2021.101460

64. Lin J, L Zhu, WM Chen, WC Wang and S Han. 2023. Tiny machine learning: Progress and futures. *IEEE Circuits and Systems Magazine,***23**(3): 8-34. https//doi.org/ 10.1109/MCAS.2023.3302182.

65. Liu J, M Wang, L Bao and X Li. 2020, December. EfficientNet based recognition of maize diseases by leaf image classification. *In Journal of Physics: Conference Series,* **1693**(1):012148. IOP Publishing.

66. Liu X, SM Javidan, Y Ampatzidis and Z Zhang. 2026. A Hybrid Model of Deep Feature Extraction and Weighted Ensemble Classifiers for Accurate Barley Disease Detection. *Journal of the ASABE,* **69**(1):13-24.

67. Lu J, J Hu, G Zhao, F Mei and C Zhang. 2017. An in-field automatic wheat disease diagnosis system. *Computers and Electronics in Agriculture,***142**: 369-379. https://doi.org/10.1016/j.compag.2017.09.012

68. Lv Z, S Yang, S Ma, Q Wang, J Sun, L Du and H Zhang. 2025. Efficient Deployment of Peanut Leaf Disease Detection Models on Edge AI Devices. *Agriculture*, **15**(3): 332. https://doi.org/10.3390/agriculture15030332

69. Lv Z, S Yang, S Ma, Q Wang, J Sun, L Du, ... and H Zhang. 2025. Efficient Deployment of Peanut Leaf Disease Detection Models on Edge AI Devices. *Agriculture*, **15**(3):332.

70. Majeed Y, MO Ojoand A Zahid. 2024. Standalone edge AI-based solution for Tomato diseases detection. *Smart Agricultural Technology,***9**: 100547. https://doi.org/10.1016/j.atech.2024.100547

71. Mattihalli C, F Gared and L Getnet. 2021. Automatic plant leaf disease detection and auto-medicine using IoT technology. *In Intelligent Computing and Innovation on Data Science,* (pp. 101-109). Springer. https://doi.org/10.1007/978-3-030-71172-6_11

72. Meuser T, L Lovén, M Bhuyan, SG Patil, S Dustdar, A Aral. 2024. Revisiting edge ai: Opportunities and challenges. *IEEE Internet Computing,* **28**(4): 49-59.

73. Mohanty SP, DP Hughesa and M Salathé. 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, **7**: 215232. https://doi.org/10.3389/fpls.2016.01419

74. Mudavat SM, A Mitra, SP Mohanty, E Kougianos. 2025. LiteViT: Leveraging the Power of Transformers for Edge AI in Crop Disease Classification. *IEEE Computer Society Annual Symposium on VLSI (ISVLSI),***1**: 1-6. https://doi.org/10.1109/ISVLSI65124.2025.11130336.

75. Mutka AM and RS Bart. 2015. Image-based phenotyping of plant disease symptoms. *Frontiers in Plant Science,***5**: 734. https://doi.org/10.3389/fpls.2014.00734

76. Nakatumba-Nabende J and S Murindanyi. 2025. Deep learning models for enhanced in-field maize leaf disease diagnosis. *Machine Learning with Applications*, 100673.

77. Nanda TR, A Shukla, TR Srinivasa, J Bhargava and S Chauhan. 2025. Advancing real-time crop disease detection on edge computing devices using lightweight convolutional neural networks. *In Intelligent systems conference*, 537-552. Cham: Springer Nature Switzerland

78. Nocker M, L Henke and PSchöttle. 2025. FHE ML Tuxedo: A Tailored Wrapper Architecture for Homomorphic Encryption in Machine Learning. *55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 266-273. IEEE. https://doi.org/10.1109/DSN-W65791.2025.00075.

79. Nugroho H, JX Chew, S Eswaran and FS Tay. 2024. Resource-optimized cnns for real-time rice disease detection with ARM cortex-M microprocessors. *Plant Methods*, **20**(1): 159. https://doi.org/10.1186/s13007-024-01280-6

80. Nyakuri JP, C Nkundineza, O Gatera, K Nkurikiyeyezu and G Mwitende. 2025. AI and IoT-powered edge device optimized for crop pest and disease detection. *Scientific Reports*,**15**(1): 22905. https://doi.org/10.1038/s41598-025-06452-5

81. Oliver A, A Odena, CA Raffel, ED Cubuk and I Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in Neural Information Processing Systems*, 3235–3246.

82. ONNX. 2017. Open Neural Network Exchange. https://onnx.ai. Accessed 1 December 2025

83. Pan P, GUO Wenlong, L Hengbo, SHAO Yifan, G Zhihao, JIN Ye and J Zhang. 2025. Accelerating Wild Rice Disease-Resistant Germplasm Exploration: Artificial Intelligence (AI)-Powered Wild Rice Blast Disease Level Evaluation and Disease-Resistance Identification. *Rice Science*, **32**(5): 727-746.

84. Patil RR and S Kumar. 2022. Priority selection of agro-meteorological parameters for integrated plant diseases management through analytical hierarchy process. *International Journal of Electrical and Computer Engineering*,**12**(1):649-659. https://doi.org/10.11591/ijece.v12i1.pp649-659

85. Patil RR, S Kumar, S Chiwhane, R Rani and SK Pippal. 2022. An artificial-intelligence-based novel rice grade model for severity estimation of rice diseases. *Agriculture*, **13**(1):47. https://doi.org/10.3390/agriculture13010047

86. Paul H, H Udayangani, K Umesha, N Lankasena, C Liyanage and K Thambugala. 2025. Maize leaf disease detection using convolutional neural network: a mobile application based on pre-trained VGG16 architecture. *New Zealand Journal of Crop and Horticultural Science*,**53**(2):367-383. https://doi.org/10.1080/01140671.2024.2385813

87. Paymode AS and VB Malode. 2022. Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. *Artificial Intelligence in Agriculture*,**6**:23-33. https://doi.org/10.1016/j.aiia.2021.12.002

88. Priyatharsini RD and GRNKumari. 2024. Cutting-edge AI for Plant Health: Advanced Residual Neural Networks in Leaf Diseases Segmentation and Classification, 3rd edn. *IEEE Delhi Section Flagship Conference (DELCON) New Delhi, India* 1-5. https://doi.org/10.1109/DELCON64804.2024.10866162.

89. Py Torch. 2020. PyTorch Mobile. https://pytorch.org/mobile/

90. Qazi S, BA Khawaja and QU Farooq. 2022. IoT-equipped and AI-enabled next generation smart agriculture: A critical review, current challenges and future trends. *Ieee Access*, **10**:21219-21235. https://doi.org/10.1109/ACCESS.2022.3152544.

91. Rajapakse V, I Karunanayake and N Ahmed. 2023. Intelligence at the extreme edge: A survey on reformable TinyML. *ACM Computing Surveys*, **55**(13s): 1-30. https://doi.org/10.1145/3583683

92. Rajeena PPF, A SU, MA Moustafa and MA Ali. 2023. Detecting plant disease in corn leaf using efficientnet architecture—an analytical approach. *Electronics*, **12**(8):1938.

93. Rana ME, VA Hameed, IKY Eng, HK Tripathy and S Mallik. 2025. Harnessing artificial intelligence for sustainable rice leaf disease classification. *Frontiers in Plant Science*, **16**: 1594329.

94. Rezaei M, S Gupta, D Diepeveen, H Laga H, MG Jones and F Sohel. 2024. Barley disease recognition using deep neural networks. *European Journal of Agronomy*, **161**:127359. https://doi.org/10.1016/j.eja.2024.127359

95. Richey B, S Majumder, M Shirvaikar and N Kehtarnavaz. 2020. Real-time detection of maize crop disease via a deep learning-based app. *Real-time image processing and deep learning*, **11401**: 23-29. https://doi.org/10.1117/12.2557317

96. Sahin Ö. 2020. Introduction to Apple ML tools. In Develop Intelligent iOS Apps with Swift. Apress, Berkeley, CA. 17-39. https://doi.org/10.1007/978-1-4842-6421-8_2

97. Saleem MH, S Khanchi, J Potgieter and KM Arif. 2020. Image-based plant disease identification by deep learning meta-architectures. *Plants*, **9**(11): 1451. https://doi.org/10.3390/plants9111451

98. Santhi S, R Dhivyasri, KS Kumar and D Srimathi. 2025. Edge AI-Driven Corn Leaf Disease Detection for Sustainable Agriculture. *In 2025 Third International Conference on Emerging Applications of Material Science and Technology (ICEAMST),* 1710-1716. IEEE.

99. Santhosh BJ and C. Balarengadurai. 2024. Edge-AI based Plant Leaf Disease Identification and Prevention for Smart Agriculture. *Grenze International Journal of Engineering & Technology (GIJET)*, **10**:6556

100. Savary S, L Willocquet, SJ Pethybridge, P Esker, N McRoberts and A Nelson. 2019. The global burden of pathogens and pests on major food crops. *Nature Ecology & Evolution*, **3**(3): 430-439. https://doi.org/10.1038/s41559-018-0793-y

101. Shafi O, C Rai, R Sen and G Ananthanarayanan. 2021. Demystifying tensor rt: Characterizing neural network inference engine on nvidia edge devices. *IEEE International Symposium on Workload Characterization (IISWC)*, 226-237. https://doi.org/10.1109/IISWC53511.2021.00030

102. Shankar V. 2024. Edge AI: a comprehensive survey of technologies, applications, and challenges. *Proceedings – International Conference on Advanced Computer Information Technologies, ACIT*, 1-6. https://doi.org/10.1109/ACET61898.2024.10730112

103. Singh Y, S Shukla, N Mohan , SE Parameswaran and G Trivedi . 2023. Real-time plant disease detection: a comparative study. *In International Conference on Agriculture-Centric Computation*, 210-224. https://doi.org/10.1007/978-3-031-43605-5_16.

104. Soni D, R Gaiiar and N Gajjar. 2023. Real-Time Castor Leaf Disease Detection Using Machine Learning on Edge AI Device. *IEEE-iSES*, 365-368. IEEE. https://doi.org/10.1109/iSES58672.2023.00080

105. Soro A. 2021. TinyML for Ubiquitous Edge AI. *arXiv*:2102.01255 https://doi.org/10.48550/arXiv.2102.01255.

106. Tariq MU, SM Saqib, T Mazhar, MA Khan, T Shahzad and H Hamam. 2025. Edge-Enabled Smart Agriculture Framework: Integrating IoT, Lightweight Deep Learning, and Agentic AI for Context-Aware Farming. *Results Engineering*, **28**:107342. https://doi.org/10.1016/j.rineng.2025.107342

107. Tensorflow for Mobile and Electron Devices. 2020. [Online]. Available: https://www.tensorflow.org/lite

108. TensorFlow Lite for Microcontrollers, https://www.tensorflow.org/lite/microcontrollers

109. Tiwari V, RC Joshi and MK Dutta. 2021. Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. *Ecological Informatics*, **63**, 101289. https://doi.org/10.1016/j.ecoinf.2021.101289

110. Vijayakumar S, SP Kane, S Vaiappuri, F Louis, and P Prabha. 2025. Quantum-enhanced hybrid CNN-QNN framework for intelligent rice disease diagnosis and real-time field recommendations on edge AI devices. *IEEE International Symposium on Computing and Informatics.* https://doi.org/10.1109/isci65687.2025.11167361

111. Wan A, X Dai, P Zhang, Z He, Y Tian, S Xie, B Wu, M Yu, T Xu, K Chen and P Vajda. 2020. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*:12965-12974.https://doi.org/10.1109/CVPR42600.2020.01298

112. Wang S, D Xu, H Liang, Y Bai, X Li, J Zhou, C Su and W Wei. 2025. Advances in deep learning applications for plant disease and pest detection: A review. *Remote Sensing*, **17**(4):698. https://doi.org/10.3390/rs17040698

113. Wu Z, H Zhao, Z Chen, Y Suo, S Joseph, X Yuan and W Liu. 2025. FHBDSR-Net: automated measurement of diseased spikelet rate of Fusarium Head Blight on wheat spikes. *Abiotech*, **6**(4): 726-743.

114. Xan CJ, H Nugroho, S Eswaran and TF Siang. 2024. Effective Edge Solution for Early Detection of Rice Disease on ARM-M

Microcontroller. *IEEE Access*, https://doi.org/10.1109/ACCESS.2024.3470970

115. Xie X, Y Ma, B Liu, J He, S Li and H Wang. 2020. A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Frontiers in Plant Science*, **11**:751. https://doi.org/10.3389/fpls.2020.00751

116. Yang C, X Sun, J Wang, H Lv, P Dong, L Xi and L Shi. 2024. YOLOv8s-CGF: A lightweight model for wheat ear Fusarium head blight detection. *PeerJ Computer Science*, **10**: e1948.

117. Yousif AJ, FK Zaidan and NJ Ibrahim. 2025. A Portable AI-Driven Edge Solution for Automated Plant Disease Detection. *Diyala Journal of Engineering Sciences*, 124:135. https://doi.org/10.24237/djes.2025.18308

118. Zeeshan M, MS Baghini and A Pandey. 2025. EdgePlantNet: Lightweight edge-aware cyber–physical system for plant disease detection using enhanced attention CNNs. *Pervasive and Mobile Computing*, **110** (102059). https://doi.org/10.1016/j.pmcj.2025.102059.

119. Zhang J, D Shen, D Chen, D Ming, D Ren and Z Diao. 2024. ISMSFuse: Multi-modal fusing recognition algorithm for rice bacterial blight disease adaptable in edge computing scenarios. *Computers and Electronics in Agriculture*, **223**: 109089.

120. Zhang X, Z Cao and W Dong. 2020. Overview of edge computing in the agricultural internet of things: Key technologies, applications, challenges. *Ieee Access*, **8**:141748-141761. https://doi.org/10.1109/ACCESS.2020.3013005

121. Zhang Y, J Yu, Y Chen, W Yang, W Zhang and Y He. 2022. Real-time strawberry detection using deep neural networks on embedded system (rtsd-net): An edge AI application. *Computers and Electronics in Agriculture*, **192**:106586. https://doi.org/10.1016/j.compag.2021.106586

122. Zheng J, Z Lv, D Li, C Lu, Z Shen, H Chen and J Zhang. 2025. High-efficiency real-time rice leaf disease classification using convolutional neural network accelerator on FPGA for edge computing in precision agriculture. *Measurement*, 118122.