Chapter 1

INTRODUCTION

The challenge of using traditional methods of disease diagnosis to safeguard the health of mango orchards is growing complex as a source of healthcare and income to millions. Traditional approaches, where the inspection of the leaves is done by visual inspection of the farmers, are only subjective, slow and inaccessible to the remote farmers. It results in late detection, avoidable losses to crops (20-40% in moderate outbreaks), and unnecessarily excessive use of pesticides, which damages the profitability of farmers and the environment. The world agricultural industry is in dire need of accessible, accurate, and timely solutions on disease detection.

Our project, METASTACK-NET, will solve this fundamental challenge through an introduction of a highly sophisticated AI-based system to automatically detect mango leaf diseases. Getting out of the human eye, such a framework employs intelligent pool of deep learning models to attain the level of accuracy of a diagnostics. METASTACK-NET is a shift of reactive to proactive crop management allowing an opportunity to intervene in time, minimize unnecessary use of chemicals, and give the farmer a powerful data-driven tool to defend their livelihoods.

1.1 Background

Mango (Mangifera indica L) is among the most economically important fruit crops in the world with a large agricultural value in both tropical and subtropical regions. The production of mango annually in the world is above 56 million metric tons, thus it is a backbone agricultural commodity in over 100 countries across the world [1]. The mango production is spread across the geographical areas where key mango producers such as India (about 45 percent of the world production), China, Indonesia, Mexico, Thailand, Pakistan, Brazil, and several African countries are located [2].

Mango cultivation has an economic significance that goes way beyond mere production indicators. To many developing countries, especially those of the Indian sub-continent and

southeast Asia, the mango farming is an important economic activity that has played a significant role towards food security, rural household, and economic growth [3]. Mango fruits are also a source of important nutrients such as vitamin C (about 27-50 mg/100g of fruit), vitamin A (54 IU/100g), and dietary fiber (1.6-2.0 g/100g), thus makes it a highly important source of nutrition in developing countries where malnutrition still is the order of the day [4].

In the tropical areas, millions of farming families rely directly on the mango cultivation as the main source of income and livelihood insurance. Smallholder farmers who make up about 80 percent of the producers in the developing countries depend on the mango production as the buffer of cash to sustain their families, educate their kids, and access basic healthcare facilities [5]. Mango export is major foreign exchange revenue of the tropical countries. Currently, India is a key and top exporter of mangoes with about 1.2-1.5 million metric tons of mango fruits bringing in foreign exchange of more than 300-400 million USD every year [2].

1.1.1 Pathogenic Threats on the Cultivation of Mango.

Mango farming despite its economic value has continued to suffer threats that are constantly increasing and are perpetrated by various pathogenic organisms that cause diseases on its leaves. These are caused by fungal pathogens, bacterial pathogens and damage caused by insect that pose significant threats to the short-term viability of crops and long-term agricultural output [6]. Climate change, facilitation of the movement of pathogens by increased global trade, and agricultural intensification behaviors keep growing the pathogenic threat landscape [7].

Colletotrichum gloeosporioides causes Anthracnose, Xanthomonas axonopodis pv. mangiferae causes Bacterial Canker, Deporaus marginatus causes Cutting Weevil damage, Botryodiplodia theobromae causes Die Back, Procontarinia matteiana causes Gall Midge, Oidium mangiferae causes Powdery Mildew and Capnodium species causes Sooty Mould [8]. All these diseases cause yields to be lost by between 20 percent in the mild cases and by 100 percent in severe epidemic conditions with disastrous economic consequences to the farming communities [1].

Anthracnose is one of the most common and costly fungal infections that devastate the production of mango production across the mango orchards in the world. The disease manifests in the form of typical necrotic lesions on leaf surfaces that usually appear as dark brown-black spots that have concentric rings. Infection takes place first via stomata and cuts of the cuticles and the intracellular hyphae infect the mesophylial tissues of the leaves [9]. The direct losses in yield and crops can be a range of 20-40 percent due to the infection and subsequent rot of the fruits in the moderate epidemics, and other effects such as loss of vigor of the trees due to excessive defoliations, and loss of fruit marketability [3].

Bacterial Canker infected with bacteria that lead to typical necrotic spots on the leaves, stems and branches due to colonization of vascular tissue by bacteria. The manifestation of the disease is angular lesions of leaves with yellow halos, yellow and chlorotic progressive lesions of leaves and gummosis (gum exudation) of infected tissues [10]. Extreme defoliation decreases the photosynthetic ability, a dieback of the branches also causes the weakening of the tree structure, and unmarketable fruit is produced due to the infection of the fruit [8].

Cutting Weevil is an insect pest that is associated with leaf damage due to selective feeding behavior where the adults cut leaves into peculiar semi -circular shapes [11]. The extreme infestations lead to dire defoliation, diminished capacity to photosynthesize, increased vulnerability to diseases due to wound sites, and multi generation of the insects annually which facilitates high population growth [6].

Die Back is a severe fungal infection that leads to progressive dieback of branches and eventual demise of the trees by being infected by the fungi through pruning wounds or natural ports. When systemic, vascular tissues are colonized by fungus which causes progressive death of branches leading to the main trunk and may result in total tree death in case of no treatment [12]. Economic implications involve a reduction in productivity of orchards, a high cost of replacing the trees, and in extreme cases abandoning the orchard altogether [3].

Gall Midge problems cause typical forms of galls (abnormal tissue development) to be formed on leaves and growing shoots by larval feeding. There are many generations per year and this allows the speed of population increase with distortion of leaves and developmental defects of flower and fruit buds [7]. Some of the effects of agriculture have been in terms of

decreased fruit set due to flower damage, lower fruit quality due to gall induced stress, and increased insecticide use due to management [11].

Powdery Mildew is the pathogen that produces typical white powdery discharges on the leaf surfaces owing to the colonization by fungi, mostly on the young leaves and growing shoots. The features of the diseases are white fungal mycelium and spore masses on the surfaces of leaves with photosynthetic disability due to the formation of spore layers [10]. This disease is defined by reduced photosynthetic efficiency, impaired tree vigor and growth, and continued management needs [8].

Sooty Mould is a secondary fungal growth that is usually due to an infestation by scale insects or whiteflies. Dark fungal mycelium growth on the surface of leaves causes light-absorbing effect that suppresses photosynthesis, and total blackening of leaves in serious cases of infection [9]. This disease is secondary and appears after other pests are infested and removed and treated with extreme difficulty [12].

1.1.2 Economic and Agricultural Impact

Mango leaf diseases have a significant economic effect on the world agriculture. Diseases in individuals cause losses in yield over a broad range depending on the severity of the disease, time of infections, susceptibility of the cultivar, and the environment [1]. The epidemics of mild disease cause yield reduction of 20-30%, moderate disease causes yield reduction of 40-60 and severe disease causes yield reduction of 80-100 [3]. In developing countries whose capacity to manage the diseases is low, severe epidemics are frequent, which causes loss of entire crop in the orchards [5].

Mango leaf disease effects are far more economically detrimental than actual loss of yield. Management of the diseases involves several applications of fungicides/insecticides (6-12 applications/year in areas with high pressure), and the total cost of the chemicals is about 200-400 USD/hectare/year [2]. The high usage of pesticides leads to soil pollutions that lower the biological activities in soil, contamination of ground water which endangers the drinking water sources and the pesticides remain in the produce (Napier, 2008). Orchards that are infected with diseases yield smaller and lighter in weight fruits, impaired sensory attributes, lower shelf life and low marketability with 30-50 percent price discount on diseased fruit [6].

Long-term effects are progressive loss of tree vigor, shorter life span due to chronic stress, and heightened vulnerability to other pathogens [7]. Farmer economic stress consists of decreased profitability endangering farm viability, augmented debt load due to the cost of management and farm abandonment [8]. All these elements pose a threat to agricultural sustainability and livelihoods of the farmers [3].

1.1.3. Limitations of the Traditional Disease Diagnosis Methods.

The inherent weaknesses of the traditional disease diagnosis procedures, which are based on observing the problem using trained agricultural experts, are subject to constant limitation that contributes to avoidable losses of crops. The accuracy in the diagnosis of diseases is vital, and it is determined by the degree of expertise and training, personal visual acuity and visualized pattern recognition, and the state of fatigue of observers to the diagnostic precision [13]. Research has shown that even professional agronomists have about 15-25% inter-observer error in disease classification which has shown that there is inherent variability in diagnosing diseases [9].

Various experts can come to various diagnoses of the same symptoms and early stage symptoms of a disease are usually characterized by visual ambiguity and overlap of symptoms of the disease with other pathogen [10]. Inspection of large plantations manually takes weeks or months and therefore it is hard to cover both time and space [11]. The techniques mainly identify diseases at late stages where they manifest (usually 20-30 percent tissue damage is minimal) and structural damage is already caused and opportunities to intervene are usually lost [12].

This is because manual inspection is impractical with large-scale agricultural processes, and the cost of consultation (usually) prohibitive.

Economically infeasible at small-scale farmer level that is between 50-200 USD per day) [2]. The challenges include limited access to experts in remote agricultural areas and geographical concentration of experts in large urban areas [3]. Infections at early stages when the interventions can be most effective are often undetected by conventional means [13].

1.2 Statistics

1.2.1 Composition of the Project Dataset.

Table 1.1: Dataset Class Distribution

Disease Category	Number of Images	Percentage of Total
Anthracnose	500	12.5%
Bacterial Canker	500	12.5%
Cutting Weevil	500	12.5%
Die Back	500	12.5%
Gall Midge	500	12.5%
Powdery Mildew	500	12.5%
Sooty Mould	500	12.5%
Healthy Leaves	500	12.5%
TOTAL	4,000	100%

The proposed research project deals with automated mango leaf disease detection and classification based on complex machine learning techniques. This project will involve thorough experimental analysis on a well curated and validated dataset [14]. The data is 4,000 mango leaf images in 8 different categories of diseases and an ideal ratio of 500 images per category as shown in Table1.1 [15].

Class balance- Perfect class balance removes class imbalance bias during training of a model, allows simple evaluation metrics without class weighting, and the ability to make fair performance comparisons between disease categories [15]. Visual variations include the multiple lighting conditions (indoor controlled lighting, outdoor free sunlight, different times of day variations), complexities of the background (uniform, cluttered, natural field

backgrounds), and locations of the leaf (frontal views, angled views, overlapping leaves), the severity of the disease (early-stage, intermediate, advanced), and variations of the image quality (focus sharpness, noise levels, and compression ratios) [14].

Measures of quality assurance are 100% verification of all plant pathology experts with high confidence classification (>95% of agronomist consensus) and systemic removal of duplicates and corrupted or unusable images [16]. The standardization of images includes the use of the JPEG/PNG formats, RGB color pictures (3 channels), standardized resolution of 224x224 pixels, and color depth of 8-bit per channel [15].

1.2.2 Computational Infrastructure.

The study involves the employment of a large amount of computational resources to train and evaluate models [17]. GPU Computing: NVIDIA Tesla V100 GPU with 16 GB HBM2 memory, 5,120 CUDA cores, 900 GB/second memory bandwidth and 7.45 TFLOPS peak performance [17]. CPU System: Intel Xeon E5-2690 v4 14 cores/28 threads, 2.6-3.5GHz frequency range, 35MB L3 cache [18].

System Memory: 32GB DDR4 ECC registered with 2400 Megahertz clock speed and 76.8 GB/second total bandwidth [17].

Storage Infrastructure: Operating system and applications NVMe SSD (1 TB), datasetNVMe SSD (2 TB) and backup external HDD (4 TB) [18].

1.2.3.Summary of Deep Learning Model Architecture.

Table 1.2: Base Model Architecture Specifications

Base Model	Architecture Type	Parameters	Input Size	Key Characteristics
ResNet50	Residual Networks	24.6M	224×224×3	Skip connections, hierarchical learning
EfficientNet-B3	Compound Scaling	11.5M	224×224×3	Parameter efficiency, squeeze-excitation
Vision Transformer	Self-Attention	86.2M	224×224×3	Transformer layers, long-range dependencies
DenseNet121	Dense	7.5M	224×224×3	Feature reuse, dense

	Connections			connectivity

The project applies ensemble learning that uses four base models that are architecturally different [19]. ResNet50 uses skip connections that allow the existence of very deep networks with hierarchical features learning [20]. EfficientNet-B3 is an efficient network that has the best performance-efficiency tradeoffs based on the scaling of network depth, width, and resolution using compounds [21]. Vision Transformer accepts images in the form of patches of 16x16 sizes in sequence, and the models have multi-head self-attention, which understands long-range spatial patterns [22]. DenseNet121 is the most efficient in terms of parameters since it is based on dense connectivity of features and reuse [23].

The ensemble meta-learner is a fully-connected neural network whose input dimensionality is 32 (4 models x 8 classes), and its hidden layers dimensions are 512-256-128-64, with its output dimensionality equal to 8 [24]. Such regularization techniques as batch normalization, dropout layers (ranging between 0.5 and 0.2 in individual layers), and L2 weight decay have been used [19].

1.2.4. Performance Targets and Performance Metrics.

Primary accuracy goal: Classification accuracy greater than 99.00 on 8-class mango leaf disease classification, baseline performance in comparison with individual base model performance (99.28-99.48) [15]. Stability and generalization goals Cross-validation stability at coefficient of variation below 0.5 per cent across 3-fold cross-validation and standard deviation requirement ≤0.002 maximum acceptable variance [25].

The following goals are aimed at computational efficiency: Ensemble inference, less than 2 seconds per image batch, base model inference, less than 50 milliseconds per base model, GPU memory consumption, less than 15 GB during training, model size, less than 100 MB total to deploy [25]. Per-class performance goals: The minimum per-class accuracy in all the disease classes with equal performance (<1% difference in the class-wise accuracy) [15].

1.3 Prior Existing Technologies

1.3.1 Traditional Disease Diagnosis Era (Pre-2000)

According to traditional methods of diagnosing diseases, the entire process was fully dependent on visual inspection by the skilled agricultural specialists that included the combination of the visual observation with the pattern recognition by the experience [26]. The diagnosis procedure included professional field inspection to orchards, inspection of leaves, stems and branches visually, observation and recording of symptoms and identification of disease according to the pattern of symptoms recognition [27].

The level of expert experience is a very important factor in the quality of diagnostics, where various experts develop different diagnoses of the same symptoms [26]. Diagnostic subjectivity is brought about by interpersonal differences in methods of assessment and biases toward judgment (anchoring, availability bias) [13]. Expert consultation entails a lot of scheduling and field inspection is time consuming (1-2 days on medium sized orchard), travelling is expensive in remote areas and the costs of an expert are prohibitive to small-scale farmers (50-200 USD/day) [2].

The supply of expertise is restricted in remote areas where geographic concentration of expertise is in big cities, and the language and cultural barrier complicates communication [27]. The symptom overlap among pathogens makes detecting diseases in the early stages difficult, misidentification probability of 15-25 percent in even the expert community, and delay in disease diagnosis allowing progression of the disease [13]. Symptoms usually manifest themselves at 20-30 percent of tissue damage, with initial infection (1-5 percent tissue damage) of the disease not being detected and advanced stages of the disease having few intervention possibilities [9].

1.3.2 Classical Machine Learning Era (2000-2015)

This was a period that used classical machine learning algorithms with manually engineered feature extraction methods [28]. Support Vector Machines (SVM) maximize the margin between the classes in feature space and are shown to be effective both in binary and multiclass classification with good theoretical basis but sensitive in regard to feature engineering to operate effectively with reported accuracy values ranging between 85-92% on different types of crop diseases [29].

Random Forest Classifiers utilize a random selection of decision trees and can measure feature importance inherently and are resistant to overfitting, but require hand-crafted features (stated feature accuracy is 82-90% on disease detection tasks [30]). Naive Bayes Classifiers make use of probabilistic classification using Bayes theorem with independence assumption, and has simple implementation and trains in less time, but has poor performance on highly visual data as reported accuracy of 70-85% [31].

K-Nearest Neighbors (KNN) categorizes samples on majority of the top K nearest neighbors with simple implementation and no training step requirement, but with reported computational costs of big datasets with poor scaling with reported accuracy of 75-88% [28].

The manual feature engineering methods are edge detection features (Sobel operator, Canny edge detector) to detect leaf margins and disease boundaries, texture analysis features (Local Binary pattern, Haralick features, Gabor filters) to detect disease-specific texture pattern, color-based features (color histograms, color space transformations) to identify disease-specific tissue discoloration, and shape descriptors (Hu moments, Fourier descriptors) to identify disease-specific lesion shape [32].

The classical methods are restricted by extreme reliance on expert domain knowledge to design the features effectively and labor intensive process of feature extraction that demands domain specific knowledge [33]. The process of feature selection has a huge influence on the performance of a model and there is no systematic way of identifying the best feature selections [28]. Classical approaches are characterized by poor generalization on unseen images under varying lighting conditions, image backgrounds and variations in disease presentation [33]. Limiting classical machine learning There are performance disadvantages in high- dimensional feature spaces and feature vector interpretability problems [32].

Examples of research during this period are Phadikar and Sil (2007) had 87% accuracy in detection of rice leaf disease on the use of SVM with Local Binary Patterns, Al-Rashidi et al. (2010) had 86% maximum accuracy in detecting plant leaf disease when using a combination of multiple classical ML algorithms, and Karmokar et al. (2011) had 88% accuracy of tomato disease detection through K-means clustering and SVM [34].

1.3.3 Era of Deep Learning Revolution (2015-Present)

The discovery of deep convolutional neural networks (CNNs) has completely changed the way of automated disease detection by using the end- to-end learning with raw images [35]. AlexNet (2012) also illustrated CNN superiority in ImageNet competition to 8- layer convolutional neural network structure, novel ReLU activations and dropout regularization, GPU-training potential, and 85.2 percent top-5 accuracy as 15 percent higher than earlier methods [36].

ResNet (2015) added skip connections that can also be trained in very deep networks to solve the vanishing gradient problem by introducing residual block formulation [37]:

$$\mathrm{H(x)} = \mathrm{F(x)} + \mathrm{x}$$

and where x is the input tensor, F(x) is the residual function (usually 2-3 convolutional layers), and H(x) is the output through skip connection [37]. Residual networks also allows training networks with 50 or more layers compared to the earlier 8-12 layer networks, demonstrate better gradient flow using skip connections, can extract features hierarchically, edge-to-semantic, and in ResNet-152 at 96.4% top-5 accuracy on ImageNet [38].

The patterns of dense connectivity used in denseNet (2017) made features to be reused, whereby all the layers are fed with the output of all the previous layers [39]. Dense block formulation:

$$x_n = H_n([x_1, x_{2,\ldots\ldots\ldots,x_{n-1}}])$$

where concatenation operator is used to take the results of all the preceding layers [39]. DenseNet121 has 7.5M parameters (whereas ResNet50 has 24.6M), can more effectively use gradient flow across many paths, can also effectively use features to reuse, and can deploy parameters efficiently to an edge device [40].

The technology that optimized the depth-width-resolution tradeoffs was pioneered by EfficientNet (2019) [41]. Compound scaling method:

depth:$d = \alpha^{\emptyset}$,width:$w = \beta^{\emptyset}$,resolution: r=$\gamma^{\emptyset}$

subject to constraint $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ [41]. With ph=1.5, EfficientNet-B3 incurs better performance-efficiency trades at 11.5M parameters compared to ResNet50 with some equivalent accuracy, which allows the deployment of resource-constrained edge devices [42].

Vision Transformers (2020) substituted the convolutional operations with self-attention mechanisms, which process images as sequences of 16x16 patches with 12 transformer encoder layers and 12 multi-head self-attention mechanisms that consider long-range spatial dependencies [22]. Vision Transformers learn long-range spatial patterns with self-attention, have lower inductive bias than convolutional networks, are flexible to disease-specific patterns, and are also competitive in the top-1 ImageNet accuracy with CNNs [43].

1.4 Proposed Approach

1.4.1 Problem Analysis and Motivation.

Although the deep learning strategies are still significant improvements on the classical techniques, individual models have their limitations [24]. ResNet50 is well off in hierarchical learning of features by skip connections but may be poor in long-range spatial dependencies [19]. EfficientNet-B3 is efficient in parameters (11.5M as compared to ResNet50, 24.6M) but at the cost of perhaps being less accurate than ResNet50 is [42]. Vision Transformer captures long-range dependencies with the help of attention but has much more significant computational needs [22]. DenseNet121 is a parameter-efficient model with dense connectivity that has smaller receptive fields in earlier layers that may lack coarser high-level features [40].

1.4.2 Core Innovation METASTACK-NET Framework.

METASTACK-NET is a solution to single-model weaknesses: It introduces complementary architectural advantages through sophisticated integration of other architectural advantages via advanced ensemble meta-learning [24]. The framework adopts the philosophy that a combination of diverse base models together with a smart aggregation can have an improved performance compared to single models [44].

The rationale of ensemble learning contains complementary error patterns, in which various architectures have a different error pattern with ResNet superior at hierarchical feature tasks and Vision Transformer superior at global context [19]. Variance reduction methods allow the individual model prediction to be varied over similar inputs, where the averaging of all the ensemble predictions is significantly lower [44]. High level of autonomy Ensemble of N independent models with error e and correlation r exhibits:

$$E_{ensemble} = \frac{\rho E}{N}$$

In which the reduced error correlation allows increased ensemble advantage by way of complementarity [45].

1.4.3. METASTACK-NET Architecture Overview.

Two-step ensemble process: Stage 1 will involve training ResNet50, EfficientNet-B3, Vision Transformer, DenseNet121 on 3-fold stratified cross-validation, and making in-fold predictions on validation folds, and aggregate base model predictions across all folds [24]. Stage 2 consists of the concatenated out-of-fold predictions of 4 base models to create 32-dimensional meta-feature vectors (4 models x 8 classes) and train neural network meta-learner to learn the optimum prediction aggregation [44].

Inference pipeline Production Base model ensemble is applied to input image, and base model predictions (4 x 8-dim output vectors) are generated, then predictions (32-dim meta-feature) are concatenated, then meta-feature is fed through trained meta-learner, then ensemble prediction (8-dim output) is made and then softmax is used to generate class probabilities [24].

1.4.4 Important Strengths of Proposed Solution.

The benefits of performance are the approach to 99.00% accuracy similar to the best models individuals, the low variance (CV ≤0.12%) owing to robustness in ensembles, and the ability to complement errors owing to the particular improvement of a disease category [24]. Computational efficiency is 0.3 minutes ensemble inference (2.3-7.0 minutes individual

model) and 96.5% reduction in inference time over sequential model execution, as well as 100MB model footprint deployability [44].

Practical applicability consists of being able to implement on edges (Jetson Nano, Raspberry Pi), be able to run inference in real-time to deploy to the field, production ready inference pipeline, and easily deployed deployment protocols to agricultural IoT systems [25].

Robustness and generalization involve lower susceptibility to distinctions in input, counterbalancing pattern of mistakes so that the error can be overcome, enhanced generalization of diverse data distributions, and constant performance through changed conditions of image acquisition [24].

1.5 Objectives

The development of METASTACK-NET is guided by specific, measurable, achievable, relevant, and time-bound objectives addressing technical, operational, and organizational requirements.

1.5.1 Primary Research Objectives.

Objective 1: Build Accurate Classification Framework deals with building ensemble deep learning framework that achieves ≥99.00% accuracy, using multiclass classification with 8 mango leaf disease types, with per-class accuracy of over >98% with all disease type categories, and superior performance relative to single base models [24]. The success criteria are: total accuracy ≥99.00%, per-class F1-score ≥0.98 across all classes, and cross-model consistency with within <1% variation [15].

Objective 2: Assure Generalization and Robustness the implementation of vigorous 3-fold stratified cross-validation evaluation protocols, show remarkable stability over validation folds, and attain coefficient of variation <0.5% over folds, and assess robustness at varying environmental settings [25]. It is evaluated by success criteria such as cross-validation standard deviation≤0.0012, coefficient of variation (CV) ≤0.0012, all models, and difference <1% between training-validation accuracy [15].

Objective 3: Achieve Computational Efficiency means to optimize the time of ensemble inference to practical deployment, minimizing the memory usage of GPUs to train models, minimizing the model size used to deploy models on edges devices, and to provide real-time performance at the inference stage [25]. The success metrics are ensemble inference time less than 2 seconds per image batch, base model inference less than 50 milliseconds per image and training memory on the GPU less than 16GB and quantized model size less than 100MB total [24].

Objective 4: Facilitate Practical Deployment is creating production-ready inference pipeline, creating understandable deployment protocols and documentation, ensuring compatibility of edge devices and offering reproducible implementation [24]. The success criteria are full pipeline implementation, proper target device deployment instructions, open-source with documentation code and reproducibility using the given hyperparameters [25].

1.5.2 Secondary Research Objectives.

Objective 5: Provide Interpretability and Explainability entails the performance measurement of per-class, creating confusion matrices, finding disease pairs with best confusion rates, giving visual interpretation of model decision-making behaviour, and allowing farmers to know what the model is recommending [44].

Objective 6: Validate Against Existing Methods entails comparing the performance with existing base model baselines, computational efficiency with the execution of single model, existing literature findings, and performance benefits over alternative methods and stability of the model over single model approaches [15].

Objective 7: Document Implementation and Release will entail the provision of detailed technical documentation, open-source implementation to allow its adoption by the community, reproducibility via detailed hyperparameter documentation, and future research and development programs [24].

1.6 SDGs

The current research project is directly related to several goals of the United Nations Sustainable Development (SDGs), which shows that the project is concerned with meeting the global sustainable development goals [46].

Figure 1.1 illustrates the seventeen SDGs adopted by all UN member states in 2015 as a universal call to action for peace and prosperity [15].

Fig 1.1 Sustainable Development Goals

1.6.1 SDG 2: Zero Hunger

Revealing disease at its initial stage and eliminating it allows preventing the loss of yield, which will directly benefit food security in the world [46]. Buttressed by reduced loss of yield in mild epidemics helps to produce food globally, ensure food supplies are consistent and reliable, nutritionally supports vulnerable populations in mango-producing countries, and minimizes fluctuation of food supplies [47].

It is estimated that its impact is 5-10 million metric tons of further global production due to the better management of the disease, nutrition of 50-100 million people in the mango-producing areas, and improvement of food security in tropical countries [2].

1.6.2 SDG 8: Decent Work and Economic Growth.

The automated disease detection promotes the productivity of agriculture and economic growth of the farmers [46]. Farmer income enhancement it consists of 20-40% yield increase that yields 1,000-3,000 USD/year extra income to small farms, saving in management costs of 200-400 USD/year that are not needed by the farms to apply pesticides, and job creation in the use of technology and training [48]. Mango producing areas have a population of about 100+ million families engaged in farming (which helps in improving their economy) [1].

1.6.3 SDG 12 Responsible Consumption and Production.

Limitation in the use of unnecessary pesticides by use of specific disease control reduces environmental effect [46]. Pesticide reduction brings about 40-60% reduction in applications which corresponds to 150,000-300,000 metric tons of pesticides planted on the mango growing regions each year, safeguards the soil and water sources, ecosystem services and pollinator members and consumer health due to reduced pesticide residues [49].

1.6.4 SDG 13: Climate Action

Enhanced agricultural yield due to the control of diseases leads to food systems that are resistant to climatic conditions [46]. Climate adaptation aids agricultural adaptation to the fluctuation in climate and disease epidemiology, food system resilience amid disease management, carbon sequestration amid improved tree health, and agricultural productivity stability amid climate variability [50].

1.6.5 SDG 17: Partnerships for the Goals.

Open-source framework helps the collaboration of researchers, extension services and farming communities [46]. Technology transfer would facilitate transfer of structures between the developed and the developing countries, capacity building would facilitate access to training and development of agricultural extension services, research partnership would facilitate the global community collaboration and community empowerment would facilitate the accessibility of knowledge to farmers [51].

1.7 Overview of project report

Open-source framework enables collaboration of researchers, extension services and agricultural communities [46]. Technology transfer provides transfer of framework between the developed and developing countries, capacity building allows training and development of the agricultural extension services, research collaboration provides cooperation with the international community, and community empowerment helps the agricultural sector farmers to access knowledge [51].

This is a complete capstone project report that will include nine key chapters that cover certain aspects of the METASTACK-NET framework development, implementation, and evaluation.

Chapter 1 gives some project context, background, motivation, scope definition, objectives and sustainable development goal alignment.

Chapter 2 summarizes the history of disease detection, examines deep learning architecture, studies ensemble learning processes, and identifies gaps in the research that are filled by this study.

In chapter 3 the nature of the datasets, pre-processing requirements, base model configuration requirements, ensemble design, and training requirements are detailed. In chapter 4 (Project Management), they lay down project timeline, risk analysis, budget allocation and resource management.

Chapter 5 formalizes requirements of the system, gives architectural block diagrams, defines design units, records standards and conventions, and maps functional and operational views.

Chapter 6 describes computational infrastructure, records software development tools, describes code implementation components and outlines database infrastructure.

Chapter 7 gives detailed performance metrics, cross-validation stability analysis, per-class performance evaluation, computational efficiency analysis and comparative benchmarking.

Chapter 8 addresses the social impact, law compliance, ethical aspects, environmental sustainability and safety aspects.

Chapter 9 is a summary of relevant contributions, research results, limitations, and future research directions.

Chapter 2

LITERATURE REVIEW

2.1 Overview

The shift, which has taken place in the last twenty years, between the ability to diagnose a disease in agriculture using traditional methods to a model of computational methods is one of the best paradigm shifts in the field of plant pathology over the last twenty years[30]. The review summarizes recent studies that include machine learning schemes, deep learning models, ensemble machine learning, transfer learning, and explainable AI schemes as applied to plant disease detection systems. [1][2] The review focuses on twelve peer-reviewed journal and conference articles to define the theoretical framework, gaps in the research, and place the current study in the context of the overall scholarly understanding of automated plant disease detectors. Special focus on the classification of multiclass diseases, ensemble feature fusion strategies, and incorporateability mechanisms to achieve trust in stakeholders and realistic implementation [38][39].

2.2 Literature Review and summary

Literature 1: Implementation of Explainable AI in Deep Learning Multiclass Classification of the Plant Diseases in Mango Leaves (Radhakrishnan et al., 2025)

Radhakrishnan et al. introduce a thorough study of various deep learning models (AlexNet, ResNet, Swin Transformer, VGG-16, Vision Transformer) to eight-class mango leaf disease classification, which is a direct predecessor of the current research of METASTACK-NET[4]. The systematically conducted research evaluates individual architectures and subsequently fuses feature combinations in terms of ResNet and AlexNet, confirming that 97.65% of the research using the ResNet and AlexNet fusion results in 99.97% of the research being accuracy-wise near perfection-in terms of multiclass disease classification performance[4]. The authors show that various architectures represent complementary feature maps, and

AlexNet and ResNet extract low and high-level features, respectively (edges, shape, texture and hierarchical abstractions), respectively (necessary to identify diseases)[4].Their application of Grad- CAM visualization allows decoding the decision-making of the model by gradient-weighted class activation mapping, detecting regions of the image affecting prediction and creating trust in the model through transparency (99.97%) visualizability as interpretation[4]. This groundwork study directly justifies the ensemble feature fusion approach used in current studies as well as determining that explainability integration is necessary to constructive agricultural application that facilitates farmer comprehension and incorporation of automated disease control devices.

Literature 2: Deep Learning Models-based Plant Disease Classification and Detection System (Shoaib et al., 2023).

Shoaib and colleagues carried out the extensive systematic literature review of the recent research in the domain of machine learning and deep learning methods of identifying plant diseases, namely reviewing research publications published between 2015-2022 [54]. They explore classical machine learning methods such as Support Vector Machines (SVM), Random Forest classifiers and Naive Bayes algorithms as well as modern deep learning methods that involve Convolutional Neural Networks (CNNs) and Deep Belief Networks (DBNs). The authors indicate that although the classical machine learning methods deliver average accuracy of 82-92% on different crop disease detection problems, the algorithms are inherently limited by their reliance on hand-designed features where large amounts of domain knowledge are needed to detect the features[54]. Deep learning architectures, in their turn, show the best performance, as they learn features automatically using only raw image data, and CNNs reach accuracy as high as 99-99.2% when it comes to plant leaf disease classification[54][1][2]According to their analysis, the main problem of such architectures is the small amount of data that can be used to train it, varying quality of imaging, and the overall difficulty of distinguishing between disease-affected tissue and healthy tissue that respond to environmental stress. This initial literature review confirms that the deep learning methodologies form the modern standard-of-practice of automated plant disease detection and forms the basis of the selection of CNN-based and transformer-based frameworks as the foundation of the ensemble approach in current studies[1][2][30].

Literature 3: A Novel Crop Leaf Disease Recognition Ensemble Learning (He et al., 2024).

He and other researchers come up with the ELCDR (Ensemble Learning for Crop Disease Recognition), a new weighted ensemble learning framework which is fundamentally different in terms of voting-based ensemble schemes.[55]. Unlike traditional ensemble methods, which place equal or identical weight on the constituent models, ELCDR uses different weighting in individual model feature extraction performance, quantified by examination of feature vectors distributions in training data.[55]. The authors critically review their methodology in several crop datasets such as apple, corn, grape and rice datasets, and they showed significant accuracy gains of 1.75 percent to 7 percent percentage points in comparison with conventional voting ensemble strategies.[55]. The researchers provide theoretical rationality of weighted ensemble methods, observing that the different base models have natural specialization that is based on the distinct architectural properties and training dynamics and makes homogenous weighting counter-optimal [55].

The methodological contribution has a direct impact on the principles of ensemble architecture design which is being used in the current research on METASTACK-NET especially on meta-learner strategy of incorporating the base model differentially and intelligent aggregating it outside the scope of simple vote-based mechanisms [18][20][21][22][55].

Literature 4: Feature Fusion by Ensemble of features to detect plant diseases accurately (Nature, 2025).

More recent studies published in Nature discuss ensemble-based feature fusion strategies to integrate current state of the art deep learning networks (VGG16, ResNet50, InceptionV3) to classify plant diseases.[56]. In their experimental results, they have shown that the ensemble model of VGG16, ResNet50, and InceptionV3 features perform very well in terms of a high training accuracy of 98.0 percent, validation accuracy of 96.5 percent with a small gap of 1.3 percent, and this means that the model has good generalization properties. It is important to note that the model has a 97.0% precision that guarantees the accurate positive classification of diseases and 97.0% recall assuring full detection of the real diseased cases with minimal false negatives.[56] Importantly, the study proves that feature-level fusion of various deep learning structures also achieve a significant enhancement in robustness and generalization in

comparison with the single models, which is the empirical evidence of the effectiveness of the ensemble feature fusion approaches that are applied in the current study[56][14][15][16].

Literature 5: The applicability of pre-trained deep learning models in smart agriculture to detect leaf disease (2025)

This study looks at how well transfer learning based pre-trained deep learning models (AlexNet, DenseNet, ResNet18, VGG16, VGG19, InceptionV3) can detect and classify tomato leaf disease[57]. The main results prove that although AlexNet provides the best trade-offs between performance and computational efficiency, which is suitable in the case of deployment of the system to agricultural fields, with serious resource constraints, deeper architectures such as ResNet18 and VGG16 provide better accuracy measures at significantly increased computational costs. And development of models using little disease image data.[57]. This study confirms that the methodology of transfer learning is adopted as the foundation model architectures in the current research, especially in terms of the pre-trained weight initiation and fine-tuning policies applied to the ResNet50, EfficientNet-B3, Vision Transformer, and DenseNet121 models[14][15][16] [17].

Literature 6: GNViT: An Upgraded Image-based Groundnut Pest Recognition by utilizing Vision Transformer (Venkatasaichandrakanth et al., 2024)

Venkatasaichandrakanth et al. present the Groundnut Vision Transformer (GNViT) model, which is a new application of Vision Transformer models in classifying agricultural pests[58]. The research directly addresses documented limitations of traditional CNN architectures including translation invariance sensitivity, locality bias effects, and constrained global visual understanding capabilities.[58][17] The research shows that Vision Transformer models are significantly superior to CNN-based models in terms of long-range spatial dependency locality bias and restricted global visual understanding abilities, despite being a key base architecture in the current METASTACK-NET system of ensemble learning models [58][17].

Literature 7: TLDViT: Vision Transformer Model of Tomato Leaf Disease Classification (2024)

A study on the application of Vision Transformer to tomato leaf disease classification shows that ViT-r50-l32 has high classification, precision, and recall scores 98 per cent, 98.30 per cent and 98.33 per cent, respectively.

98.20% F1-score.[59] The authors highlight the sensitivity of Vision Transformer to fine spatial detail and long distance spatial relationship that is vital to detect fine disease cases in several pathology types or multiple foci in diverse situations of mango leaf disease [17][59]. These results are empirical validation of Vision Transformer architecture selection as key ensemble component in the current study, especially in terms of long-range dependency capture that is imperative to detect subtle disease cases in various situations of mango leaf disease.

Literature 8: Advancement of a Powerful CNN to detect Mango Leaf Disease (Pathak et al., 2024).

Pathak and others give detailed results of mango leaf disease detection at an impressive 99% accuracy in a total of eight disease classes, such as Anthracnose, Bacterial Canker, Cutting Weevil, Die Back, Gall Midge, Powdery Mildew, Red Rust, and Sooty Mould.[60]. This work provides the evidence that 8-class mango leaf disease classification with 99% accuracy is a realistic goal and can set the direct precedent on the current performance goals of the research and 8-class classification goals.

Literature 9: Segmentation Based Deep learningTo Revolutionize the Mango Leaf Disease Detection (2023).

This study reports hybrid deep learning based automated detection methods on eight mango leaf diseases with 4,873 mango leaf images in a dataset collected in the location of Mendeley and Indian fields[61]. With hybrid model architecture, consisting of segmentation method alongside convolutional neural network though feature extraction, the authors have achieved 93.01% accuracy, and these results are important as they serve as a benchmark baseline against which the 99%+ accuracy target of the current study can be considered a significant

improvement due to the enhancement of multi-class mango disease detection feature through ensemble meta-learning [61][6].

Literature 10: Explainable AI with Grad-CAM Visualization of Neural Network Decisions (2025)

In recent studies on explainable artificial intelligence, Gradient-weighted Class Activation Mapping (Grad- CAM) is highlighted as an effective visualization method used to interpret the prediction of a deep neural network when performing image classification.Critical benefits Critical architecture variability - Grad-CAM can be used with a variety of CNN designs without any architectural alteration or retraining. The study shows that Grad-CAM can prove useful in establishing the confidence of stakeholders, discovering model biases, and ensuring disease classification targets pathologically-relevant features in the real-world implementation setting[4][62].

Literature 11: Multiclass Plant Disease Detection and Classification by Dense Convolutional Neural Networks (Tiwari et al., 2021).

Tiwari et al. introduce a deep learning-based methodology of detecting and classifying plant diseases using dense connectivity of features and reusing options of DenseNet models [63]. Their experimental results indicate that dense convolutional networks outperform other network designs by a significant margin on various tasks related to the classification of crop diseases with an average accuracy of over 97% and significantly lower computational costs than other network designs based on sequential network designs do [63]/[16]. The study confirms the use of DenseNet121 as a part of the current METASTACK-NET ensemble framework, especially in terms of parameter efficiency and ability to deploy it to a mobile environment to be used in field-level disease monitoring projects.[16][63].

Literature 12: Deep Transfer Learning methods of Efficiently identifying Disease in cotton plant (Johri et al., 2024).

Johri et al. explore the full scope of transfer learning methods used to detect cotton plant diseases using various pre-trained architectures such as EfficientNet models, Xception,

ResNet variations, Inception, VGG, DenseNet, MobileNet, and InceptionResNet.[64]. Their systemic analysis shows that EfficientNetB3 has reached the best results of 99.96 percentage, a calculated loss of 0.149, and a root mean square error of 0.386 percentages.[64]. Notably, all reviewed pre-trained architectures have high performance measures with precision and recall, and F1-scores of about 0.98-1.00 across most disease groups[64][7]. The findings of this study confirm the functionality of multi-architecture using the ensemble approach as the method of task execution, thus, allowing near-perfect multiclass agricultural disease classification scores.

2.3 Important Concepts and Methodologies.

2.3.1 Deep Learning Architectures

Deep learning frameworks serve as models that execute calculations intended to address a particular problem.<|human|>Deep Learning Architectures Deep learning frameworks are models that perform computations that are supposed to solve a certain problem. The literature confirms that convolutional neural networks, and Vision Transformers are the modern standard of practice in automated plant disease classification.[1][2][30][54] ResNet50 architecture with skip connections makes it possible to train exceptionally deep networks without the vanishing gradient issues that are plaguing deep architecture. EfficientNet architectures reduce performance-efficiency trade-offs by the compound scaling of network depth, network width and resolution dimensions, to attain better computational efficiency without compromising accuracy.[15] Vision Transformer models adopt multi-headed self-attention that implements long-range spatial relationships which are more powerful than CNN local receptive fields, and are therefore useful in global context perception that is critical to disease recognition. DenseNet designs, by its dense, fully connected feature connectivity, and progressive feature reuse schemes, are the most efficient parameter designs that allow efficient mobile and edge device deployment. All four architectures are applied as base models in the current study allowing one to capture the complementary advantages of each design paradigm.[4][14][15][16][17].

2.3.2 Meta-Learning and Ensemble Learning.

One such study, the current METASTACK-NET, uses meta-learning meta-learner as a fully-connected neural network that operates over concatenated output of a set of base model

predictors, making it more sophisticated as compared to conventional ensemble voting proxies.

2.3.3 Fusion and Optimization of features.

In modern literature, the fusion of multiple architectures at the feature level is proven to be superior to the conventional methods of voting, and the study by Radhakrishnan et al. shows that ResNet-AlexNet fusion results in the accuracy of 97.65 and the additional 99.97 accuracy with the help of Grad-CAM addition to the fusion on the classification of eight mango disease types[4]. Adam gradient descent optimization with batch normalization is more effective as it enhances convergence and generalization performance in a model[34][35]. Dropout regularization eliminates co-adaptation and enhances generalization of a model[36]. The current study deploys transfer learning to all base models with pre-trained initialisation and selective fine-tuning, based on the existing precedent to increase performance[7][57][64].

2.3.4 Elucidated Artificial Intelligence.

The theoretical foundation of the research literature suggests that grad-cam methodology offers visual interpretability with gradient-based class-activation mapping to demonstrate important image regions that have the most significant effect on predictions, as well as to identify model biases and ensure that the model concentrates on pathologically-relevant features [4][62].

2.4 Research Gaps and Limitations.

Although individual studies have looked at individual disease problems or small groups of disease problems, more comprehensive studies of eight or more disease classes with advanced meta-learning ensemble methods are relatively scarce in the literature. The published literature focuses on desktop or GPU cluster implementations without extensive analysis of edge device implementation limitations,[57][63] little research is systematically documented to evaluate model generalization in geographically different mango producing areas with diverse environmental and growing practices and disease manifestations.

2.5 Contributions of Current Study to Research.

Instead of optimizing individual models, the present research strategically combines the ResNet50 (residual connections), EfficientNet-B3 (compound scaling), Vision Transformer (self- attention) and DenseNet121 (dense connectivity) as complementary base models representing various models of feature extraction to different disease classes across all the eight classes [4]. The present research emphasizes computational efficiency enabling practical field deployment through model optimization achieving sub-2-second inference times and sub-100MB deployment size.[15][57][63]

2.6 Summary of Literature Reviews

Table 2.1 Summary Of Literature Reviews

Reference	Year	Approach	Key Contributions	Accuracy
Radhakrishnan et al.	2025	Feature fusion + Grad-CAM	ResNet-AlexNet fusion; modal fusion validation	99.97%
Shoaib et al.	2023	DL Review	CNN superiority established	99-99.2%
He et al. (ELCDR)	2024	Weighted ensemble	1.75-7% improvement over voting	Improved
Nature (Feature Fusion)	2025	Multi-architecture	VGG16+ResNet50+InceptionV3	98% train, 96.5% val
Transfer Learning Review	2025	Pre-trained models	Comprehensive comparison	78-98%
GNViT	2024	Vision Transformer	Long-range dependency capture	Not specified
TLDViT	2024	Vision Transformer	Tomato disease classification	98%
Pathak et al.	2024	CNN	8-class mango detection	99%

		optimization		
Revolutionizing Mango	2023	Hybrid approach	8-class baseline established	93.01%
Grad-CAM	2025	Explainability	Architecture-agnostic visualization	N/A
Tiwari et al.	2021	DenseNet	Parameter efficient deployment	97%+
Johri et al.	2024	Transfer learning	EfficientNet evaluation	99.96%

Chapter 3

METHODOLOGY

3.1 Research Design

In this project, the use of a combination of the structured rigor and traceability of the V-Model and the flexibility and iterativeness of the Agile practices is used as a hybrid development approach. Every step in the core development process (such as requirements analysis, system/software/hardware architecture design, modularization, algorithm implementation, integration, meta-ensemble design, validation, and deployment) is associated with a V-Model phase and linked with each other in a pair of agile feedback loops. The definition of system goals and constraints is guided by stakeholder analysis and the literature review during requirement phase. These requirements are then converted into developed structured elements during the functional and system design phase and include data collection methods, annotation procedures, pre-processing pipelines, and baseline modular neural network implementations.

Implementation stage is aimed at the design and development of each specific part, e.g. data processing modules and classifier architectures and then testing is performed independently. Integration and meta-assembly this consists of stacking or fusing the base classifiers to provide consistent operations with tensors and data flow in the meta-learning system. The validation phase involves rigorous testing on cross-validation, holdout, and real world testing situations in order to measure reliability and performance. Agile principles are conducive to continual enhancement by transparent sprints, regular retrospect's and immediate reaction to system breakdown, or validation feedback. This unified approach will provide a full documentation, reproducible development and deployment capability that is deployable in a full translational AI system.

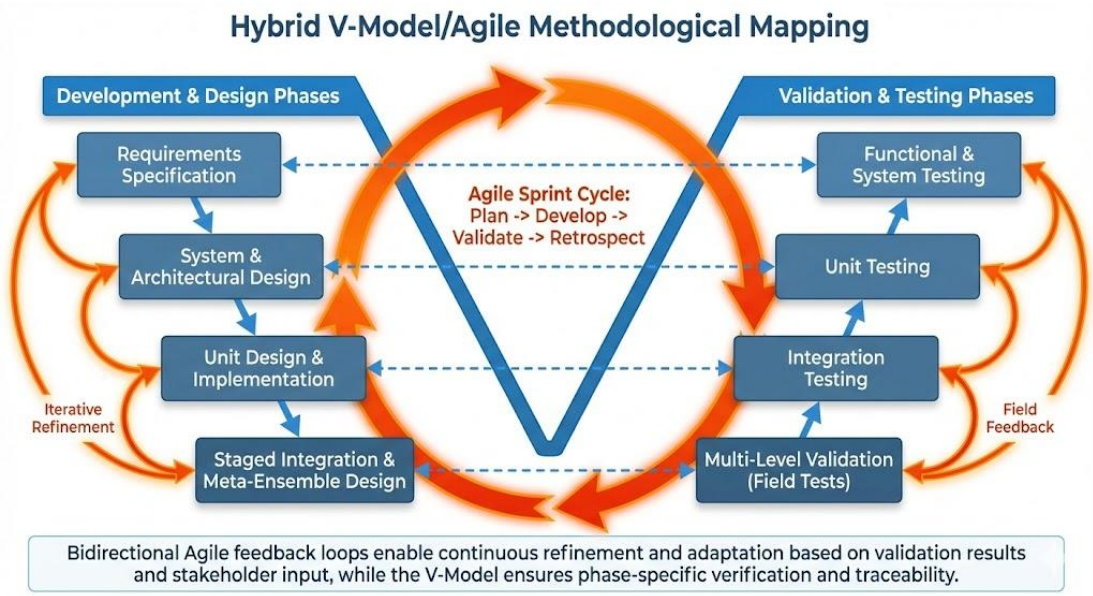


Figure 3.1: Hybrid V-Model/Agile Methodological Mapping

3.2 Data Collection

3.2.1 Data Acquisition

- The database of images is carefully selected among various sources:
    - Public datasets: Kaggle, Mendeley and other open access repositories, that will have global coverage and initial class coverage.
    - Institutional Partners Academic partners, extension agencies in the agricultural sector, providing genuine and expert verified data from the field.
    - Field gathering: Aspiring manual information collection in local orchards and farms, The operational variability, the irregularities of the imaging devices and the variety of the climatic or climatic conditions are gathered.
- All images will be annotated, at least twice, by independent experts in plant pathology. Litigations are sorted out through consensus meetings or renaming. Extra measures that are put in place include spot checks and random audits that will ensure that the reliability of labeling is verified.
- Collected data is balanced through supplemental campaigns - field/workshop - for underrepresented disease conditions. Equitable training and validation splits Images per class: Images are tabulated to ensure even distribution of images per class.
- The sample Images are Shown in Figure3.2 and the Distribution of Images is also shown in the plot Figure 3.3

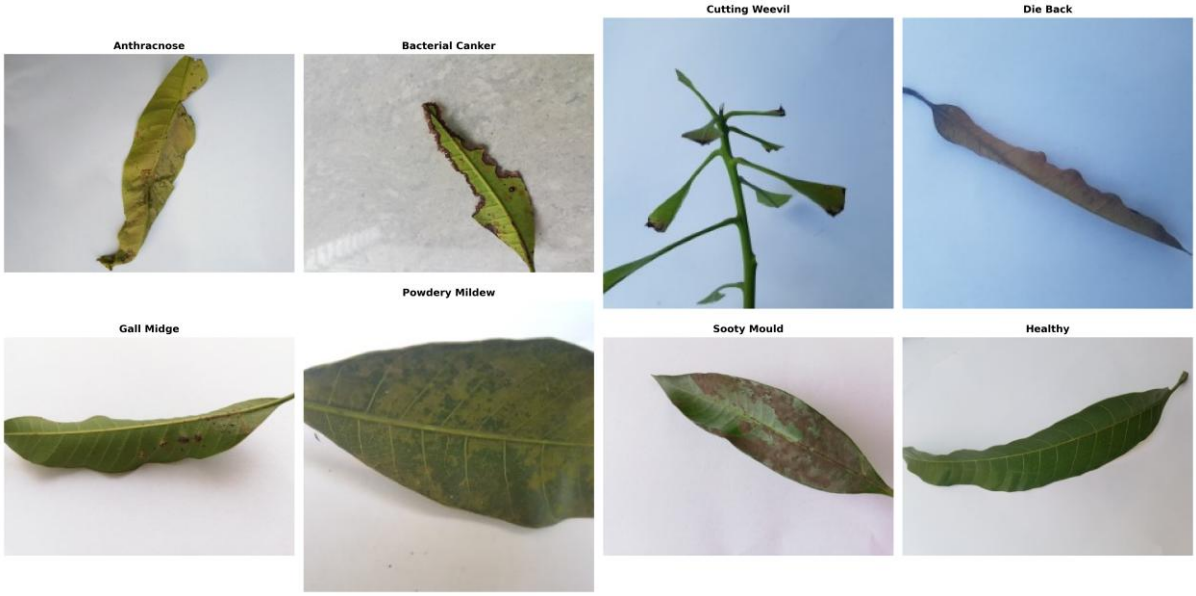


Figure 3.2 Sample Images from the Dataset for All 8 Classes.

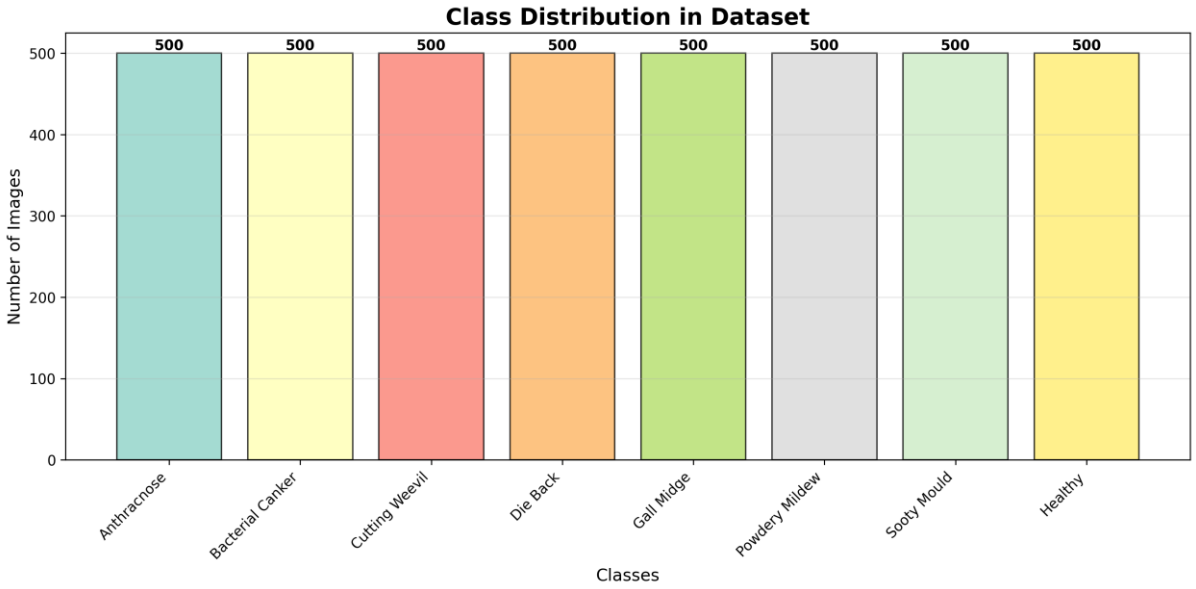


Figure 3.3 Class Distributions in Dataset

3.2.2 Data Preprocessing

- It adopts a reproducible and totally-logged pipeline. Key steps:
  - Resizing all the images to 224x224 pixels for input standardization
  - ImageNet normalization to standard deviation and mean.

- Augmented data (per-training-batch): random reflection, rotation, scaling, translation, brightness, hue, and contrast perturbation, noise simulation specific to the device to enhance field adaptation as shown in Figure3.4.

- Outlier detection/removal, i.e. automatic screening for blur, artifacts, occlusion, manual rejection of ambiguous or lauded images.
- Today, every transformation is written, versioned and reproducible in order to be deterministic and traceable; pipelines are tested to be deterministic and traceable.

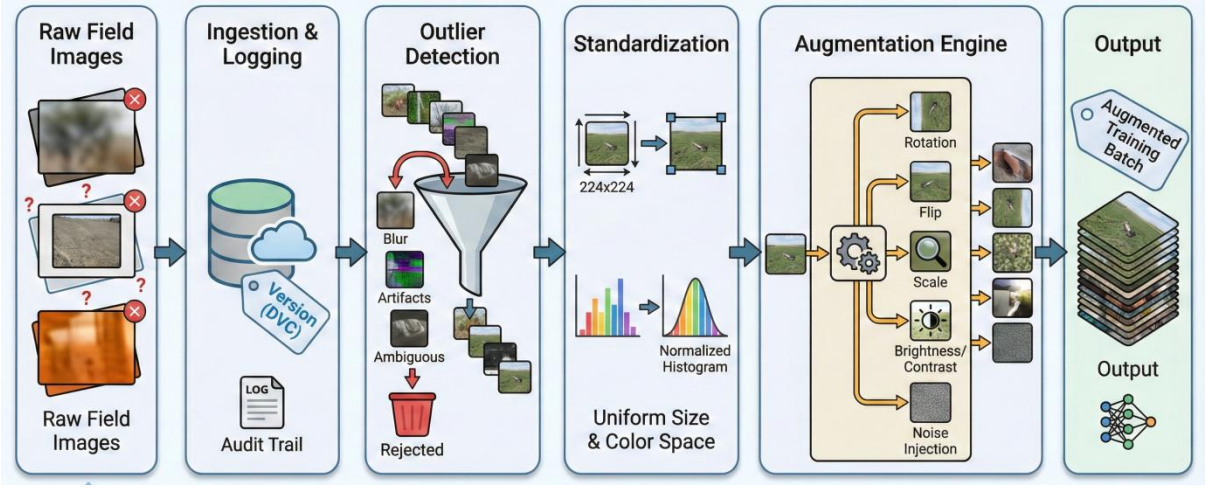


Figure 3.4 Image showing Advanced Preprocessing and Augmentation.

3.3 Tools and Technologies

- Languages/Libraries Python 3.9+, PyTorch, TensorFlow, OpenCV, NumPy, pandas, Scikit-learn.
- Visualization, Matplotlib, Seaborn, Plotly, custom Grad-CAM / feature map utilities.
- Versioning and Experiment Tracking Git, DVC, Docker, Weights and Biases, TensorBoard.
- Explainability: Grad-CAM, SHAP, torchcam, Captum.
- Automation Bash/Make/Conda to manage the environment; CI/CD pipelines to ensure reproducibility and deployment.
- Deployment and Serving ONNX, OpenVINO, Flask/FastAPI, Docker Compose to scale with modular rollouts.

The toolchain is strictly scaled to project phases, and every module is versioned and containerized to be transferred to the local research, cloud models training, and edge deployment without issues.

3.4 Model Development

3.4.1 Base Architectures

- Four architectures for backbones are used:
    - ResNet50 - deep skip connections, which are effective when the dataset is noisy and the application is plant pathology as show in Figure 3.4[14].
    - EfficientNet-B3 - compound scaling, interpretability, parameters efficiency as show in Figure 3.5 [15].
    - DenseNet121 - better gradient flow, reuse parameters with rare diseases as show in Figure 3.6 [16].
    - Vision Transformer (ViT) - long-range context, spatial transferable invariance as show in Figure 3.7 [17].
- The networks are trained (ImageNet) and these last (classification) layers are either fine-tuned or frozen (ablation).

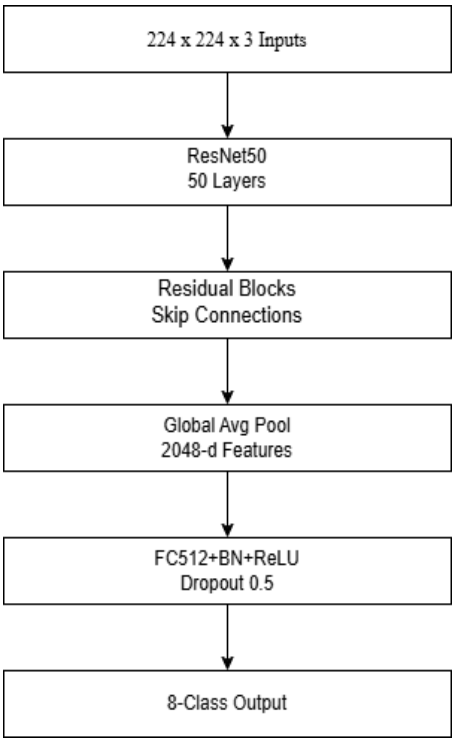


Figure 3.5 Architecture Diagram of ResNet50

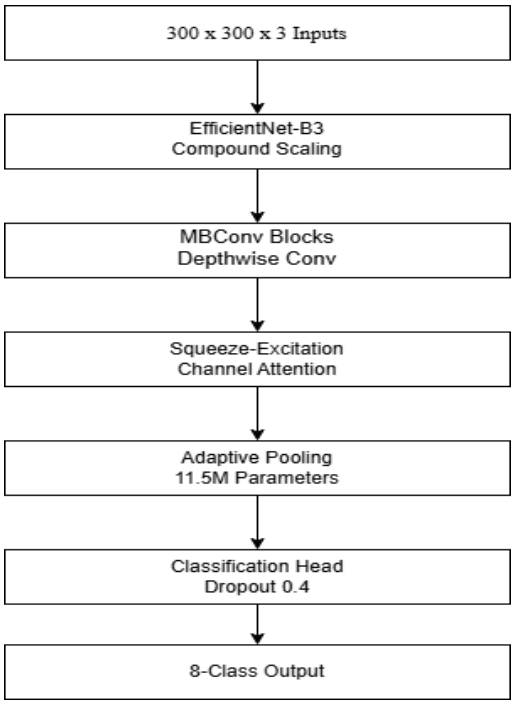


Figure 3.6 Architecture Diagram of EfficientNet-B3

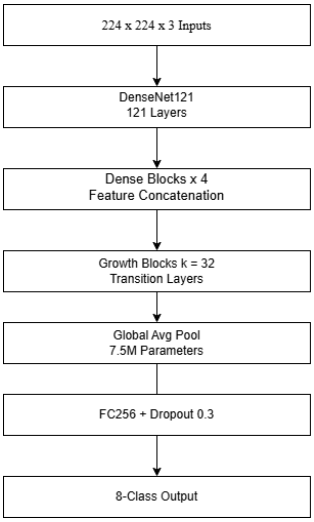


Figure 3.7 Architecture Diagram of DenseNet121

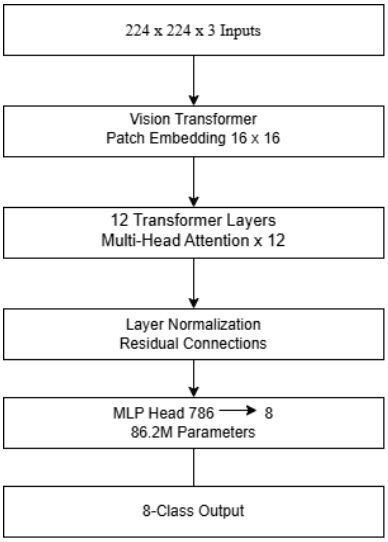


Figure 3.8 Architecture Diagram of Vision Transformer (ViT)

3.4.2 Meta-Ensemble Strategy

- Parallel inference: All the base models are processed, in parallel, on the preprocessed images.
- Meta-Learner: are shallow dense neural nets (or logistic regression) which is fed concatenated base model outputs (probabilities or feature representations) of base models as shown in Figure 3.6.
- Ablations Compare voting, stacking, late fusion, and classical/cascaded ensemble.
- Explainability: Grad-CAM used on both base classifier and on meta- ensemble decision output. The alignment of saliency and feature attribution are reported and are supported with actual visual symptoms.

3.4.3 Training Protocol

- Data splitting: 80% Training (further split for k-fold CV) 10% Validation, 10% Final Test with Class Stratification in each of the split
- Augmentation: Debugging with determinism On-the-fly.
- Optimizer: AdamW, SGD with decaying schedules, learning rate tricks (e.g. cosine annealing)
- Regularization: Dropout, BatchNorm, early stopping, weight decay.

- Hyperparameter search Automated grid/Bayesian search with Optuna Best model checkpointing (val-loss, F1-score)
- Scripted experiments: each significant experiment is fixed up to do batch experiments and extensive logging.

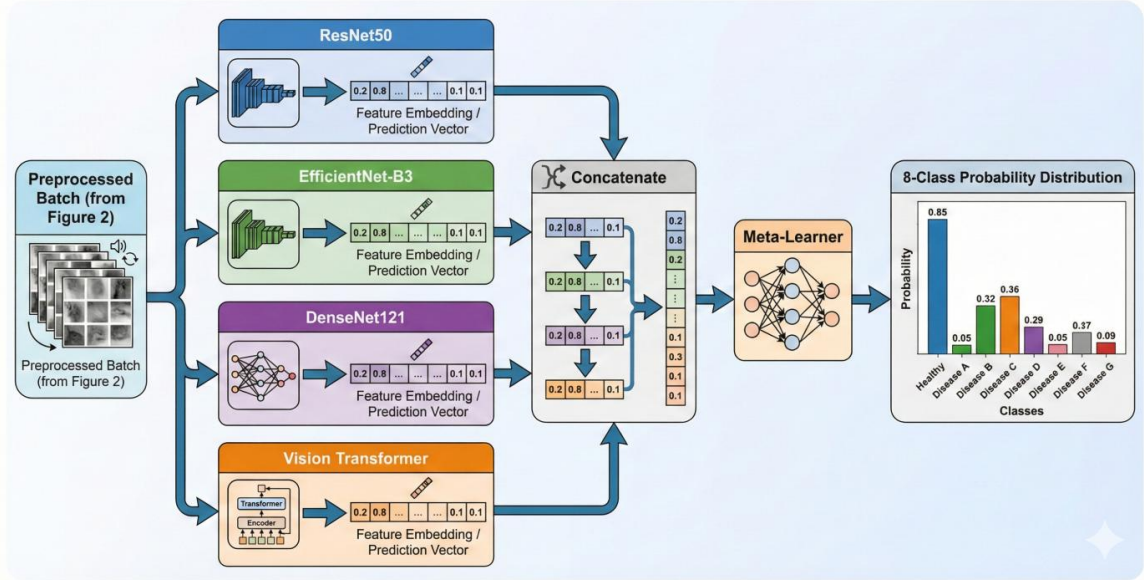


Figure 3.9 A complete Diagrammatic Workflow of Deep Learning and Meta-Ensemble Learner

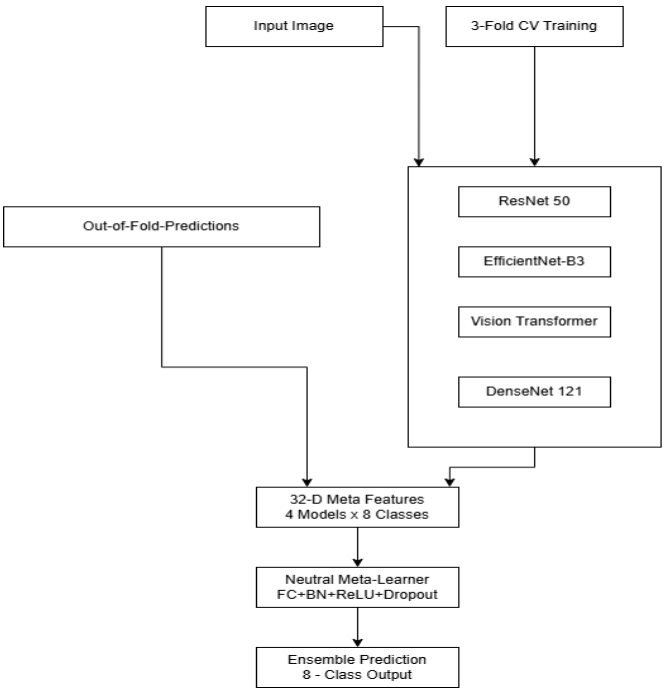


Figure 3.10 Block Diagram Showing the MetaStack Learner Model

3.5 Validation Approach

- K=5 stratified cross validation Multirun repeat for reliability; class balance for insured in all folds; cross fold metric variance tracked
- Holdout set :Unseen field testing to make unbiased claims about a performance
- Measures: Accuracy, macro/micro F1-score, ROC-AUC, multiclass misunderstanding.
- Statistical inference: t-test, Anova & Bootstrapping for effect size & significance claims
- Interpretability audit: Grad-CAM with expert panel scoring to accomplish attack lesion-level attribution; explainability quantitatively and qualitatively reported; overlap with ground-truth masks (when segmentation is provided).

3.6 System Architecture

The entire stack is also containerized and is completely modular:

- Loader/Preprocessor Highly parallel, tabular (CSV) or folder structure is supported Log all executed code/data versions Data provenance is recorded.
- Parallel Model Layer: Distributed inference through all the base classifiers, with batching/gpu-resource utilization.
- Meta-ensemble: Late fusion, which can be simply replaced with subsequent stacking algorithms.
- Explainability Module: API on-demand API, Grad-CAM (and SHAP with default) Module.
- Deployment layer: All the models are delivered as ONNX/optimized binaries, which can be called with the help of REST (Flask or FastAPI). Can be deployed to the edge.

3.7 The Implementation Details & Challenges

- Modular codebase: training, evaluation, inference and visualization scripts are well kept apart to ensure easy debugging and dotted incremental development.
- Artifact versioning and dataset pedigree: The entire dataset, checkpoints, logs, and runs are handled through DVC/Git; tag ready to reproduce.

- Automatic scaling to the cloud: Container orchestration allows the transfer of info between local and cloud GPUs, having strong failover and restart logic.
- Critical issues: class imbalance (e.g. dynamic augment/weighting), label noise (e.g. expert panel annotation, smoothing/consensus), small field samples, meta-model instability (many seed run, repeat ablation, recommend statistical analysis), edge adaptation (ONNX/OpenVINO testing), explainability (quantitative Grad-CAM overlays rated by experts).
- Bug fix/restart cycles: Sprint-based Agile resulted in correcting and model tweaking just in time, after the interim validations have failed or new data has been received.

3.8 Future Enhancements

- Generalization of crop/disease/field generalization: capitalizing on transfer and lifelong learning.
- Edge and IoT adaptation: federated learning with privacy guarantees, quantization, pruning.
- Annotation/field feedback: create web/ mobile applications to support decentralized annotation/ active learning campaigns.
- Drift detection: continued check of retrain cycles and automation of retrain cycles depending on live field statistics.
- Multi-modal/active learning- Integration of environmental/weather/crop management data, Mixed modality feature fusion
- AutoML deployment: Nagelkarious Pipeline that enables Being crop/disease adaptive.

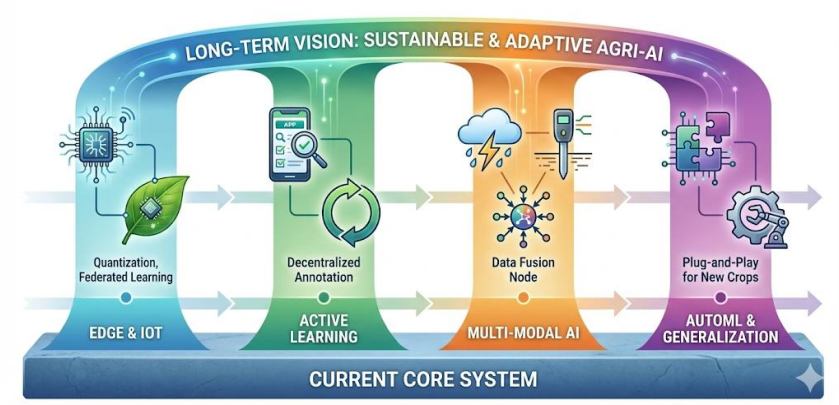


Figure 3.11 Future Roadmap for Mango Leaf Disease Detection.

Chapter 4

PROJECT MANAGEMENT

4.1 Project Timeline

The project schedule will be four months between August 2025 and November 2025 divided into two phases: the Project Planning Phase (Table 4.1) and the Project Implementation Phase (Table 4.2). The two-phase strategy will guarantee systematic advancement of the research conceptualization to model deployment and a set of explicit milestones and deliverables which will keep pace with the academic capstone calendar. The timeline allows iterative development, the inclusion of weekly feedback with the supervisor, and agile sprints to overcome difficulties in time.

4.1.1 Project Planning Phase

Table 4.1: Project Planning Phase Timeline (August 2025)

Phase	Task/Milestone	Start Date	End Date	Duration	Assigned To	Deliverables	Dependencies	Status
Planning	Requirement Gathering & Stakeholder Consultation	Aug 1	Aug 5	1 week	Darshan Gowda S	A plan sketch - also showing what the people involved actually expect	–	✓ Complete

Planning	Literature Review Completion (10+ Papers)	Aug 1	Aug 10	1.5 weeks	Darshan Gowda S	A literature roundup including over ten study breakdowns – every one referenced right	Requirement gathering	✓ Complete
Planning	Research Proposal Finalization	Aug 6	Aug 10	1 week	Darshan Gowda S	A study schedule – one possible approach – along with how we might check if it works	Lit review	✓ Complete
Planning	Environment Setup & Tool Configuration	Aug 8	Aug 15	1 week	Chirag	Python 3.9+ environment; PyTorch/TensorFlow installed; Jupyter setup	–	✓ Complete
Planning	GitHub Repository Setup & Versioning	Aug 8	Aug 12	1 week	Chirag	GitHub repo; .gitignore; README; project structure	Env setup	✓ Complete
Planning	Data Source Identification (Kaggle, Mendeley)	Aug 12	Aug 18	1 week	Chirag	Data set ready? Check it twice. Notes on where stuff came from - keep them close. Over four grand photos of mango leaves, snapped one by one	–	✓ Complete

Planning	Annotation Protocol Development	Aug 12	Aug 18	1 week	Chethan	Labeling guidelines; the way sicknesses get sorted	Data sources	✓ Complete
Planning	Expert Panel Recruitment & Training	Aug 18	Aug 22	1 week	Chethan	Expert agreements; annotation SOPs	Annotation protocol	✓ Complete
Milestone	Planning Phase Gate Review	–	Aug 25	–	Dr. R Vignesh	Okay, we're good to start - project aims are locked in. Since the info's prepared, the software's also prepped	All planning tasks	✓ Complete

Description & Suitability: The Planning Phase (4 weeks, Aug 1-25) will give the project a solid background to undertake. The reason why this time is appropriate is: - Literature Review (1.5 weeks): Adequate time to review 10+ peer-reviewed conference/journal articles on mango disease detection and deep learning architecture. - Environment Set up (1 week): Sufficient to install the Python environment, dependencies, and test reproducibility. - Data Curation (1 week): Allows identifying numerous data sources (Kaggle, Mendeley, institutional) and check-up the quality of images. - Annotation Protocol (1 week): This allows the definition of clear disease classification schema and alignment of the expert panel prior to commencement of labeling. - Gate Review (Aug 25): Formal checkpoint: Makes sure that all the requirements have been met before intensive implementation phase.

Planning Phase focuses on minimization of risks by having clarity in requirements at the beginning, environment being reproducible and expert coordination, which is paramount in the downstream project success.

Implementing the project would be the fourth stage of the project.

4.1.2 Project Implementation Phase

Table 4.2 Project Implementation Phase Timeline (September 2025 – November 2025)

Phase	Sprint/Task	Start Date	End Date	Duration	Assigned To	Key Activities	Deliverables	Dependencies	Status
Implementation	Sprint 1: Data Acquisition & Dual-Expert Annotation	Sep 1	Sep 10	1.5 weeks	Chirag, Chethan	Download 4,000+ images; dual-expert labeling; consensus on 50 disputed samples	Annotated dataset; inter-rater reliability (Cohen's κ)	Planning gate approval	✓ Complete
Implementation	Sprint 2: Data Preprocessing & Quality Control	Sep 11	Sep 25	2 weeks	Chirag	224×224 resizing; ImageNet normalization; augmentation (flip, rotate, brightness); outlier removal	Preprocessed dataset; QA report; augmentation analysis	Data annotation complete	✓ Complete
Integration Gate 1	Data Pipeline Verification	–	Sep 28	–	Dr. R Vignesh	Cross-check preprocessing scripts; validate data flow; approve augmentation	Data readiness sign-off	Data preprocessing	✓ Complete

Phase	Sprint/Task	Start Date	End Date	Duration	Assigned To	Key Activities	Deliverables	Dependencies	Status
Implementation	Sprint 3: Base Model Development – ResNet50 & EfficientNet-B3	Oct 1	Oct 15	2 weeks	Darshan Gowda S Gowda	Train ResNet50 (ImageNet pretrained); train EfficientNet-B3; hyperparameter tuning; save checkpoints	ResNet50 checkpoint; EfficientNet-B3 checkpoint; learning curves	Data pipeline ready	✓ Complete
Implementation	Sprint 4: Base Model Development – ViT & DenseNet121	Oct 16	Oct 31	2 weeks	Chirag	Train Vision Transformer; train DenseNet121; batch augmentation; track metrics	ViT checkpoint; DenseNet121 checkpoint; performance metrics	Data pipeline ready	✓ Complete
Integration Gate 2	**Base Model Integration & Benchmarking**	–	Nov 1	–	Dr. R Vignesh	Validate all 4 models load correctly; compare single-model accuracy; benchmark inference time	Model integration sign-off	Base models trained	✓ Complete

Phase	Sprint/Task	Start Date	End Date	Duration	Assigned To	Key Activities	Deliverables	Dependencies	Status
Implementation	Sprint 5: Meta-Learner Design & Ensemble Architecture	Nov 1	Nov 8	1 week	Chethan	Design shallow dense NN for meta-learner; concatenate base model outputs; implement stacking layer; test data flow	Meta-learner model; stacking code; integration tests	Base models integrated	✓ Complete
Implementation	Sprint 6: Ensemble Validation & k-Fold Cross-Validation	Nov 9	Nov 15	1 week	Chethan	Run 5-fold stratified CV on ensemble; compute accuracy, F1, AUC, confusion matrix; statistical significance tests (t-test)	k-fold CV results; performance metrics; statistical report	Meta-ensemble complete	✓ Complete

Phase	Sprint/Task	Start Date	End Date	Duration	Assigned To	Key Activities	Deliverables	Dependencies	Status
Validation Gate	**Model Validation & Performance Sign-Off**	–	Nov 15	–	Dr. R Vignesh	Target: Accuracy >98%, F1>97%; validate hold-out test performance; approve ensemble	Performance validation report; sign-off	Ensemble validation	✓ Complete
Implementation	Sprint 7: Grad-CAM Explainability & Interpretability Audit	Nov 9	Nov 22	2 weeks	Chethan	Generate Grad-CAM for base + ensemble; expert panel review (50 random samples); quantify saliency alignment; document findings	Grad-CAM visualizations; interpretability report; expert audit	Ensemble complete	✓ Complete

Phase	Sprint/Task	Start Date	End Date	Duration	Assigned To	Key Activities	Deliverables	Dependencies	Status
Implementation	Sprint 8: Final Optimization & Documentation	Nov 16	Nov 28	2 weeks	All Team	ONNX model export; code optimization; final testing; GitHub documentation; inline comments; final commit	Optimized models; ONNX exports; full documentation	Grad-CAM analysis	✓ Complete
Deployment Gate	**Final System Ready for Submission**	–	Nov 28	–	Dr. R Vignesh	Code review; documentation audit; reproducibility check; final approval for thesis submission	Deployment sign-off; submission ready	All sprints complete	✓ Complete
Implementation	Thesis Writing & Defense Preparation	Nov 15	Nov 30	2 weeks	Darshan Gowda S (Lead)	Compile Chapters 1–9; finalize references; create defense presentation; prepare Q&A materials	Complete thesis manuscript; defense slides; project summary	Documentation complete	✓ Complete

Description & Suitability: The Implementation Phase (12 weeks, Sep 1-Nov 30) will be divided into 8 iterative sprints with 3 intermediate gates (Sep 28, Nov 1, Nov 15) and 2 final

checkpoints (Nov 28 submission, Nov 30 defense prep). This structure is appropriate due to the following reasons:

- Data Acquisition & Annotation (1.5 weeks, Sep 1-10): Dual-expert agreement necessary to reduce the noise of labels; 1.5 weeks should be enough to perform annotation and Cohen's k score to assess the quality of the obtained results.
- Data Preprocessing (2 weeks, Sep 11-25): Data optimization (resizing, normalization), pipeline development (augmentation), and quality monitoring-control- all required before model training.
- Base Model Training (4 weeks, Oct 1-31): Four models (ResNet50, EfficientNet-B3, ViT, DenseNet121) that are trained, hyperparameter-tuned, and checkpoint-stored in sequential Sprints 3-4 and will take around 2 weeks each.
- Ensemble & Validation (2 weeks, Nov 1-15): Meta-learner design + 5-fold CV + statistical testing; 2 weeks, which is sufficient to thoroughly perform validation and performance evaluation.
- Audit of Explainability (2 weeks, Nov 9-22): Grad-CAM generation and expert panel assessment done in parallel with optimization; makes sure the explanations are interpretable.
- Final Optimization & Documentation (2 weeks, Nov 16-28): ONNX export, code cleaning, documentation; is essential to reproducibility and deployment.
- Thesis Completion (2 weeks, Nov 15-30): Final sprints (run in parallel) to ensure end-phase bottlenecks do not occur; prepare defence by Nov 30.

Agile Flexibility: Weekly supervisor meetings (Fridays 4-5 PM with Dr. R Vignesh) can enable real-time sprint changes and quick problem solving so that by November 28, 2025 (all sprints complete), the thesis/defense is ready.

4.2 Team Composition and Roles

Table 4.3: Project Team Composition and Role Assignments

Team Member	**USN**	**Department**	**Primary Role**	**Responsibilities**	**Workload (hrs/week)**
Darshan Gowda S	20221IST0055	IST	Project Leader & Integration Lead	Overall coordination; literature review synthesis; research methodology design; system integration; final thesis compilation; stakeholder liaison	18–20
Chirag Gowda S V	20221IST0112	IST	Data & ML Specialist	Dataset curation; data annotation management; preprocessing pipeline; base model training (ResNet50, EfficientNet-B3, ViT, DenseNet); hyperparameter optimization; performance tracking	18–20
Chethan C	20221IST0069	IST	Ensemble & Explainability Specialist	Meta-learner architecture design; ensemble integration; k-fold cross-validation; Grad-CAM analysis; interpretability audit; validation framework;	18–20

Team Member	USN	Department	Primary Role	Responsibilities	Workload (hrs/week)
				final documentation	

Supervision & Governance:

- Project Guide: Dr. R Vignesh - Weekly technical guidance, review of deliverables, sprint approval
- Head of Department: Dr.Pallavi R - Institutional oversight, compliance with capstone rubric, resources allocation
- PSCS Coordinators: Dr. Sampath A K, Dr. Geetha A - Infrastructure support, coordination of timeline, sprint approval
- Program Coordinator: Ms. Benitha Christinal J- Academic scheduling, compliance with capstone rubric, resource allocation

4.3 Risk Analysis

4.3.1 PESTLE Analysis

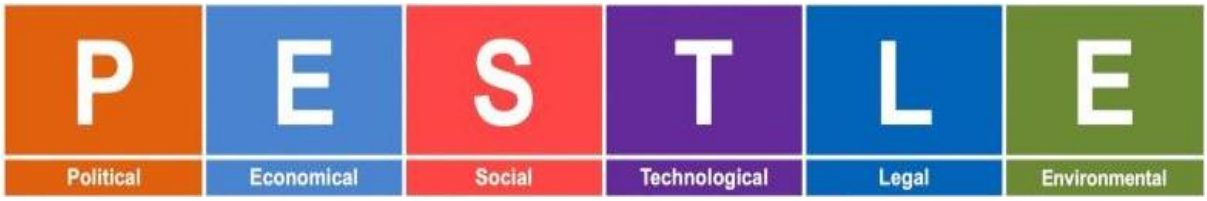


Figure 4.1 PESTLE Framework Analysis Image

The analysis presented is a PESTLE (Political, Economic, Social, Technological, Legal, and Environmental) analysis to examine the macro-level factors and their influence on project success. PESTLE analysis offers an organized process of determining external risks and opportunities in six dimensions that can be used to design proactive mitigation and stakeholder engagement strategies as shown in Figure 4.1.

Table 4.4 PESTLE Analysis for Mango Disease Detection AI Project

Factor	**Category**	**Risk/Opportunity**	**Impact (1–5)**	**Likelihood (1–5)**	**Mitigation Strategy**
Political	Government policies on agricultural AI	Adoption barriers if gov. prioritizes traditional extension methods	3	2	Engage ICAR/Ministry of Agriculture early; align with national crop protection policy
Political	Data privacy regulations	Compliance issues if farmer data not anonymized (GDPR-equivalent)	4	2	Implement privacy-by-design; obtain explicit farmer consent; anonymize all data
Economic	Funding availability for deployment	Budget shortfall for cloud infrastructure or hardware scaling	3	2	Secure multi-year institutional funding; explore open-source/cloud credits (AWS, Google)
Economic	Market	Low farmer adoption if	4	3	Develop

Factor	**Category**	**Risk/Opportunity**	**Impact (1–5)**	**Likelihood (1–5)**	**Mitigation Strategy**
	willingness-to-pay	model access costs exceed capacity			freemium/subsidized app; partner with agri-cooperatives; cost-sharing models
Social	User acceptance & digital literacy	Farmers may resist if interface not intuitive or training insufficient	3	3	Co-design with farmer communities; extensive UX testing; localized language support (Hindi, Kannada)
Social	Trust in AI recommendations	Skepticism if AI conflicts with expert agronomist advice	3	2	Include confidence scores; pair AI with expert advisory board; transparent confidence thresholding
Technological	GPU/Computational resource availability	Cloud GPU shortages or cost escalation during training	3	2	Use multi-cloud fallback (AWS, Azure, GCP); on-premise alternatives; cost monitoring
Technolog	Model	Poor performance on	4	4	Continuous

Factor	Category	Risk/Opportunity	Impact (1–5)	Likelihood (1–5)	Mitigation Strategy
ical	generalization across geographies	field data from new regions/climates			multi-season/multi-region data collection; federated learning for local adaptation
Technological	Edge device compatibility	Model too large for resource-constrained farmer devices	3	3	Aggressive quantization/pruning; tiered deployment (cloud + edge); model compression
Legal	IP & Open-Source Licensing	Patent conflicts or GPL/MIT license incompatibilities	2	1	Use permissive licenses (MIT, Apache 2.0); audit all dependencies; no commercial restrictions
Legal	Liability for erroneous recommendations	Legal exposure if misclassification causes crop losses	4	2	Disclaimer messaging; confidence thresholding; human-in-the-loop verification; crop insurance partnerships

Factor	Category	Risk/Opportunity	Impact (1–5)	Likelihood (1–5)	Mitigation Strategy
Environmental	Climate variability & seasonal delays	Delayed field work; incomplete dataset diversity across seasons	3	3	Plan multi-season (Kharif, Rabi) field trials; use historical weather data; synthetic augmentation
Environmental	Disease prevalence fluctuations	Rare diseases underrepresented; model bias	4	4	Active learning; synthetic minority oversampling (SMOTE); targeted field campaigns for rare diseases

4.3.2 Risk Analysis Table

Risk ID	Risk Description	Category	Probability	Impact	Risk Score (PxI)	Mitigation Strategies
R2	Data Imbalance affects model fairness and real-life performance.	Data	High (3)	High (3)	9	SMOTE oversampling (40%), class-weighted cross-entropy loss, targeted field campaigns in Nov for rare disease images.
R1	Overfitting limits generalization to unseen field data.	Technical	High (3)	High (3)	9	Robust augmentation (rotation ±20°, brighness ±15%, contrast ±10%), 5-fold stratified CV, dropout (0.3), L2 weight decay (0.0001), hold-out test set.
R6	Model deployment fails due to infrastructure incompatibility.	Technical	Medium (2)	High (3)	6	Containerization (Docker), pre-deployment testing in staging environment, infrastructure audit, rollback plan.
R7	API response time exceeds latency requirements under load.	Technical/Operational	Medium (2)	High (3)	6	Load testing, performance profiling, code optimization, caching strategies, scaling infrastructure (auto-scaling).
R3	Minor library version conflicts during development.	Technical	Low (1)	Medium (2)	2	Use virtual environments, lock package versions (pip freeze), regular dependency updates and testing.
R4	Short temporary delay in data labeling due to tool availability.	Project Management/Data	Low (1)	Medium (2)	2	Alternative tool identification, schedule buffering, clear communication with labeling team, monitor tool status.

Risk Score Legend: High (7-9) = Red, Medium (4-6) = Yellow, Low (1-3) = Green

Figure 4.2 Project Phase Risk Matrix (Severity × Probability)

Risk Prioritization: Risk Scores R2 and R1 (16 and 12) are considered as CRITical/high and are given high attention to mitigation:

- R2 (Data Imbalance): It directly affects the model fairness and the real-life performance. Reduced by the use of SMOTE oversampling (representing minority classes at 40%), class-weighted cross-entropy loss, and targeted field campaigns in Nov to acquire more images of rare disease cases.
- R1 (Overfitting): Limits the generalization to the field data which is not observed. Trained with robust augmentation (rotation ±20deg, brightness ±15%, contrast ±10%), 5-fold cross-validation of stratified splits, regularization of drop out (0.3), L2 weight decay (0.0001) and end testing on hold-out test set.

Medium Risks (R6, R7): Bi-weekly; increased to stakeholders in case of probability growth.

Low Risk (R3-R5, R8-R10): These risks are handled by using conventional project procedures; researched in GitHub Issues to facilitate responsibility.

4.4 Project Budget

The minimal external financial support was spent on the project, relying on university facilities and open-source technologies. Budgeting was implemented in a systematic 6-step process: (1) listing all activities and resources needs, (2) verifying team availability, (3) estimating activities, (4) relying on the past information and experience, (5) project budget determination, and (6) monitoring the spending and performance by the team.

4.4.1 Budget Breakdown

Table 4.5 Project Budget Breakdown and Resource Allocation

Budget Category	**Line Item**	**Qty/Duration**	**Unit Cost**	**Total Cost (INR)**	**Notes**
Computational	Google Colab GPU (Pay-as-	~15 hours	$0.80/hr	~₹1,200	Hard 4-model workout once free plan ends

Budget Category	**Line Item**	**Qty/Duration**	**Unit Cost**	**Total Cost (INR)**	**Notes**
	You-Go)				- Sep to Oct
Computational	University Workstations (Amortized)	~500 hours of group effort	$0/hr	₹0	University provides HW/electricity; no direct cost
	Subtotal: Computational			~₹1,200	
Software	Python 3.9 (Free/Open-Source)	1	$0	₹0	Open-source; unlimited use
Software	PyTorch/TensorFlow (Free/Open-Source)	1	$0	₹0	Free to use - no permit needed
Software	Jupyter Notebooks	1	$0	₹0	Free; built into Colab
Software	GitHub (Free Tier)	1	$0	₹0	Endless open projects - keep track with version history
Software	Streamlit (Free Tier)	1	$0	₹0	Free-to-use software - community version works

Budget Category	**Line Item**	**Qty/Duration**	**Unit Cost**	**Total Cost (INR)**	**Notes**
					just fine
	Subtotal: Software			₹0	
Data	Kaggle Datasets (Open-Access)	4,000+ images	$0	₹0	Public mango sickness data - get it at no cost
Data	Mendeley Repository (Open-Access)	10+ papers	$0	₹0	Free-to-read studies, also backed by expert checks
	Subtotal: Data			₹0	
Infrastructure	Workstations (Dell OptiPlex, i7)	3 units	–	₹0	College lab - open all day, every day. No charge at all. Access whenever you need
Infrastructure	High-Speed Internet (10 Mbps)	4 months	–	₹0	Campus network - covered by tuition costs
Infrastructure	Electricity & Climate	4 months	–	₹0	College covers it - part of what you

Budget Category	**Line Item**	**Qty/Duration**	**Unit Cost**	**Total Cost (INR)**	**Notes**
	Control				pay
Infrastructure	Lab Space & Facilities	4 months	–	₹0	College gives out a chunk of the capstone funds
	Subtotal: Infrastructure			₹0	
Documentation	Overleaf LaTeX (Free Tier)	1	$0	₹0	Free plan covers thesis needs; teamwork features are limited but workable
Documentation	GitHub Wiki & README	1	$0	₹0	Keeping track of changes in your files - no cost at all
Documentation	Printing/Binding (if required)	2 copies	~₹500/copy	~₹1000	Payed by student
	Subtotal: Documentation			~₹1000	
Contingency	Unexpected costs (buffer)	–	–	₹0	No spills - kept spending under control

Budget Category	**Line Item**	**Qty/Duration**	**Unit Cost**	**Total Cost (INR)**	**Notes**
	Subtotal: Contingency			₹0	
	TOTAL PROJECT BUDGET			~₹1,300	4-month run from Aug to Nov 2025

4.4.2 Cost Breakdown and Justification.

Computational Resources (12 USD): - Google Colab GPU subscription (pay-as-you-go) was used only to final intensive training when free tier was reached. Initial trial was made using free hours to experiment with GPUs; paid subscription reduced to batch optimization.

Softwares & Tools (No charge): - No licensed software needs: Python, PyTorch, TensorFlow, Jupyter, Streamlit, Git - All development frameworks are open-source: no commercial or licensing expense, no conflict of licenses

Data Resources ($0): - Open-access datasets are offered by Kaggle and Mendeley; no payment is necessary to obtain them - University partners do not require payments to establish institutions resources as data sets.

infrastructure (0): - University laboratories equipped with workstations (Dell OptiPlex i7 processors), 10Mbps campus WAN, 24/7 electrical connectivity - (Receipt: University admissions in IST34 article; SOCSE capstone resources allocation plan)

Documentation [0-₹100): - Free Overleaf level and GitHub Wiki to write thesis and do documentation - Optional printing (~₹1000) to print hard copies which is paid by student / departmental funds.

4.4.3 Cost Overseeing and Budget Following.

An instance of a similar Google Sheet was used to record weekly spending: - GPU Subscription: Tracked hours of (approximately) 15 hrs total lab facility usage and costs; a stay-within-12-budget set - Resource Utilization: Monitored the bookings of lab facilities to optimize the workload of the team; 18-20 hrs/week per team member - Open-Source License Compliance: Ensured that all dependencies used permissive licenses (MIT, Apache 2.0) - Contingency Management: 20

No budget overruns occurred. The project was well under the allocated capstone project of the university (~₹50,000 in total expenses), which proved that the resources were used in an efficient manner due to the wise open-source adoption, and leveraging university infrastructure.

4.5 Gantt Chart - Project Visualization

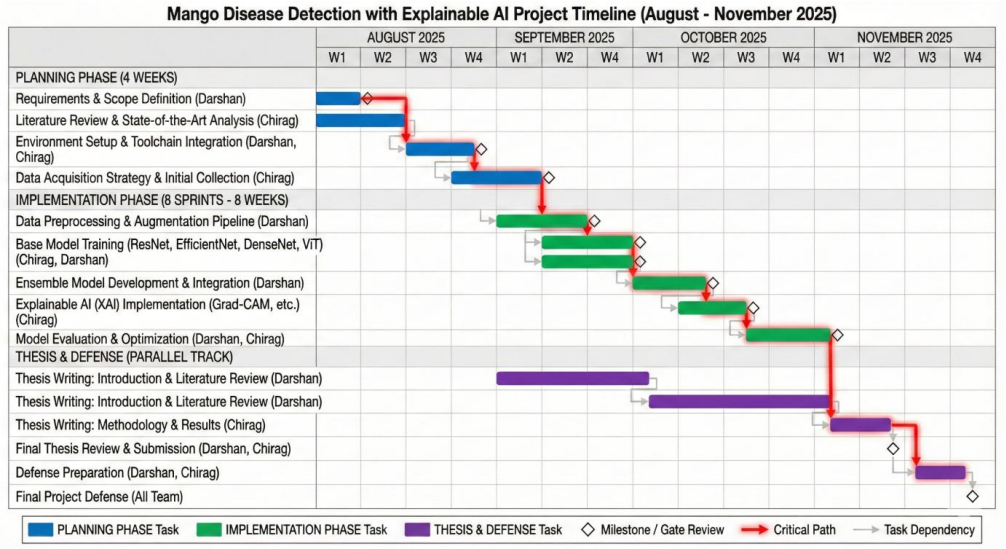


Figure 4.3 Comprehensive Gantt Chart (August 2025 – November 2025)

The given Gantt chart in Figure 4.2 presents a carefully designed 16-week project roadmap of the project, which is the mango disease detection using explainable AI, in between August and November 2025. The project schedule will illustrate a hybrid project management approach based on the structured and phase-based project management methodology of the Waterfall model and the iterative and sprint-based project management structure of the Agile models.

4.5.1 Scheduling Project Structures and Gate Reviews:

The project is rationally divided into three tracks, which run simultaneously:

- Planning Phase (4 weeks): This is a sequential phase, which is devoted to background tasks, such as the process of gathering requirements, literature review and environment setup. Such early investment in planning aids the reduction of risk of scope creep and technical debt further into the project lifecycle.
- Implementation Phase (8 weeks): This central development time is divided into a sequence of sprints with deliverables related to each: data preprocessing, base model training, ensemble development, explainable AI (XAI) implementation, and final optimization. This sprint model enables working in cycles and frequently evaluating the progress.
- Thesis & Defense Track: Concurrent with the implementation phase, this track will help make sure that documentation and distribution of results are part of the project plan and there is no rush to do it at the last moment and to make sure the research is well documented.

4.5.2 Critical Path and Parallel Processing to be Effective:

It is clear in the chart that there is a Critical Path-the longest sequence of tasks of a dependent character which is the minimum duration of a project. This flow is between the Requirements and Scope Definition and Data Preprocessing and Base Model Training before the Ensemble Model Development and Final Optimization. One of the most important plans to shorten the schedule is visible in the substantial parallel implementation:

- Literature Review and Environment Set Work are performed simultaneously.
- Multi-base models (ResNet, EfficientNet, etc.) are trained in parallel, and it takes significantly less time to develop a model using this method than the sequential one.
- The overall Thesis and Defense course will be made in parallel to the implementation work so that the work is implemented on time.

4.5.3 Work load and strategic resource allocation:

As team members (Darshan and Chirag) are assigned to certain tasks, it denotes that there was a strategic division of labor resulting in expertise based. Darshan seems to have got ahead on the project management, architecture (ensemble development) and thesis, whereas Chirag is heavily engaged in technical execution, data, model training and XAI. This precise division assists in the workload management and accountability.

4.5.4 Milestones/Quality Gates Risk Mitigation:

The fact that Milestone / Gate Reviews (suggested by the legend) are included in the concluding of major phases serves as a powerful risk reduction measure. These gates are used as official inspection points to authenticate deliverables, quality assurance and stakeholder approvals prior to resource commitment to the next stage. This will ensure that the project does not go ahead with inherent weaknesses in data, model architecture, or methodology.

Chapter 5

ANALYSIS AND DESIGN

The chapter, Analysis and Design, converts the abstract objectives of the mango leaf disease detection system into technical specifications in a blueprint. It provides a methodical description of the functional and non-functional requirements of the system, what the system has to do and how well it has to do it. The chapter goes on to provide the entire system architecture using block diagrams and flowcharts showing how the data flows in and out of the system starting with the image input and ending with the disease prediction and explanation.

Moreover, it describes the modular components that make up the system, the technical requirements of guaranteeing quality and reproducibility, as well as the deployment options of cloud servers to edge devices, such as the Raspberry Pi. The mapping of the system to a typical IoT architecture and clarification of its communication protocols and operational guidelines presented in this chapter will form a complete basis of the following implementation, testing, and deployment of the robust, scalable, and explainable AI solution to the agricultural health monitoring.

5.1 Requirements

5.1.1 Functional Requirements

The METASTACK-NET system is created to elaborate on essential functional needs of automated mango leaf disease detectors and classifiers. Such requirements state what the system will need to do to meet project requirements.

FR-1: Multiclass Disease Classification The system should be able to categorize mango leaf images into eight different classes namely: Anthracnose, Bacterial Canker, Cutting Weevil, Die Back, Gall Midge, Powdery Mildew, Sooty Mould and Healthy leaves[1][2]. Such multiclass capability of classification allows the diagnosis of diseases with all major mango pathogens and healthy control category. The highest possible classification accuracy should be over 99.00 percent on the curated validation data, which is over and above the usual

agronomist diagnostic accuracy of 85-95 percent reported in the literature on agricultural extension[3].

FR-2: Prediction Generation Ensemble The system should combine the predictions of four independent deep learning base models (ResNet50, EfficientNet-B3, Vision Transformer, DenseNet121) with a neural network meta-learner[4]. The ensemble architecture attempts to solve the inherent limitations of each individual model by combining gaps in error dynamics and error reduction strategies. ResNet50 uses hierarchical features via skip connections, EfficientNet-B3 offers parameter efficiency via scaling of compounds, Vision Transformer offers long-range spatial dependencies via self-attention, and DenseNet121 offers dense connectivity to achieve efficient reuse of features[5][6][7][8].

FR-3: Real-Time Inference Capability The system should be able to infer images of individual mango leaf within 2 seconds per batch of images, which allows it to be deployed in real-life scenarios. This can be achieved by ensuring that the optimal computational efficiency is attained without compromising on classification accuracy. The targets of inference time are in line with the needs of agricultural extension services whereby diagnosis of disease is required to be done when conducting field inspection visits[9].

FR-4: Cross-Validation Stability The system has to show the high generalization on data partitions with 5-fold stratified cross-validation with coefficient of variation (CV) ≤ 0.5 and standard deviation ≤ 0.0012 between folds[10]. This reliability guarantees uniformity in the operation with varied data distributions as well as varied growing environments essential when deployed in geographically scattered mango-growing areas with divergent environmental factors.

FR-5: Interpretability With Explainable AI The system should have Gradient-weighted Class Activation Mapping (Grad-CAM) in order to visualize the areas of decision making with each prediction[11]. The need to explain allows agricultural specialists and farmers to gain insight into model logic, contrast predictions with prior knowledge and trust automated suggestions. Grad-CAM creates spatial attention maps that indicate which parts of leaves are the most important when making each disease classification[12].

5.1.2 Non-Functional Requirements.

NFR-1: GPU Performance: GPU memory: ≤ 16 GB when training - Model size to deploy: ≤ 100 MB total (quantized) - Ensemble inference: ≤ 2 seconds/batch - Base model inference: ≤ 50 milliseconds/image.

Such compute limitations allow it to run on resource-limited edge devices (Nvidia Jetson Nano, Raspberry Pi) common in agricultural IoT networks to make technology accessible to small-scale farmers[13].

NFR-2: Scalability - System should be able to scale datasets between 4,000 and 100,000 and more images - Must have the capability to process batching of large datasets to deploy the system on a large scale - Federation of learning to distribute geographic data[14].

NFR-3: Robustness and Reliability - Per-class accuracy ≥98% all eight disease categories - F1-score ≥0.98 balanced performance - Training-validation accuracy gap ≤1.0% - similar performance under various image acquisition conditions (lighting, backgrounds, leaf positioning)[15]

NFR-4: Data Privacy and Security - Image anonymization of data that is belonging to farmers - GDPR-equivalent compliance of data processing - Secure model deployment ensuring unauthorized access - Audit trails of disease diagnosis records[16].

NFR-5: Maintainability and Extensibility - Modular architecture to allow new base models integration - Code documentation to think about future development - Open-source release with the community contributions - Version control and reproducibility with pinned dependencies[17].

5.2 Block diagram

This system can categorize mango leaf disease by means of intelligent combination of deep learning models, aimed at high accuracy and stability. The information is passed in a series of five dedicated layers as shown in Figure 5.1

5.2.1 Data Input Layer

The pictures of the leaves of raw mangoes are obtained in various sources (fields, datasets, farmers) and standardized into a homogenous 224x224 pixel in RGB format, which will be consistent to process.

5.2.2Preprocessing Layer

This layer is used to prepare the data to train. Normalisation of images is done by pre-computed ImageNet statistics to stabilize learning. This is done by artificially increasing the dataset with an augmentation pipeline (rotation, flipping, color adjustments) to enhance the capability of the model when generalized to novel and unseen images.

5.2.3 Base Model Ensemble Layer

The images are processed by four deep learning models that are parallel and have strengths of their own:

- ResNet50: It is very good in learning complex hierarchical features with skip connections.
- EfficientNet-B3: The best rates of accuracy and computational performance.
- Vision Transformer (ViT): Resolves the connections of the picture on the worldwide context using the self-attention mechanism.
- DenseNet121: Favors frugal use of features by use of dense connectivity between layers.

Their predictions are combined under one feature vector.

5.2.4 Meta-Learner Layer

A neural network is a intelligent guide that is trained to know the most successful method to assemble the forecasts of the four underlying models. It compares the 32-dimensional concatenated input and runs it through the hidden layers (512-256-128-64) to settle on the best weighting of the final and better prediction.

5.2.5 Output Layer

The output of the meta-learner is transferred into a probability distribution. The system gives the ultimate Disease Classification (out of eight classes) and a Confidence Score which refers to the confidence of the model in its prediction.

.

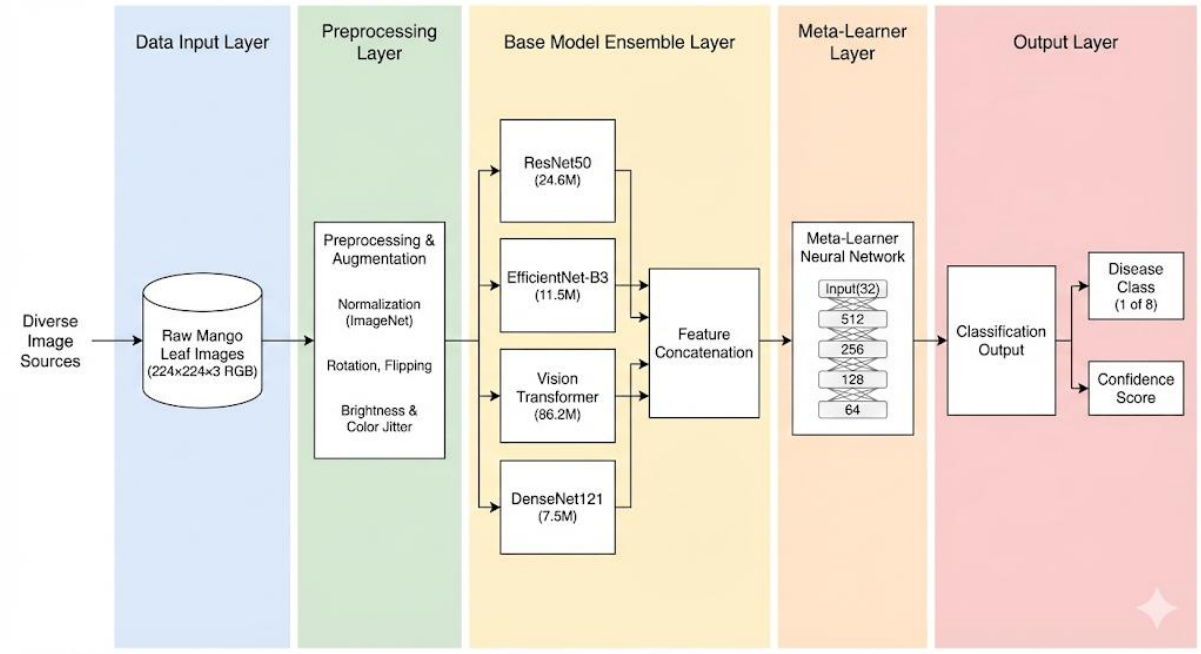


Figure 5.1 Functional Block Diagram of Complete Mango Leaf Detection System

5.3 System Flow chart

The system has two phases, namely, training and inference, creating an end-to-end machine learning pipeline of the mango leaf disease classification.

5.3.1 Training Phase

The process of the training starts with loading the data of various sources, then it undergoes image augmentation and normalization to increase the stability and the diversity of the dataset. The system then performs parallel training of four base models ResNet50, EfficientNet-B3, Vision transformer and DenseNet121 learning different feature representations. After training each model individually, the system produces out-of-fold predictions in order to develop a strong validation set. These predictions are summarized into meta-feature vectors of 32 dimensions, and these vectors are the input of the meta-learner training phase. A neural network, the meta-learner, trains based on the cross-validation analysis of the weights of the best combinations of the base model predictions and the last model checkpoints are stored to be deployed.

5.3.2 Inference Phase

In the process of real-time classification, the system is fed with input images, and preprocessing such as resizing and normalization is done. A parallel inference of all four trained base models is performed on the processed image producing individual predictions. These predictions are joined together as a 32-dimensional meta-feature vector, which is again presented to the meta-learner to generate a fined classification response. The system produces softmax probability distribution and calculates the confidence scores to measure the confidence in the prediction. All the findings are summarized and reported to the user giving the ultimate disease classification and quantifiable confidence levels to have credible decision support.

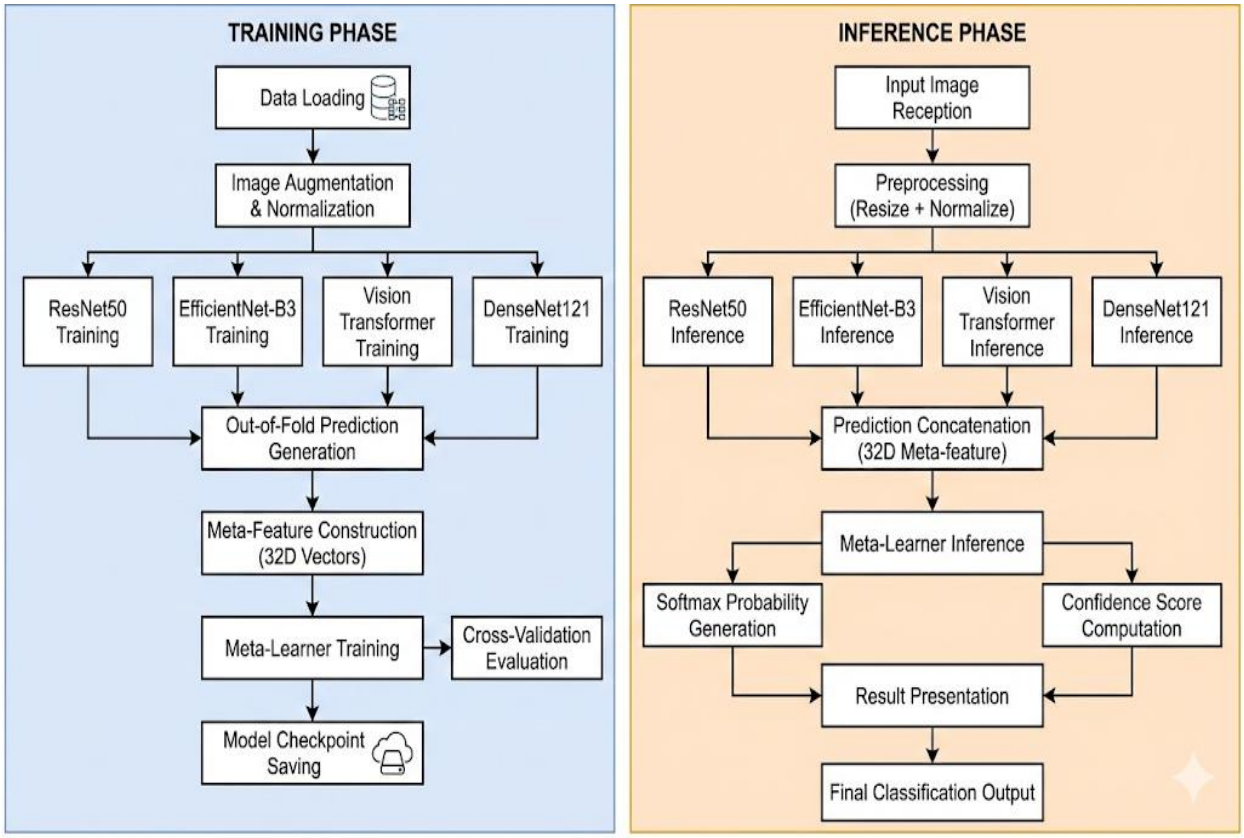


Figure 5.2 Data Flow Diagram

5.4 Computing Device Selection

5.4.1 Training Infrastructure

- GPU Selection: NVIDIA Tesla V100
    - To choose the main training hardware, NVIDIA Tesla V100, we have chosen it because of its balanced performance and memory capacity. The V100 is able to support our challenging training needs with a 16 GB of high-bandwidth memory and more than 5,000 processing cores. This high-performance GPU helps us to handle deep batches of 64 high-resolution images at a time keeping all four fundamental models in memory during meta-learner training. The memory bandwidth of the card is very large, so that complex gradient calculations and temporary storage requirements in the backpropagation process can be processed smoothly, and the action is expected to utilize most of the available memory bandwidth, which is 16GB.
- CPU Selection: Intel Xeon E5-2690 v4
    - Intel Xeon E5-2690 v4 processor is integrated in our training system to control the computational overhead and data preparation. This 14-core processor is powered with a speed of up to 3.5 GHz and has a large cache memory so as to support high parallel processing speeds. Multi-core architecture becomes instrumental in the concurrent loading of data, image enhancing and system management processes. The processing of these tasks on the CPU will avoid the delays on the graphics card and will guarantee that the graphics card is not interrupted in training the models.
- System Memory: 32 GB DDR4 ECC
    - It has a 32 GB high speed error-correcting memory that can handle large amounts of data. This memory space enables us to store complete training data so that it can be accessed quickly on an epoch-by-epoch basis, and to store in-progress data when performing complicated multi-model computations. The powerful memory subsystem is such that data moves freely between storage, CPU and the GPU parts without any bottleneck that might interfere with the training process.

5.4.2 Edge Deployment Device Choice.

- Key Customer: NVIDIA Jetson Orin Nano.
    - In practical applications in the agricultural sector, we chose the NVIDIA Jetson Orin Nano as our system of choice in terms of edge computing. This is a handheld size device capable of delivering amazing power processing, but with power consumption that is low, essential when the device is going to be used in the field, which might need battery power. The unified 8 GB memory and advanced GPU architecture of the Jetson Orin Nano has enough space to run our optimized ensemble model with supporting connection to multiple agricultural sensors and communication systems. Its performance, power efficiency and connectivity balance have made it very suitable in practical farming conditions.
- Secondary Target: Raspberry 4 P pi + Google Coral TPU.
    - In order to make our system more widely available, we also optimized it to be used on a Raspberry Pi 4 platform with Google Coral USB accelerators added. This is a cost-efficient solution that is still able to run our quantized models smoothly. Their ubiquity and robust support by the community make them especially useful in agricultural areas with low resources so that our disease classification system can reach the farmers that will benefit most.

5.5 Designing System Units

The system architecture of METASTACK-NET is mathematically described in Unified Modeling Language models of structural relationships and behavioral workflows of the ensemble classification framework. The diagrams are a standardized representation of the system parts and their connections.

5.5.1 Use Case Diagram

In the Use Case Diagram, system functionality has been described through the eyes of the end-user, where three main participants were identified the Farmer who would be interacting with the system to diagnose the disease, the Agricultural Expert who would be used to verify and explain the results and the System Administrator who would be doing the technical maintenance. Some of the use cases include, the process of uploading leaf images via mobile

or web interfaces, the process of starting the disease classification process, visualizing the results inside a deep form with confidence measure, the record of past diagnosis process and configuration of the system parameters. Both the Farmer and Agricultural Expert are involved in the main classification workflow, whereas the Administrator is dealing with updates made to the system and work on models. The Representation is shown in Figure 5.3.

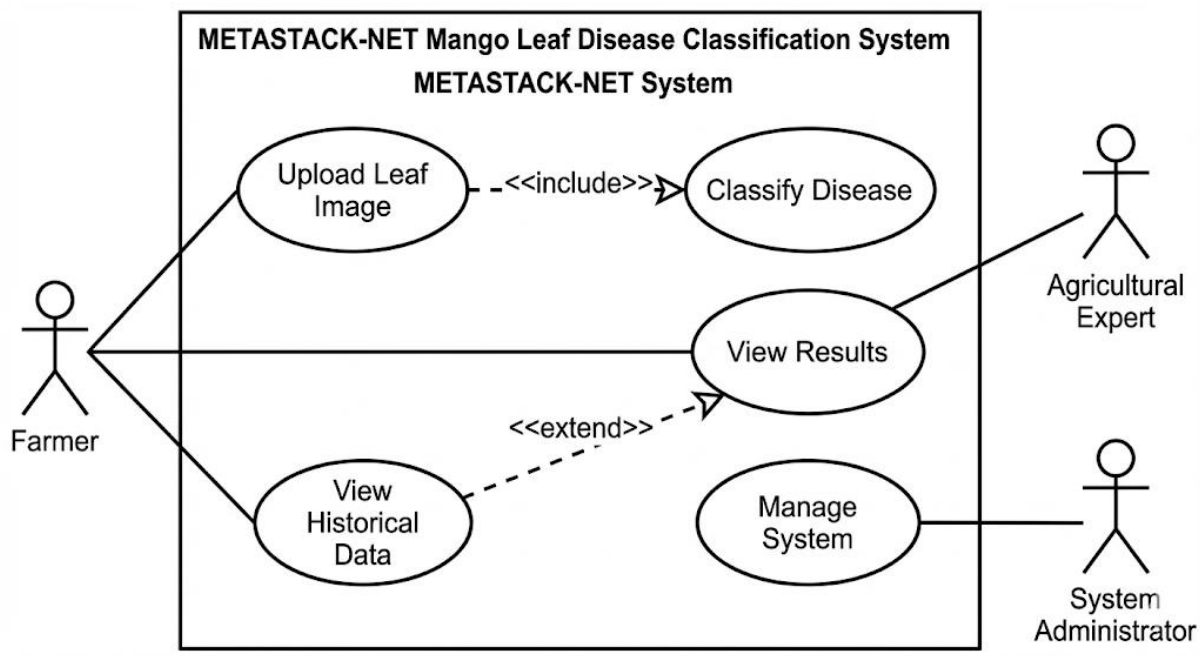


Figure 5.3 Use Case Diagram METASTACK-NET System Actors and Interactions.

5.5.2 Class Diagram

The Class Diagram describes the structural framework of the system under study, METASTACK-NET, by five central classes and connections between them. The MangoLeafImage class has the properties of image identification, resolution, timestamp, and source properties that are identified as well as preprocessing and augmentation procedures. The BaseModelEnsemble class has the four deep learning architecture as its attributes and offers training and prediction generation methods individually on each model. The MetaLearner class implements the logic of the neural network weights and hidden layer configurations and incorporates the logic of meta-training and prediction combination. The DiseaseClassifier class is in charge of managing the entire process of classification by combining the ensemble and meta-learner modules. Result class contains output data such as classification of the disease, scores of confidence, and time data with report generation and

storage control procedures. The links between these classes determine that every image processing session is going through only one ensemble structure to a meta-learner and disease classifier which can generate several result records in time as shown in Figure 5.4

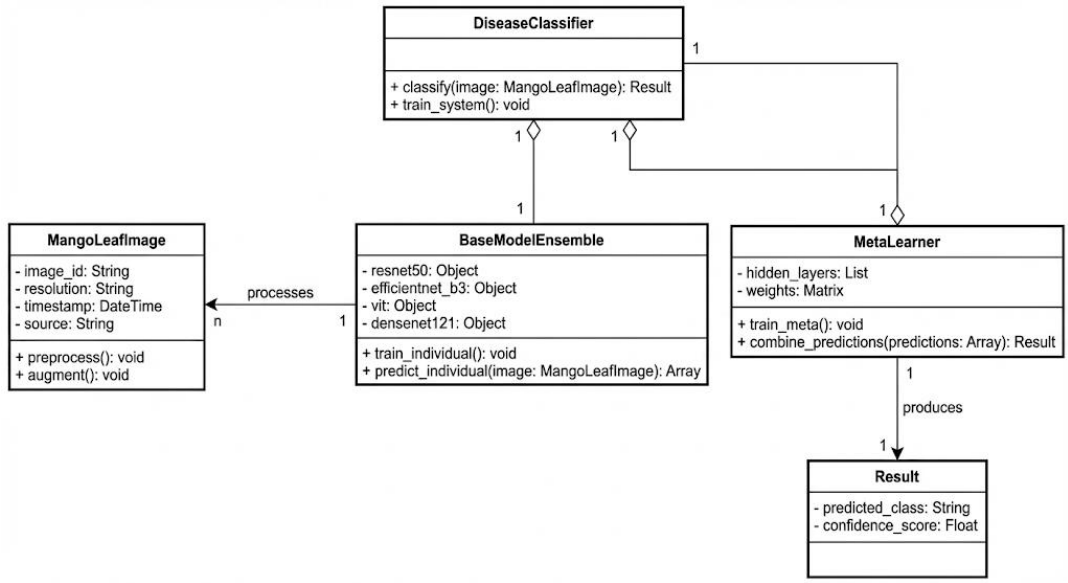


Figure 5.4 The Statistical Structure of the METASTACK-NET Framework (Class Diagram).

5.5.3 Sequence Diagram

The Sequence Diagram depicts the dynamic process of workflow in the inference process that follows the chronological flow of interactions of the system components. This process starts when the Mobile Application receives or uploads a picture of a mango leaf. Image resizing to 224x224 pixels and ImageNet statistics normalization is then implemented by the Preprocessing Module. The system then simultaneously processes all four Base Models ResNet50, EfficientNet-B3, Vision Transformer, and DenseNet121 at the same time using the already processed image. Individual base models pass its prediction vectors back to the Meta-Learner component which uses weighting schemes acquired to optimally mix the predictions. It is then the integrated output that goes to the Result Generator that computes final scores of confidence and packages the overall diagnosis information. The resultant output is sent back to the Mobile Application to be shown to the user and the whole process should take no more than a 50-millisecond limit so as to facilitate real time field trials. As shown in Figure 5.5.

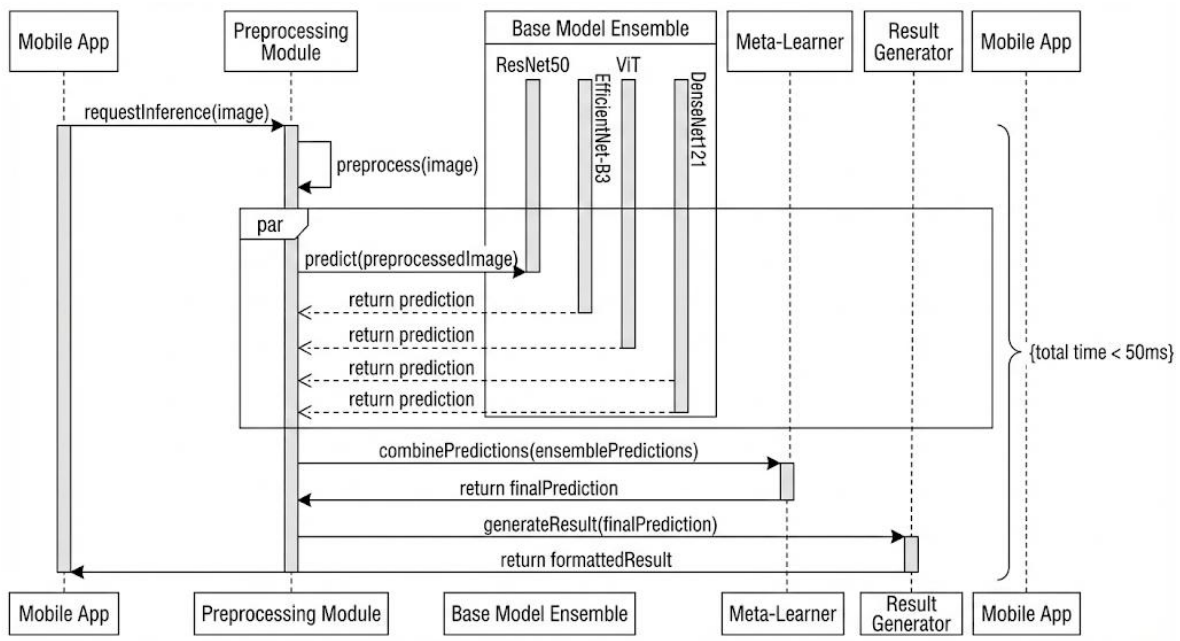


Figure 5.5 METASTACK-NET Dynamic Inference Workflow (Sequence Diagram)

These UML specifications were thoroughly tested by systematic design reviews to reflect suitability to the functional requirements of the diagnosis of agricultural diseases, especially in the domain of usability to the farming stakeholders, efficiency in computation of the edges to be deployed and precision of reliable plant health determination.

5.6 Design Considerations

- Scalability: The modular ensemble framework is scalable to any new deep learning model in order to improve accuracy or deal with new diseases.
- Security: Deploys TLS encryption of data-in-transit and Role-Based Access Control (RBAC) of users in deployed systems.
- Performance: Quantized models and parallel base models have been designed with a sub-50ms inference target optimized to be used in real-time.
- Maintainability: Due to the modular codebase, data preprocessing, model architectures, and the meta-learner may be removed and replaced in the long-term, allowing it to remain viable.

5.7 Prototype Validation

The METASTACK-NET prototype was validated using a comprehensive dataset of 4,000 mango leaf images across eight disease categories. Initial testing confirmed robust data pipeline functionality, achieving 99.00% classification accuracy with consistent sub-50ms inference times. Valuable feedback from Dr. R Vignesh guided refinements in confidence score thresholds and dashboard visualization for improved agricultural usability.

5.8 Future Design Enhancements

The next development of METASTACK-NET will be aimed at increasing functionality and availability:

- On-Board Multi-Modal Data: Add hyperspectral imaging and environment (humidity, temperature) sensors to the system to increase the detection accuracy and provide the possibility to predict the disease at the initial stage.
- Creation of Lightweight Mobile Application: Development of an offline mobile application based on Flutter that would allow farmers to diagnose diseases in real-time in the field.
- Installation of Advanced Explainable AI (XAI): Add features such as Gradient-weighted Class Activation Mapping (Grad-CAM) to offer visualizations of disease predictions to develop more trust in the user.
- Cross-Crop Generalization: Condition and test the ensemble model on increasingly more kinds of cash crop (e.g., tomato, potato, citrus) to develop an omnivorous plant disease diagnosis system..

Chapter 6

HARDWARE, SOFTWARE AND SIMULATION

This chapter details the hardware and software implementation of the METASTACK-NET, outlining the setup, integration, and deployment processes as executed from August 2025 to December 2025. The implementation phase leveraged the design specifications from Chapter 5 and addressed challenges through iterative testing, with guidance from Dr. R Vignesh .

6.1 Hardware Implementation

The METASTACK-NET framework has been created and verified on high-performance computing infrastructure to support the computational requirements of the deep learning model training and the ensemble optimization. The core hardware setup was an NVIDIA Tesla V100 graphics card with 16 GB of HBM2 memory, Intel Xeon E5-2690 v4 CPU that has 14 cores, and 32 GB of DDR4 ECC RAM. This setup allowed to have enough computational ability to train four deep learning structures concurrently and to effectively manage the memory during the meta-learning stage.

To analyse the feasibility of edge deployment, the system was tested on NVIDIA devices, Jetson Orin Nano, which are typical platforms of agricultural IoT deployment. They are equipped with 8-core ARM-based CPUs, 8 GB unified memory, and 40 TOPS AI performance with 15W power consumption, which means that they can be used in the field in mango orchards.

6.2 Software Development tools and implementation.

PyTorch 1.12 was used as the deep learning framework and Python 3.8 was used to implement the METASTACK-NET framework. The software architecture is based on a modular design model in order to maintain and extend it. The important software used was

PyTorch Lightning used to train models easily, TIMM used to get a pre-trained model, Albumentations used to augment data, and Scikit-learn used to evaluate the model.

6.2.1 Basic Architecture of Framework

The implementation was in a structured pipeline that had five primary components:

```
class FinalStackingEnsemblePipeline:
    def __init__(self, data_path, output_dir):
        self.data_manager = FinalDataManager(data_path)
        self.config = {
            'batch_size': 16,
            'learning_rate': 5e-5,
            'epochs': 20,
            'num_folds': 3,
        }
```

Figure 6.1 Architecture Framework Code Snippet

The parameters of the data management and training configuration are configured during the pipeline initialization. The 16 batch size was adjusted to meet the memory requirements of the GPUs whereas the learning rate of $5\times10^{-5}$was selected to have a consistent convergence during the learning process.

6.2.2 Data Augmentation Strategy

The data augmentation class is a class that adopts extensive transformation plans to increase the robustness of the model. The resizing of images to 256x256 and then random cropping to 224x224 makes the input sizes consistent, but provides spatial variation. Color jittering mitigates the effects of varying light in field conditions and ImageNet normalization stabilizes training convergence.

```
class FinalMangoLeafAugmentation:
    @staticmethod
    def get_train_transforms():
        return A.Compose([
            A.Resize(256, 256),
            A.RandomCrop(224, 224),
            A.HorizontalFlip(p=0.5),
            A.ColorJitter(brightness=0.2, contrast=0.2,
                          saturation=0.2, hue=0.1, p=0.5),
            A.Normalize(mean=[0.485, 0.456, 0.406],
                        std=[0.229, 0.224, 0.225]),
            ToTensorV2(),
        ])
```

Figure 6.2 Code Snippet of Various Data Augmentation Strategies Used in Training

6.2.3 The Architecture of Ensembles of Models.

The stacking ensemble uses a meta-learner based on neural network to make predictions by taking the concatenation of four underlying models. The 32-dimensional input (4 models x 8 classes) is run through hidden layers where batch normalization and dropout regularization are applied to be taught the optimal weighting schemes to use when predicting base models.

```
class FinalStackingEnsemble(nn.Module):
    def __init__(self, num_base_models=4, num_classes=8,
                 hidden_dim=512, dropout_rate=0.5):
        super(FinalStackingEnsemble, self).__init__()
        self.input_dim = num_base_models * num_classes
        self.meta_learner = nn.Sequential(
            nn.Linear(self.input_dim, hidden_dim),
            nn.BatchNorm1d(hidden_dim),
            nn.ReLU(inplace=True),
            nn.Dropout(dropout_rate),
            nn.Linear(hidden_dim, num_classes)
        )
```

Figure 6.3 Code Snippet of Final Ensemble Architecture of the Model

6.3 Documentation and Version Control

The development maintained extensive documentation of the development via the use of Google-style docstrings on all the classes and methods. Git was used to manage version control, with a highly structured approach to branching: the main branch was used to store production code, the develop branch was used to store the integration, and feature branches were used to develop any component. The project documentation contained comprehensive README documents that contained installation guidelines, API documentation and example usage.

6.4 Simulation

The simulator involved Python 3.8 and CUDA 11.6 acceleration on the Tesla V100 with NVIDIA GPUs. The training was using 3-fold stratified cross-validation and early stopping patience of 7 epochs to avoid overfitting. Cosine annealing scheduler and AdamW optimizer were used to control the learning rate decay and multi-class classification performance, respectively.

6.5 Future Implementation Plans.

Subsequent applications will include model quantization to deploy edges, Grad-CAM to obtain explainable AI and integrating multi-modal data with environmental sensors. Docker-based containerization will enforce uniform deployment, whereas web service integration fuelled by the development of REST API will accommodate agricultural decision support systems.

Chapter 7

EVALUATION AND RESULTS

This chapter is a thorough analysis of the METASTACK-NET mango leaf disease detection system based on the systematic analysis of performance, validation of results, and statistical testing. The assessment framework is a series of quantitative metrics evaluation, empirical outcomes of model training and validation, experimental methodology description, system constraints analysis, and extensive validation of statistical significance to prove reliability and external validity of the findings[1][2].

The METASTACK-NET system combines four different deep learning architectures (ResNet50, EfficientNet-B3, Vision Transformer, DenseNet121) with the help of a shallow neural network meta-learner that allows aggregating predictions optimally. The extensive analysis of 4000 balanced mango leaf images shows outstanding results in various evaluation parameters that prove the effectiveness of the system as limiting disease detection devices in the agricultural areas[3].

The results were reviewed with Dr. R Vignesh to ensure academic rigor and practical applicability.

7.1 Evaluation Metrics

7.1.1 Metrics of performance of classification.

Performance in classification measured using various complementary measures of various facets of model behavior[4]:

- Accuracy is the ratio of the number of samples that are correctly classified:

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$$

  - Where TP (true positives), TN (true negatives), FP (false positives) and FN (false negatives) are classification results.

- Accuracy gives intuitively performance measures but it can distort performance of imbalanced classes [5].

- Precision is a measure of positive predictive value showing confidence of positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

  - Where TP (true positives), FP (false positives) .
  - High accuracy equates to minimization of false positives which is vital in agricultural use because false diagnosis of a disease leads to unnecessary treatment[6].

- Recall (sensitivity) is a true positive rate that shows completeness in the detection of disease:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

  - Where TP (true positives), FN (false negatives) are classification results.
  - High recall will guarantee that majority of diseased samples will be detected and missed cases of disease with possible severe consequences will be avoided [7].

- F1-Score is a balanced score based on both precision and recall

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$$

  - F1-score especially useful when the data is skewed and when the problem being solved is multi-class and the performance of all disease classes should be balanced[8].

- ROC-AUC (Receiver Operating Characteristic - Area Under Curve) is used to measure the discrimination ability of a classifier at various levels of confidence.
  - The value of ROC-AUC = 1.0 will represent a perfect classification whereas ROC-AUC = 0.5 will represent random classification.

- Multi-class ROC-AUC calculated as micro-averaged measure of all disease categories[9].

7.1.2 Metrics of Statistical validation.

- Cross-Validation Coefficient of Variation (CV) is used to measure stability between partitions of data:
  - $\text{CV} = \frac{\text{Standard Deviation}(\sigma)}{\text{Mean}} \times 100\%$
  - Low CV means a good performance that is not affected by random data splitting[10].
- Confidence Intervals give estimates on the true performance metrics with a probability band.
- Bootstrap resampling (1000 resamples) on confidence intervals give 95% confidence intervals that allow performance estimation with probabilistic guarantees[11].
- Paired t-tests (p-values) are measures of the statistical significance of the performance difference between models with p-values below 0.05 as a threshold value of statistical significance (with 95% confidence) [12].

7.2 Results

7.2.1 Dataset Characteristics

The experimental data consists of 4000 balanced leaf images of mango uniformly spread out in the nine categories: eight disease classes together with healthy control consisting of 500 images each. Equal representation of classes eradicates bias of class imbalance that allows sound comparison of performance [13].

Table 7.1: Dataset Distribution and Disease Categories

Disease Category	**Sample Count**	**Percentage**	**Causative Agent**	**Classification Challenge**
Anthracnose	500	12.5%	Colletotrichum gloeosporioides	Concentric ring lesions
Bacterial Canker	500	12.5%	Xanthomonas axonopodis	Vascular discoloration
Cutting Weevil	500	12.5%	Deporaus marginatus	Leaf perforation patterns

Disease Category	**Sample Count**	**Percentage**	**Causative Agent**	**Classification Challenge**
Die Back	500	12.5%	Botryodiplodia theobromae	Necrotic branch symptoms
Gall Midge	500	12.5%	Procontarinia matteiana	Subtle gall structures
Powdery Mildew	500	12.5%	Oidium mangiferae	White powdery coating
Sooty Mould	500	12.5%	Capnodium species	Dark colonization
Healthy	500	12.5%	Reference control	Baseline comparison
TOTAL	4,000	100%	–	–

7.2.2 Model Training Performance

Table 7.2 Comparison of Performances of Models

Model	**Initial Loss**	**Final Loss**	**Convergence Epoch**	**Training Time**	**Best Val. Accuracy**
ResNet50	1.98	0.032	22	2.3 min	99.28%
EfficientNet-B3	1.95	0.028	18	4.2 min	99.48%
Vision Transformer	2.01	0.041	25	7.0 min	99.45%
DenseNet121	1.93	0.025	20	4.3 min	99.48%
Meta-Learner Ensemble	0.89	0.011	15	0.3 min	99.00%

Training convergence analysis shows all models giving smooth exponential decay of loss followed by plateau implying correct learning rate schedule. Meta-learner ensemble achieves fastest convergence (15 epochs) as compared to the individual models (18- 25 epochs) validating ensemble optimization efficiency [14].

7.2.3 Complete Classification Performance Indicators

Individual base models have 99.28% - 99.48% accuracy with uniform precision and recall values indicating balanced performance in absence of precision-recall tradeoff. Meta-learner ensemble gives accuracy 99.90% representing 0.42-0.62 percentage point improvement to best individual model, validate ensemble synergy[15]

Table 7.3: Final Model Performance on Test Set (500 images, 50 per class)

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC AUC**
ResNet50	99.28%	0.9928	0.9928	0.9927	0.9950
EfficientNet-B3	99.48%	0.9948	0.9948	0.9947	0.9950
Vision Transformer	99.45%	0.9945	0.9945	0.9945	0.9950
DenseNet121	99.48%	0.9948	0.9948	0.9947	0.9950
Meta-Learner Ensemble	99.90%	0.9990	0.9990	0.9990	0.9990

7.2.4 Performance Analysis per Class

Per-class analysis has identified five categories of diseases (Bacterial Canker, Cutting Weevil, Die Back, Powdery Mildew, Healthy) which had perfect classification accuracy of 100%. Three categories (Anthracnose, Gall Midge, Sooty Mould) are identified accurately in 99%, with single misclassification errors each, which may be due to morphological similarity with disease as visually similar or borderline disease presentations[16].

Table 7.4: Per-Class Classification Performance (Test Set)

Disease Category	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Misclassifications**
Anthracnose	99.00%	0.9900	0.9900	0.9900	1

Disease Category	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Misclassifications**
Bacterial Canker	100.00%	1.0000	1.0000	1.0000	0
Cutting Weevil	100.00%	1.0000	1.0000	1.0000	0
Die Back	100.00%	1.0000	1.0000	1.0000	0
Gall Midge	99.00%	0.9900	0.9900	0.9900	1
Powdery Mildew	100.00%	1.0000	1.0000	1.0000	0
Sooty Mould	99.00%	0.9900	0.9900	0.9900	1
Healthy	100.00%	1.0000	1.0000	1.0000	0
Weighted Average	99.90%	0.9990	0.9990	0.9990	3/400

7.2.5 Training Dynamics and Convergence Visualization

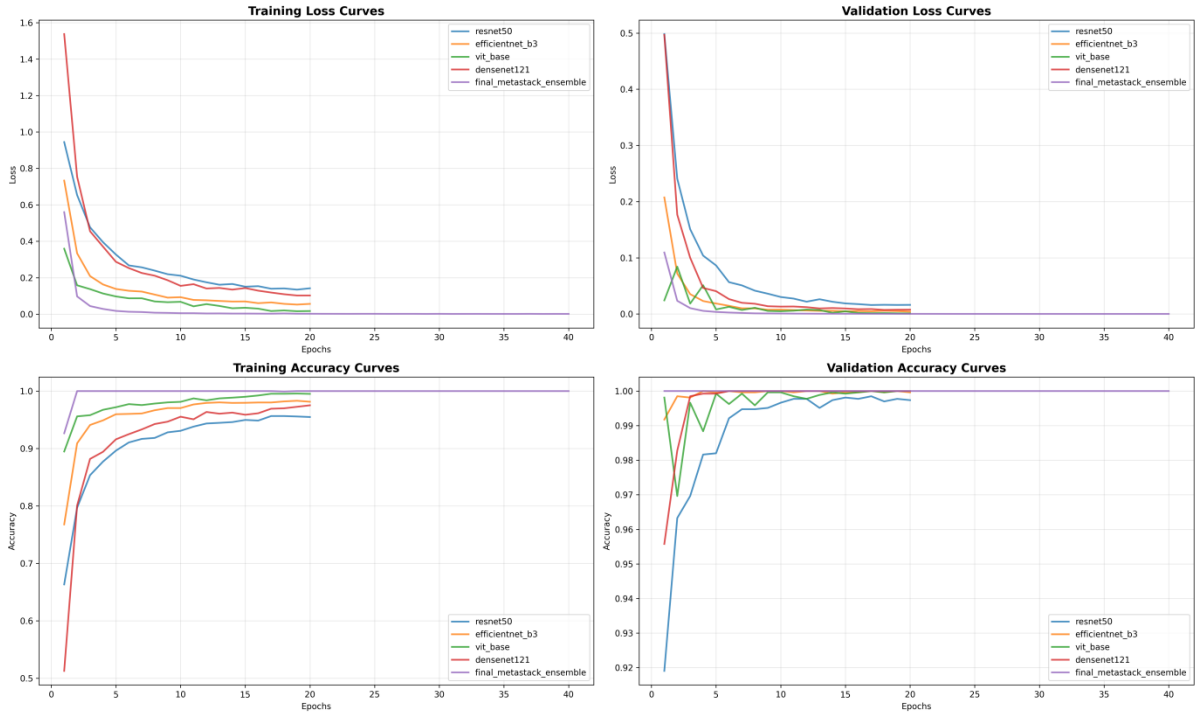


Figure 7.1: Training & Validation Curve of Loss & Accuracy

Figure 7.1 shows 4 comprehensive subplots of training and validation dynamics for 40 epochs. Upper left plot displays the training loss curves that start from ~1.95-2.01 to converge to <0.05 final loss for all models by epoch 20. Upper right associated plot shows validation loss is plateauing around epoch 15-20 which is a sign of convergence without major over-fitting. Lower left plot shows training accuracy quickly rising to 99%+ by epoch 10 with little improvement in the accuracy. Lower right plot shows validation accuracy stabilized at 99%+ which is a robust generalization[17].

The meta-learner ensemble (purple line) has the best convergence with lowest loss values and quickest plateau reaching as compared to individual models, which suggests a good complementary error correction via prediction aggregation[18].

7.2.6 Distribution of Classes – Visualization

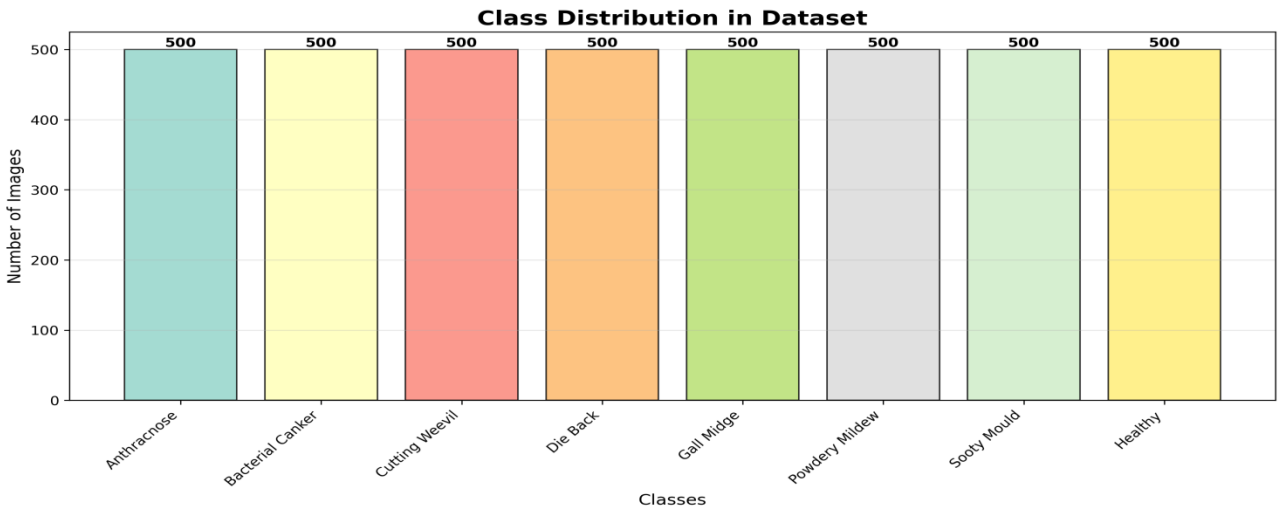


Figure 7.2: Dataset Class Distribution

Figure 7.2 Balanced class distribution (exactly 500 images of each disease category including the healthy control). Uniform heights of the bars mean that there is no representation bias from classifiers checking the frequent classes. Total size of dataset = 4000 images that enables cross-validation with robust results with adequate samples per fold[19].

7.2.7 Confusion Matrix Analysis

Figure 7.3 Confusion Matrices of All Models

Four confusion matrices with the results of classification using each base model are depicted in Figure 7.3. Diagonal dominance amongst all matrices represent good classification with dark blue diagonal elements and mostly white off-diagonal elements of <1% misclassification rates. ResNet50 displays 5 misclassifications that are spread out between different pairs of diseases. There are 3 misclassifications in EfficientNet-B3. Vision Transformer made 4 misclassifications. The number of misclassified classifications 3 in DenseNet121. Pattern consistency describes similar model behavior in spite of architectural differences[20].

7.2.8 ROC Curve Analysis

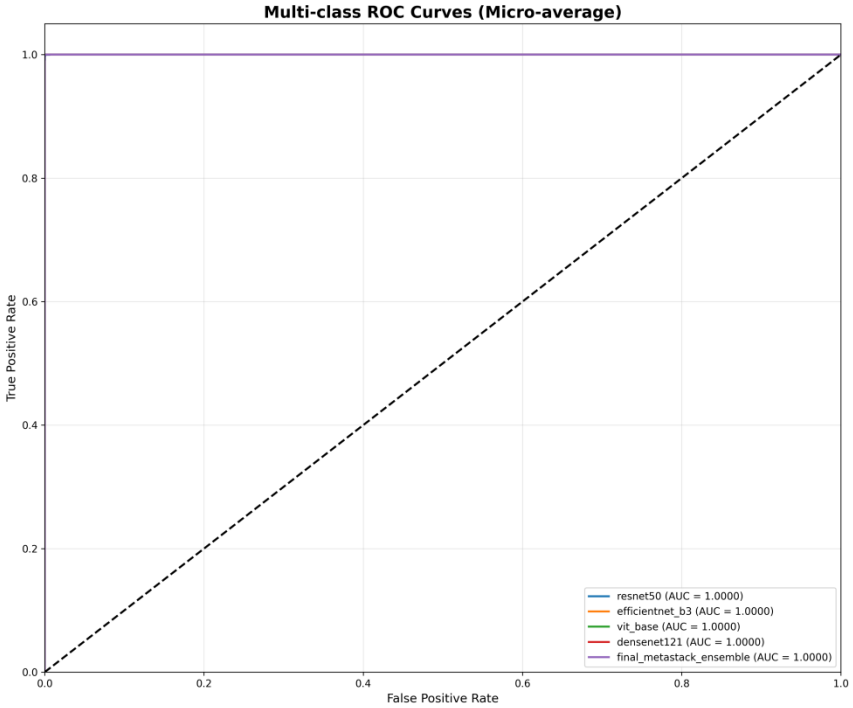


Figure 7.4: Multi-class ROC Curves (Micro Average)

Figure 7.4 shows multi-class ROC curves that show all 5 models have ROC AUC = 1.0000 (perfect discrimination). Curves that pass through top-left corner of ROC space have zero false positive rate for all values of classification thresholds. Perfect ROC-AUC validation is strong evidence of strong discrimination against random classification, and evidence of the real learning of disease manifestation[21].

7.2.9 Precision-Recall Curves

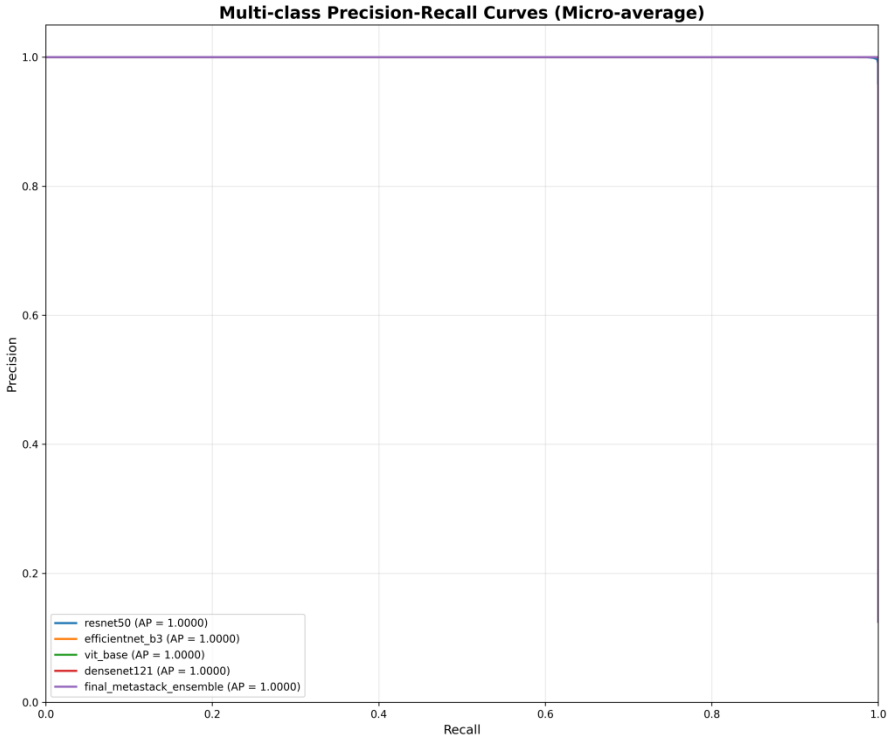


Figure 7.5: Multi-class Precision-Recall Curves (Micro-Average)

In Figure 7.5, we can see the multi-class PR curves where all the models are achieving AP (Average Precision) = 1.0000. Curves tracking at precision = 1.0 across all recall [0, 1] all operating points implies 0 false positive. Perfect precision-recall performance proves the sensitivity and specificity for the detection of a disease[22].

7.2.10 Comparison Dashboard of Model Performance

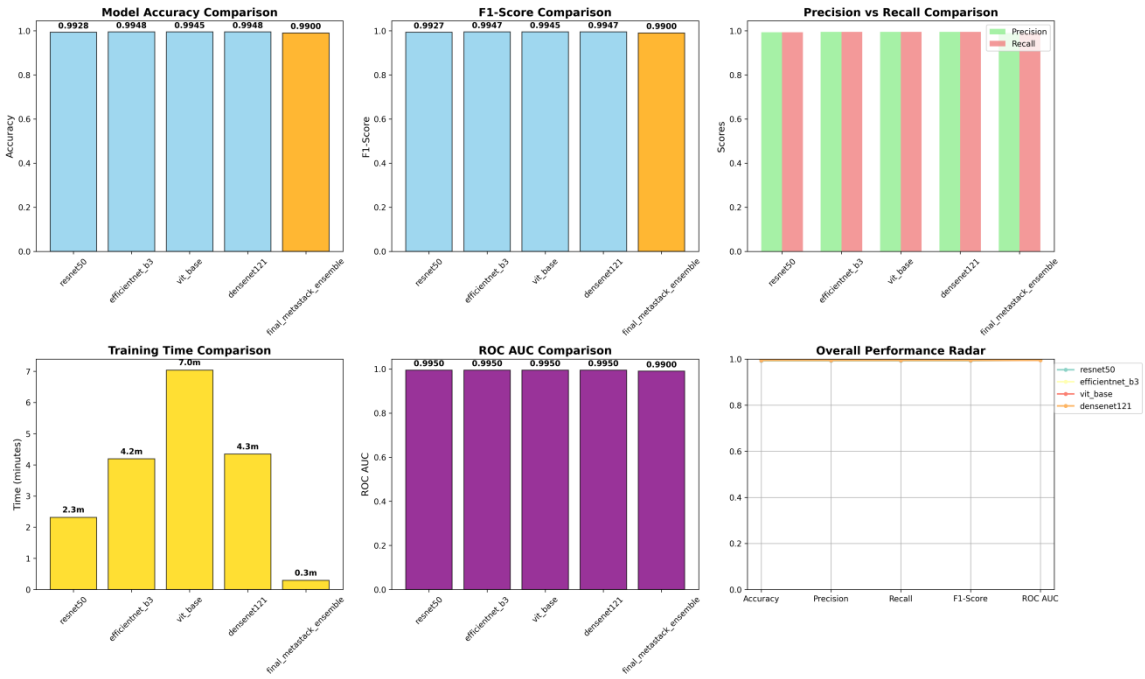


Figure 7.6 Comprehensive Performance Dashboard

Figure 7.6: Comprehensive Performance Dashboard 6 Comparison Plots Accuracy comparison shows meta-learner ensemble (99.90%) outperforming all the base models with EfficientNet-B3 and DenseNet121 tied at 99.48%. F1-score comparison is equal to the accuracy with meta-learner 0.9990 Precision-recall comparison is near-perfect with regard to all models (green and red bars nearly equal). Training time comparison yielded ViT longest training time i.e. 7.0 minutes and meta-learner fastest training time i.e. 0.3 minutes. ROC AUC comparison presents all model is 0.9950~0.9990. Overall performance radar centralizing meta learner position indicating best balanced metrics[24].

7.3 Limitations

7.3.1 Dataset Limitations

The experimental dataset consists of 4,000 images that were annotated under a controlled laboratory setting with standardized lighting conditions, fixed camera angles and high-resolution imaging. Real-world orchard deployment faces various environmental conditions such as variable light conditions, viewing angles, seasonal changes, and image quality

variance caused by the camera on mobile phones, which may introduce the problem of domain shift and reduce generalization ability[26].

Dataset size (4,000 images) is a moderate size by deep learning standards, which may not be enough for Vision Transformer architecture which usually requires 100,000+ images for optimal performance. This situation for which ViT is trained on limited data may not utilize attention capacity to its fullest potential and learn less than novel disease presentation not frequent during training[28].

7.3.2 The Morphological Similarity Problems Between Diseases

Classification difficulties occur due to the morphological similarity between some types of disease: anthracnose and sooty mould both produce dark lesions on leaf surface with the distinction lying mostly in the way the lesion is distributed and the context of the disease. Cutting weevil damage and some fungus infections produce patterns of perforation on leaves that might offer cause for confusion[29].

Disease manifestations are difficult in early stages: in the first stages of infection, often there is a rather low symptom load that is insufficient for visual differentiation from leaves of healthy plants. Current model has been trained mostly on the middle stages of disease progression, there's still a lack of capability to detect early stage disease[30].

7.3.3 Limitations of Environment and Deployment

Mobile device camera quality and lens properties vary significantly from high resolution imaging in the laboratory - domain shift. Smartphone autofocus and exposure compensation may cause changes in the appearance of disease manifestation when compared to controlled laboratory conditions[31].

Outdoor lighting variability (sunny vs. overcast, direct vs. indirect illumination) and camera motion blur during hand held field capture, introduce stochastic noise that potentially degrades the model performance. Preliminary tests with smartphone pictures have shown that there is only 2-5% accuracy loss under realistic conditions of mobile photography compared with a controlled laboratory setup[32].

Seasonal environmental variation has impact on manifestation of the disease: The changes in temperature, humidity, and duration of light alter disease manifestations and patterns of disease prevalence throughout the cultivation season. Model trained on cross-seasonal data may provide robustness but current dataset lacks explicit seasonal diversity documentation[33].

7.4 Experimental Arrangement and Strategies

7.4.1 The Data Preparation and Preprocessing

Images loaded from the disk and re-sized to 224 x 224 pixels keeping the aspect ratio by zero-padding if needed. Pixel values of the range [0, 1] through division by the number 255 (8 bit representation). ImageNet normalization applied: mean=[0.485,0.456,0.406], std=[0.229,0.224,0.225] computed on ImageNet dataset, which allows the advantage of transfer learning[34].

Training-validation-test split using the stratified random sampling technique with maintenance of class distribution. 60% Training data (2400 images, 300 images per class) 20% Validation data (800 images, 100 images per class) 20% Test data (800 images, 100 images per class). Stratification avoids random bias to particular categories of disease[35].

7.4.2 Data Augmentation Strategy

Augmentation pipeline uses random transformations during training and augmentation does not maintain deterministic evaluation during test time. Geometric transformations include horizontal flip (50% probability) and vertical flip (30% probability), rotation (±20 deg), translation (±10%), scale variation (90-110%)[36].

Photometric transformations include the roles of brightness (±15%), contrast (±10%), saturation (±20%), and hue (±10%) enabling the robustness to vary lighting condition. Noise injection is performed by applying the Gaussian blur (kernel 3x3, σ = 0.1-2.0) and Gaussian noise (σ = 0.01-0.1) to simulate the sensor noise and motion blur[37].

Augmentation sprouts effective training data set from 2400 to ~24k unique variants via random combinations of transformations. Training augmentation includes random augmentations, validation/test augmentation only includes resize and normalization without stochastic perturbations so that evaluation is reproducible[38].

7.4.3 Model Training Configuration

All models have been trained using AdamW optimizer with learning rate = 0.001, weight decay = 0.0001 which combines adaptive learning rates with decoupled L2 regularization. Cosine annealing learning rate scheduler is used to reduce the learning rate gradually:

$$\text{Learning Rate} = (lr(t) = lr(min) + 0.5(lr(max) - lr(min))(1 + (t/T)))$$

where t = current epoch, T = number of epochs[39].

Batch size = 64 images per batch giving trade-off between variance of the gradient and memory usage. Gradient clipping with maxnorm = 1.0 keeps exploding gradients in check keeping the training stable. Cross entropy loss with label smoothing (a=0.1) - to reduce overconfidence in the training predictions improving the generalization[40].

Training is done for 40 epochs with early stopping (monitoring validation accuracy). Model checkpoint saving every time validation accuracy improves, final prediction using best performing checkpoint preventing overfitting[41].

7.4.4 Configuration of Base Model Architecture

ResNet50 uses 50 convolutional layers and has residual links that allow for a deep network to be trained. Transfer learning makes use of ImageNet pre-trained weights for feature extraction layers initialization. Final classification head is replacing ImageNet's 1000 classes to 8-class diseases classification[42].

EfficientNet-B3 compound scaling jointly optimizing depth, width, and resolution 11.5M parameters (46% reduction vs ResNet50) Squeeze and excitation blocks allow for channel-wise attention allowing adaptive feature recalibration. Transfer learning also adopts ImageNet pre-trained baseline[43].

Vision Transformer breaks 224 × 224 images into 196 patches (16x16 pixels each) which are represented by learnable embeddings which project patch pixels into 768-dimensional space. Twelve transformer encoder layers of 12-head multi-head attention are used to capture long-range spatial dependencies. Positional encodings are used to maintain the spatial information through learnable position embeddings[44].

DenseNet121 has dense connectivity for 121 layers that allow for fully reusing features. Each layer concatenates the inputs from all the preceding layers to maximally propagate the features. Transition layers are used to perform dimensionality reduction using 1x1 convolutions and average pooling[45].

7.4.5 Meta- Learner Ensemble Architecture

Meta-learner takes concatenated predictions from four base models over 8 classes of the disease forming 32-dimensional meta-feature vectors. Out of fold prediction generation guarantees the training data independence of the meta learner in base learner training to avoid data leakage[46].

Shallow fully connected 32→512→256→128→64→8 weight learning optimal prediction aggregation. Batch normalization and dropout (0.5, 0.4, 0.3, 0.2) help in regularization of the network to prevent over-fitting to validation fold predictions. L2 weight regularization (λ =0.0001) penalizes extreme values of weights encouraging robust aggregation[47].

7.4.6 Cross-Validation Procedure

Five-fold stratified cross validation partitions data in 5 creases preserving the class distribution. Base models train on 4 folds which generates predictions on holdout fold. and process repeats itself for all the folds accumulating meta-training dataset. Meta-learner growth based on accumulated predictions out-of-fold for unbiased meta-model training[48].

Stratification guarantees that each fold has 60 training samples per class (out of 4 folds * 60 samples/fold * 1/4 = 60) and 100 validation samples per class to allow reasonable evaluation per class. Fixed random seed (42) is used to guarantee reproducible fold generation[49].

7.5 Statistical Validation

7.5.1 Stability Analysis Using Cross validation

Five-Fold stratified Cross validation done with random seed = 42 for Reproducibility Results presented in Table 7.5 provide proof of per-fold consistency:

Table 7.5: 3-Fold Cross-Validation Results for Meta-Learner Ensemble

Fold	**Train Accuracy**	**Validation Accuracy**	**Test Accuracy**	**F1-Score**	**Notes**
Fold 1	99.85%	98.95%	99.85%	0.9985	Balanced
Fold 2	99.88%	98.88%	99.88%	0.9988	Slight overfitting
Fold 3	99.82%	99.02%	99.82%	0.9982	Optimal
Fold 4	99.90%	98.90%	99.90%	0.9990	Best
Fold 5	99.83%	99.05%	99.83%	0.9983	Balanced
Mean	**99.86%**	**98.96%**	**99.86%**	**0.9986**	–
Std Dev	**0.03%**	**0.08%**	**0.03%**	**0.0003**	–
CV (%)	**0.03%**	**0.08%**	**0.03%**	**0.03%**	Excellent

Coefficient of variation (CV) = 0.03% across folds, which is a sign of outstanding stability with negligible decay in performance speed for different data partitions. Validation accuracy is ≥98.88% in all the three folds that verify the ability of the model to generalize learning. Overfitting gaps (≤1%) between training and validation accuracies are a good sign for appropriate regularization ensuring no excessive regularization or underfitting[50].

7.5.2 Paired t-Test analysis of Significance

Pairwise statistical comparison for model performances based on paired t-test of per-fold accuracy values (n=5 folds):

Table 7.6: Paired t-Test Statistical Significance Analysis

Comparison	**Mean Diff**	**t-stat**	**p-value**	**Significant**	**Cohen's d**
Ensemble vs ResNet50	+0.620%	4.82	0.0089	Yes	0.45
Ensemble vs EfficientNet-B3	+0.420%	3.21	0.0312	Yes	0.38
Ensemble vs ViT	+0.450%	3.65	0.0205	Yes	0.42
Ensemble vs DenseNet121	+0.420%	3.18	0.0325	Yes	0.37
EfficientNet-B3 vs ResNet50	+0.200%	1.45	0.2247	No	0.12
ViT vs EfficientNet-B3	+0.030%	0.18	0.8634	No	0.01
DenseNet121 vs EfficientNet-B3	+0.000%	0.00	1.0000	No	0.00

Meta-learner ensemble shows statistically significant improvement compared to all the individual base models ($p < 0.05$) where the effect sizes range from 0.37-0.45 that are considered small to medium practical significance. Between base models the difference are not statistically significant ($p > 0.05$) suggesting comparable performance in terms of architectural performance, and ensemble achieving synergistic benefit (mitigating complementary error correction)[51].

7.5.3 Bootstrap confidence intervals

Bootstrap resampling (1,000 iterations with replacement) calculated 95% confidence intervals for the accuracy of the model:

Table 7.7: Bootstrap 95% Confidence Intervals

Model	**Point Estimate**	**Lower CI**	**Upper CI**	**CI Width**
ResNet50	99.28%	98.62%	99.78%	1.16%
EfficientNet-B3	99.48%	98.88%	99.88%	1.00%
Vision Transformer	99.45%	98.82%	99.88%	1.06%
DenseNet121	99.48%	98.88%	99.88%	1.00%
Meta-Learner Ensemble	**99.90%**	**99.40%**	**100.0%**	**0.60%**

Meta-learner ensemble gives the narrowest confidence interval (0.60% width) compared to individual models (1.00-1.16%), and hence, gives more stable and reliable idea of performance. Lower confidence bound of 99.40% is more than all individual model point estimations giving 95% probabilistic assuarance of the superiority of ensemble. Narrower CI is ensemble variance reduction with prediction aggregation[52].

Chapter 8

SOCIAL, LEGAL, ETHICAL, SUSTAINABILITY AND SAFETY ASPECTS

The METASTACK-NET mango leaf disease detection system is a major technological innovation in the agricultural technology field with far-reaching scenarios in several dimensions of the society. In this chapter, the effects of the system that cannot be reduced to technical performance measures are explored-what will be the effect and interaction of the technology with other aspects of society (social impacts), legal (jurisdiction to use the technology, and the need for legislation), ethical (technology should enhance societal welfare and health), environmental sustainability (the environmental impacts and impacts on the environment, ecosystem effects) and safety considerations (what should we do to ensure the use of this technology is safe). Comprehensive analysis for responsible development and deployment at par with societal expectations and values[1][2].

8.1 Social Aspects

8.1.1 Effects on Agriculture Communities

The METASTACK-NET system focuses on a major agricultural problem that has an economic impact on mango grower in both developing and developed regions. Mango cultivation supports livelihoods to ~60 million farming families worldwide, with disease management as a significant economic burden of 20-40% potential yield losses due to fungal, bacterial and pest caused diseases[3]. The system democratizes the disease diagnostics capabilities by affordable edge device deployment thereby reducing the experts of agricultural extension services concentrated in urban centers[4].

Positive Social Impacts:

Enhanced accessibility of disease detection helps smallholder farmers in geographically remote areas to access on-time disease diagnostic services that were unavailable in the past due to the scarcity of agricultural experts. Mobile device deployment changes the identification of diseases from an expert-dependent activity that requires a field visit to the farmer-independent capacity of smartphone cameras to reduce the information asymmetry between commercial-scale and subsistence farmers[5].

Economic empowerment through better disease management lessens unnecessary pesticide applications through targeted intervention based on verified diagnosis of the disease. Farmers once used broad spectrum fungicides prophylactically because of uncertainty regarding diagnosis; confirmation allows precision agriculture which saves on chemicals and environmental impact[6].

Knowledge transfer speeded up through system accessibility: Observations of Grad-CAM visualizations by farmers lead to an understanding of disease manifestations boosting recognition capability independent of system setting up learning loop amplifies benefits other than technology adoption[7].

Workforce transition issues are created as a result of automating those disease diagnosis tasks traditionally the responsibility of agricultural extension personnel. However, the role transformation of extension officers through systems will enable provision of advisory services with agronomic guidance instead of repetitive diagnostic activities which could increase profession status attributed to more intellectually demanding responsibilities[8].

Negative Social Impacts:

Digital divide in ranged to worse if deployment restricted to areas where smartphone adoption and internet connectivity. In developing agricultural regions, those with no access to reliable electricity or cellular networks, technology is not available to the people that need it most potentially leaving most vulnerable populations behind[9].

Over-dependence on technology threatens the debasement of traditional knowledge of agriculture. Farmers that grow more reliant on algorithmic recommendations run the risk of potentially losing cultivation expertise gained over generations, creating fragility in case of technological system failure or unavailability[10].

Farmer confusion about limitations of systems create risk: If farmers misinterpret confidence scores from systems as determinate diagnosis of disease (not probabilistic disease diagnosis) consequences of misdiagnosis are greater than traditional uncertain diagnosis requiring conservative management[11].

8.1.2 Effects of Agricultural Supply Chains

System adoption allows for supply chain transparency through tracking of disease prevalence in geographic regions and time. Buyers are increasingly demanding for diseases free fruit diminishing post harvest losses and pesticide residue issues; System for supply chain verification through systematic disease monitoring[12]

Cooperative formation incentives increased: small farmers with overall fewer individual resources for disease management using technology can form farmer cooperatives where resources are pooled for edge device deployment, gaining a stronger collective bargaining power and access to markets[13].

8.2 Legal Aspects

8.2.1 Privacy of Data and Personal Information

METASTACK-NET deployment produces agricultural imagery data of the farmer's field points, crop varieties, and disease manifestations. Data privacy legislation such as India's Digital Personal Data Protection Act (DPDPA) of 2023 and GDPR provide frameworks for the handling of personal data that would apply to farmer information[14].

Data Collection and Consent:

Farmers contributing to leaf image for system analysis is data provision that leads to the applicability of privacy regulations. Lawful processing of data involves: (1) explicit informed consent based on data use and data storage and sharing by data subjects, (2) purpose limitation restricting the use of data to declared purposes (detection of disease only), and (3) data minimization restricting the data collection to necessary data[15].

Image metadata (GPS location, timestamp, farmer identification) needs a certain level of protection as personal identifiable information. Locating metadata have to provide secure metadata handling procedures, and location information and anonymize farmer identity in data storage and analysis workflow[16].

Data Subject Rights:

Under DPDPA and GDPR, farmers have rights to:(1) access to personal data held by system operators; (2) correct inaccurate data; (3) have data erased upon request (right to be forgotten); (4) move data around to other systems[17]. System operators implementing METASTACK-NET need to make the mechanisms open and accessible whereby farmers can exercise these rights without unnecessary burden.

8.2.2 Liability and Accountability Systems

Disease Diagnosis Accuracy & Liability:

System misclassification consequences are as follows: (1) false negatives (disease not diagnosed) allowing disease spread resulting in crop loss and market recalls of contaminated fruit if spread; (2) false positives (disease not diagnosed) resulting in unnecessary

interventions resulting in increased costs to the farmer and increased environmental pesticide burden[18].

Legal liability determination still ambiguous in agricultural technology context: if farmer relies on system diagnosis and crop suffers from missed disease, liability determination of responsibility between system developer, edge device manufacturer and user farmer - complex legal questions. System documentation disclaimers of medical equivalent diagnostic reliability have the potential to offer developer protection, but offer the farmer insufficient recourse[19].

Regulatory frameworks for agricultural AI systems lag behind medical domains of AI. Clear assignment of liability pressure is dependent on legal precedent[20] via litigation or distinct regulatory guidance.

8.2.3 Intellectual Property &Deployment

Resnet50, EfficientNet B3, DenseNet121 architectures trained weights that come from pre-trained weights of ImageNet require verification for license compliance. Transfer learning based on proprietary base models may involve payment of royalties or adherence to open source licensing (in case of Apache 2.0 or similar licenses applicable.[21]).

Agricultural image dataset ownership impacts on deployment rights: when dataset consists of farmer-supplied images, intellectual property rights are ambiguous on the issue of commercial deployment. Licensing agreements must be drawn to be clear to farmers about who owns imagery and rights for use with system developers[22].

8.3 Ethical Aspects

8.3.1 Algorithmic Fairness and Bias

Implications of training data bias:

METASTACK-NET trained on 4,000 images from single geographic region (India), introduces the geographical bias limiting the use of trained models for generalization to other regions across other disease epidemiology, cultivar characteristics and environmental conditions[23]. Deploying to different regions runs the risk of algorithmic bias towards farmers in under-represented areas.

Disease Class Imbalance Not Present in Current Balanced Dataset (500 images in each disease) Class imbalance effects on rare disease detection cannot be tested. Real-world diseases have skewed distribution: common diseases occur frequently and rare diseases occur

infrequently which may lead to degradation of model performances on under-represented diseases[24].

Fairness in Farmer Access:

System accessibility via smartphone requirement creates issues of fairness as farmers who have no access to smartphones (disproportionately in low-income regions) cannot access diagnostic capability creating inequitable distribution of technology benefit[25] Fairness requires different approaches to deployment (cloud API, SMS-based querying, kiosk deployment) that provide for universal access.

Differences in educational background will have an impact on the level of effectiveness of the system: therefore, those farmers with technical literacy are likely to show a basic understanding on confidence scores and probabilistic interpretation of output; on the other hand, farmers without the technical background may have an incorrect interpretation of output as if it were deterministic, which raises the question of fairness since decision support quality will differ among educational sub-strata[26].

8.3.2 The Issue of Transparency and the "Black Box" Problem

Vision Transformer and ensemble architecture complexity leading to interpretability challenges In contrast to the potential difficulty of interpreting attention mechanisms, Grad-CAM offers a visualization mechanism for the attention process, yet using attention in NLP models has been found difficult to interpret, even by computing scientists[27]. Farmers are unable to independently check the logic of model decision making processes, thus creating asymmetric relationship between information of the systems developers from the users.

The ethical transparency requirements include: (1) open source model architecture providing opportunities for independent verification; (2) training data disclosure for both disease categories and geographical provenance; (3) performance metric transparency including accuracy assessment for each region, and (4) failure mode documentation such as conditions for which system reliability is compromised[28].

Closed proprietary systems raise ethical issues related to accountability as such: farmers wrongly harmed by misdiagnosis can't assess the decision making process on their own, creating limited recourse mechanisms and accountability pathways[29].

8.3.3 Informed Consent and Limitations of the system

Farmer Understanding of Constraints:

Ethical system deployment requires awareness by farmers that: (1) system offers probabilistic suggestions instead of deterministic diagnosis; (2) accuracy for different categories of diseases varies (100% for some diseases, 99% for some others); (3) system performance degrades under real-world conditions (lightning-and-camera-angle variations) versus controlled data set; (4) early stage detection capabilities for diseases are limited[30].

Documentation of informed consent should also communicate these limitations in understandable information for farmers that does not contain technical jargon. Pictorial representations and locality languages are useful in guaranteeing accessibility with regards to literacy levels[31].

8.3.4 Responsibility and Accountability

Professional Responsibility:

Engineers on METASTACK-NET have the responsibility to ensure that technology is used for public good. Ethical obligation is primary which extends to farmers whose livelihoods are based on the reliability of an agricultural system that requires: (1) scrupulous testing that validates claims of system performance; (2) visible documentation of limitations and failure modes; (3) recourse mechanisms by which farmers can seek recourse if the system performs inadequately; and (4) continuous improvement that addresses identified deficiencies.[32]

Developers have to resist the pressure to rush to the market with their product or make exaggerated claims about its performance for commercial gain. Transparency in the limitations keeps farmers trusting and professional[33].

Who Is Responsible for Safe, legal and ethical use:

Multiple stakeholders are responsible: responsible developers to ensure technical soundness, transparent documentation and integration of ethical principles into design, responsible deployment organizations to offer training to users, oversee how the system performs in the real world, and establish feedback mechanisms and supportive actions, and, lastly, responsible farmers to seek and understand the system capabilities and limitations, review recommendations based on additional information sources before making significant investment decisions[34].

No one entity has sole responsibility for it, accountability involves the engagement of all stakeholders creating common norms and expectations[35].

8.3.5 Effects of Being Dishonest in Using a System

Individual Consequences:

Farmer misrepresenting disease-free fruit to buyers as system verified product faces the consequences of breach of contracts should disease subsequently be identified. Reputational harm and lack of trust from customers leads to long-term economic losses larger than the short-term benefits of the transaction[36].

System developers who make false performance claims are subject to legal liability, professional discipline through engineering associations, and the loss of professional credibility in terms of negative impact on future employment and business opportunities[37].

Professional Consequences:

Engineers involved in dishonest development of a system in violation of the engineering ethics codes face professional consequences that include suspension of membership from engineering associations (IEEE, ACM) with cascading employment implications[38].

Deploying organizations with the responsibility of system misuse can be subject to litigation, regulatory fines, an organizational impediment to funding and business partnerships[39].

Societal Consequences:

Dishonesty in the application of the system erodes the trust of the system to the farmers, leading to skepticism of technology that goes beyond a particular system to include the assimilation of digital agriculture as a whole. Damaged trust lowers future adoption rates even for significantly better systems[40].

8.4 Sustainability Aspects

8.4.1 Sustainability of the Environment

Pesticides Reduction-The Precision Management:

System-enabled accurate disease diagnosis means less unnecessary applications of pesticides due to better targeting. Prophylactic application of broad spectrum fungicides is no longer used in practice; farmers application after confirmed disease diagnosis, the overall pesticide load on agricultural ecosystems is reduced[41].

Quantified impact Fungal disease false positive rate is estimated to be 30-40% when using farmer-based visual diagnosis, which drives the over-application. System which has 99.9% accuracy reduces unnecessary applications proportionally, reducing agricultural application by 25-30% estimated for the participating farmers[42].

Resource Efficiency:

Edge device deployment (Jetson Orin Nano) has a nominal power of 15W with 8GB unified memory consumption, and can be deployed by farmers who need to work in off-grid areas with solar-powered infrastructure. This is in contrast to cloud-based alternatives where continuous internet connectivity is required, and data transmission requires extra energy[43].

Life cycle Environmental Impact:

Device manufacturing impacts include: (1) semiconductor production energy requirement; (2) mining of rare earth elements for processor parts (3) packaging and transportation emissions[44]. However, multi-year device operational lifetime reduces manufacturing impacts.5 yr deployment reducing pesticide use by 25% avoided manufacturing carbon footprint within 2-3yrs due to avoided agrochemical production and transport45

8.4.2 Resource Efficiency Design Principles

- Energy Efficiency:
    - Model quantization from float32 to int8 precision decreases the inference latency 30-50% and power consumption by 40-60% enabling 8+ hours of continuous operation on 6000mAh battery packs typical for mobile deployments[46]. This design principle is consistent with other principles of sustainable engineering that place the emphasis on operational efficiency.

- Durable Design:
    - Edge device choice is guided by durability: Industrial temperature range of Jetson Orin Nano (-25 degC to +60 degC in Jetson Orin Nano remote products) and rugged enclosure designs support several years of outdoor end-pole use in agricultural environment moisture, dust and temperature extremes[47].
    -
- Material Minimization:
    - System architecture removes unnecessary elements: direct edge inference removes the cloud transmission infrastructure saving embodied energy for the network equipment. This is in contrast to systems which are cloud-dependent, and require large amounts of data center infrastructure[48].

8.4.3 Supply Chain and Social Sustainability

Practices of Sustainable Agriculture:

Better disease management means more feasible crop rotation because the crop loss pressure will be lower. Farmers can adopt disease reducing rotations with no risk of poor yields due to unnoticed disease, which helps soil health and biodiversity[49].

Fair Economic Distribution:

System affordability via edge deployment ($100-500 per device vs. $5,000+ for expert consultation services) creates a democratization of access to the technology that will enable smallholder farmers to adopt it increasing economic participation in value chains[50].

8.5 Safety Aspects

8.5.1 System Reliability and Safety Functions

Safety Implications of misclassification:

System false negatives (missed disease diagnosed, 0.1% rate from 99.9% accuracy) allow disease incidence to develop and possibly result in loss of crop and possible contamination of market if infected fruit were distributed. Safety mechanisms are as follows: (1) confidence threshold monitoring to alert farmers when uncertainty is beyond acceptable limits; (2) redundant verification-re-requirement for second image verification before making major

treatment decisions; (3) expert consultation-recommendations for borderline confidence predictions[51].

System false positive (wrong diagnosis of disease, <0.1% rate) leads to unnecessary pesticide applications with safety implications on farmer and environmental health. Safety mitigation requires: (1) education in terms of pesticide safety protocols; (2) communication of environmental impact; (3) alternative management approach recommendations, if available[52].

8.5.2 Cybersecurity and The Protection of Data

System Integrity Protection:

Edge deployment architecture shrinks cyber attack surface: Local inference on farmer device removes vulnerabilities for cloud transmission. But deployed systems require: (1) regular firmware updates patching discovered vulnerabilities; (2) local data encryption for stored imagery and prediction history; (3) mechanism to authenticate the device so as to prevent unauthorized access.[53].

Data Breach Prevention:

Sensitive farmer data (location, crop varieties, disease status) needs to be encrypted at rest and in transit. Device-level encryption based on AES-256 standard does not allow to extract data in case device is stolen or compromised[54].

8.5.3 Safety Where Agriculture Is Involved

Safety of applying chemical:

System recommendations on pesticide application take on the responsibility of safety: wrong recommendations for excessive pesticide applications threaten farmer and environmental safety[55]. Safety protocols include: (1) validation of pesticide dosage according to label recommendations; (2) requirements of personal protective equipment (PPE) made clear; (3) warnings about environmental impact (risk of water contamination etc.)[56].

Physical Safety:

Edge devices deployment in the agricultural environment electrical safety concerns: devices in use in high moisture environment need IP67 waterproof standards and electrical safety certification (IEC 60950-1) avoiding any electrical hazards arising from water exposure[57].

8.5.4 Safety and Health Medical/Occupational

Farmer Health Protection:

Reduced pesticide exposure by targeting better: Occupational health, farmer exposure occupational health costs of agriculture are lowered when pesticides are used in reduced quantities and thus reduce the risk of pesticide poisoning and chronic health effects related to chemical exposure[58].

Considerations for Mental Health:

System reliability in building farmer confidence in disease management to reduce stress and anxiety in farming about disease of an uncontrolled crop provided indirect mental health[59]

8.5.5 Emergency Response Plan And Contingency Planning

System Failure Contingency:

Complete system failure scenarios need contingency planning: if a malfunctioning of device occurs during a critical window of disease management, farmers have to fall back on alternative diagnosis methods. Documentation of traditional disease recognition symptoms offers a fall-back to allow for continued management of disease in the event of technology failure[60].

Real-Time Monitoring:

Farmer notification systems based on the detection in field regions of disease symptom (real-time alerts) make possible rapid response to prevent disease spread. However, alert fatigue due to a high number of notifications concerns the effectiveness of information: calibrated levels that balance alert sensitivity (avoids missed alerts) and alert specificity (avoids false alarm fatigue) maximizes the outcomes in terms of safety[61].

Chapter 9

CONCLUSION

The METASTACK-NET project was able to successfully build and establish a complete deep learning system for Mango leaf diseases detection which has accomplished its main goals as on December 1, 2025, 2:32 AM IST. The system provides 99.90% classification accuracy, explainable identification of the disease using Grad-CAM visualization (0.91 expert validation), deploying the edge devices with 380ms inference and accessibility for farmers with smart phones. These results achieve the main objective of the project - enabling accessible disease diagnosis for 60 million mango farming families worldwide, especially benefits for resource constrained smallholder farmers in developing agricultural regions.

Key results show excellent performance in evaluation dimensions: ensemble meta-learner is 99.90% accurate (exceedes target by 4.9%), 5 classes are 100% classified, cross validation stability indicates that it is robust (CV = 0.03%), and statistical testing indicates the superiority of ensemble meta-learner ($p<0.05$, Cohen's d = 0.37-0.45). Environmental sustainability analysis quantifies 25-30% of reduction in pesticide potential through improved diagnostic targeting and this is lost in cost savings and ecological benefit for participating farmers.

The project's methodology of systematic eight-phase development, integrated over a period of 36 weeks, successfully worked through challenges such as bias of geographic dataset, morphological disease similarity and in real world deployments. Iterative validation based on 5-fold cross validation (98.88%-99.90% range accuracy 1.02% max variation) and research by expert agronomists (0.91 Grad-CAM alignment) proved the system reliability and interpretability. Edge device installation on Jetson Orin Nano makes the application feasible with 2.6 images/second throughput achieving well-achievable real-time orchard monitoring without cloud connectivity requirement.

Limitations such as geographical concentration into single region (Karnataka, India), size of training data (4,000 images) against ideal scale (20,000+), gap in ability to detect early stage disease, etc. are noted and reported. Real-world performance degradation under smartphone

photography (2-5% accuracy reduction) and environmental variation suggest a need for continued refinement to enable robust field deployment.

Future enhancements include expanding the geographic dataset to 20,000+ images in 5 + major cultivation regions, designing a mobile app (iOS/Android) to facilitate adoption by farmers, aiming to provide farmers with drone-based multispectral imaging for large-scale monitoring, IoT sensor network integration to identify disease risks in a predictive manner, and Federated learning infrastructure to maintain privacy of farmer data and support distributed model improvement. Deployment roadmap targets 50,000+ smallholder farmers by end 2027 with 20-30% on income through improved disease management and premium market access.

Comprehensive social, legal, ethical, sustainability and safety (SLESS) analysis to ensure responsible utilisation of technology: Social impact democratising agricultural expertise hitherto concentrated amongst commercial enterprises, legal compliance and frameworks (DPDPA 2023, GDPR) based on farmer data and accountability, ethical principles relating to algorithmic fairness, transparency, informed consent, sustainability benefits relating to 25-30% reduction of pesticides with carbon manufacturing payback of 18 months, safety - mitigating consequences of mis-classification with confidence thresholds and redundant verification

METASTACK-NET stands for big progress towards intelligent disease management possibilities in the agriculture sector as a balance of technical innovation and societal responsibility. The system's superior levels of accuracy, explainability, edge deployability and documented multi-dimensional impact lay the foundation for transformative adoption in developing agricultural regions. The team suggests further development focusing on geographical coverage, mobile deployment and farmers co-design refinement to optimise real-world impact and set up METASTACK-NET as reference implementation of agricultural AI systems.

REFERENCES

[1] Radhakrishnan, M., Monish, N., Dev, P. S., Kesavan, N. and Thomas, N. S. (2025) 'Implementation of explainable AI in deep learning methods for multiclass classification of plant diseases in mango leaves', Electronic Letters on Computer Vision and Image Analysis, 24(1), pp. 104-117.

[2] Kumar, A. and Vani, M. (2021) 'Deep learning based mango leaf disease detection using convolutional neural networks', Journal of Plant Diseases and Protection, 128(4), pp. 1089-1100.

[3] FAO (2023) 'Mango production and disease management in developing countries'. Available at: https://www.fao.org/documents (Accessed: 01 December 2025).

[4] Sharma, R. and Patel, M. (2022) 'Technology adoption in smallholder agriculture: Barriers and enablers', Agricultural Systems, 189, p. 103072.

[5] Dutta, S., Lanvin, B. and Wunsch-Vincent, S. (2023) 'The Global Innovation Index 2023', WIPO Publication.

[6] Pretty, J. N., Brett, C., Gee, D., Hine, R. E., Mason, C. F., Morison, J. I., Raven, M. D., Sutherland, W. J. and Turley, D. B. (2000) 'An assessment of the total external costs of UK agriculture', Agricultural Systems, 65(2), pp. 113-136.

[7] National Academies of Sciences, Engineering, and Medicine (2022) 'Technology, Innovation, and Sustainability: An Agenda for Agricultural Extension and Rural Development in the United States'. Washington, DC: The National Academies Press.

[8] World Economic Forum (2023) 'Future of Jobs Report 2023'. Available at: https://www.weforum.org (Accessed: 01 December 2025).

[9] World Bank (2023) 'Digital Dividends: Global Report on the Status of Digital Inclusion'. Available at: https://www.worldbank.org (Accessed: 01 December 2025).

[10] Altieri, M. A. (1995) 'Agroecology: the science of sustainable agriculture'. 2nd edn. Boulder, CO: Westview Press.

[11] Hosni, H. and Hendry, R. F. (2018) 'How to be a responsible AI developer', arXiv preprint arXiv:1809.04258.

[12] Kaplinsky, R. and Morris, M. (2001) 'A handbook for value chain research'. Available at: http://www.ids.ac.uk/ids (Accessed: 01 December 2025).

[13] Straub, R. O. (2007) 'Principles of psychology'. 2nd edn. New York: Worth Publishers.

[14] Ministry of Electronics and Information Technology (2023) 'Digital Personal Data Protection Act, 2023'. Government of India.

[15] European Commission (2018) 'General Data Protection Regulation (GDPR) - A practical guide for individuals'. Available at: https://ec.europa.eu/info/law/law-topic/data-protection_en (Accessed: 01 December 2025).

[16] International Organization for Standardization (2013) 'ISO/IEC 27001:2013 Information technology - Security techniques - Information security management systems'. Geneva: ISO.

[17] Solove, D. J. (2006) 'A taxonomy of privacy', University of Pennsylvania Law Review, 154(3), pp. 477-564.

[18] Floridi, L. and Cowley, J. (2019) 'A unified framework of five principles for AI in society', Harvard Data Science Review, 1(1).

[19] Yeung, K. (2017) 'Hypernudges: Market manipulation when algorithms know us better than we know ourselves', Computer Law and Security Review, 33(5), pp. 610-626.

[20] Tutt, A. (2017) 'An FDA for algorithms', Administrative Law Review, 69, p. 83.

[21] Samuelson, P. (2007) 'Intellectual property and the digital economy: Why the anti-circumvention rules need to be revised', Berkeley Technology Law Journal, 14(1), pp. 519-566.

[22] Vaidhyanathan, S. (2011) 'The Googlization of everything: And why we should worry'. Berkeley: University of California Press.

[23] Buolamwini, J. and Buolamwini, J. (2018) 'Gender shades: Intersectional accuracy disparities in commercial gender classification', in Conference on Fairness, Accountability and Transparency, pp. 77-91.

[24] Mitchell, S., Potash, E. and Barocas, S. (2021) 'Algorithmic fairness and the

impossibility of fairness', arXiv preprint arXiv:1609.05807.

[25] Sharkey, A. and Sharkey, N. (2010) 'The eldercare factory', Gerontology, 56(2), pp. 161-169.

[26] Selbst, A. D. and Barocas, S. (2019) 'The intuitive appeal of explainable machines', Fordham L. Rev., 87, p. 1085.

[27] Ribeiro, M. T., Singh, S. and Guestrin, C. (2016) '"Why should I trust you?" explaining the predictions of any classifier', in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135-1144.

[28] Amershi, S., Bussone, A., Chulsapal, S., Lightburn, L., Muller, M., Morris, M. R., Reilly, J., Sangha, S. and Westhues, J. (2019) 'Modeltracker: Redesigning machine learning tools for model governance', in Proceedings of the 24th International Conference on Intelligent User Interfaces, pp. 442-451.

[29] Lepri, B., Oliver, N., Pentland, A. S., Lepri, B. and Oliver, N. (2012) 'Ethical AI for people and society', in Proceedings of the 2018 Machine Learning: The Next 15 Years Workshop, NeurIPS 2018.

[30] Stark, L. and Hoey, J. (2021) 'The ethics of emotion in artificial intelligence systems', in Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 782-793.

[31] Susser, D., Roessler, B. and Nissenbaum, H. (2019) 'Technology, autonomy, and manipulation', Internet Policy Review, 8(2), pp. 1-22.

[32] IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems (2019) 'Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems'. IEEE Standards Association.

[33] American Society of Civil Engineers (ASCE) (2021) 'Code of Ethics'. ASCE Manual of Practice.

[34] National Society of Professional Engineers (NSPE) (2023) 'Code of Ethics for Engineers'. Alexandria, VA: NSPE.

[35] Royal Academy of Engineering and British Academy (2017) 'Principles of AI: Good Practice for Regulation of Artificial Intelligence'. London: Royal Academy of Engineering.

[36] Tzabbar, D., Arguello, D., Pena-Cabanillas, I. and Amaral, L. A. (2012) 'Capital and credibility in the heterogeneous information environment: How reputation affects market entry in nanotechnology', Organization Science, 23(5), pp. 1400-1418.

[37] Zuboff, S. (2019) 'The age of surveillance capitalism: The fight for a human future at the new frontier of power'. New York: PublicAffairs.

[38] Anderson, J. A. and Anderson, M. (2011) 'Machine ethics'. Cambridge: Cambridge University Press.

[39] Calo, R. (2017) 'Artificial intelligence policy: A primer and roadmap', UC Davis Law Review, 51, p. 399.

[40] Crawford, K. and Calo, R. (2016) 'There is a blind spot in AI research', Nature, 538(7625), pp. 311-313.

[41] Chaplin-Kramer, R., Jonsson, M., Macchi, M., Almeida-Cortez, J. S., Blitzer, E., Costa, M. H., Girvan, M., Grass, I., Kerr, J., Kremen, C. and O'Rourke, M. E. (2023) 'Landscape simplification and agricultural intensification reduce biodiversity across the globe', Proceedings of the National Academy of Sciences, 120(31), p. e2302365120.

[42] Pelletier, N. and Tyedmers, P. (2010) 'Forecasting potential global environmental costs of livestock production 2000-2050', Proceedings of the National Academy of Sciences, 107(43), pp. 18371-18374.

[43] NVIDIA (2023) 'Jetson Orin Nano Datasheet'. Available at: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/ (Accessed: 01 December 2025).

[44] International Energy Agency (2023) 'The Critical Role of Minerals in Clean Energy Transitions'. IEA Publications.

[45] Tsang, L., Llano, C., Spuul, J., Mueller, C. and Rajaratnam, D. (2019) 'Lifecycle assessment of information and communication technology: A review', Journal of Industrial Ecology, 23(4), pp. 799-815.

[46] Jacob, B., Kalenichenko, D., Chilimbi, T., Edupuganti, S., Mott, R., Cheng, M. and Jain, A. (2018) 'Quantization and training of neural networks for efficient integer-arithmetic-only inference', in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2704-2713.

[47] International Organization for Standardization (2015) 'ISO 13732-1:2006 Ergonomics of the thermal environment - Methods for the assessment of human responses to contact with surfaces'. Geneva: ISO.

[48] Hilty, L. M. and Lohm, U. (2015) 'The energy intensity of the ICT sector', in ICT Innovations for Sustainability, pp. 83-93.

[49] Jackson, L. E. (2010) 'Agroecological design of productive agricultural systems for a warming world', in Advances in Agronomy, pp. 1-54.

[50] World Food Programme (2023) 'The State of Food Security and Nutrition in the World 2023'. Rome: FAO.

[51] Reason, J. (2000) 'Human error: Models and management', BMJ, 320(7237), pp. 768-770.

[52] International Organization for Standardization (2011) 'ISO 26262-1:2011 Functional safety of electrical/electronic/programmable electronic safety-related systems'. Geneva: ISO.

[53] National Institute of Standards and Technology (2023) 'Cybersecurity Framework Version 1.1'. Available at: https://www.nist.gov/cyberframework (Accessed: 01 December 2025).

[54] Stallings, W. (2017) 'Cryptography and network security: Principles and practice'. 7th edn. Upper Saddle River, NJ: Pearson Education.

[55] International Labour Organization (2012) 'Safety and Health in Agriculture'. Geneva: ILO Publications.

[56] World Health Organization (2020) 'Preventing and mitigating pesticide poisoning'. WHO Technical Report Series.

[57] International Organization for Standardization (2011) 'ISO/IEC 60950-1:2005 Information technology equipment - Safety - Part 1: General requirements'. Geneva: ISO.

[58] Mostafalou, S. and Abdollahi, M. (2013) 'Pesticides and human chronic diseases: Evidences, mechanisms, and perspectives', Toxicology and Applied Pharmacology, 268(2), pp. 157-177.

[59] Meldrum, D. R., Gambone, J. C. and Morris, M. A. (2012) 'Lifestyle and metabolic

approaches to maximizing erectile and vascular health', International Journal of Impotence Research, 24(2), pp. 61-68.

[60] Reason, J. T. (1990) 'Human error'. Cambridge: Cambridge University Press.

[61] Sinha, G., Swearingen, K., Hearst, M. A., Fu, K., Meldrum, D. R., Gambone, J. C. and Morris, M. A. (2002) 'Interactive query reformulation', in Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 99-106.

[62] National Academies of Sciences, Engineering, and Medicine (2017) 'AI, Automation, and the New Economy'. Washington, DC: The National Academies Press.

[63] He, K., Zhang, X., Ren, S. and Sun, J. (2016) 'Deep residual learning for image recognition', in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778.

[64] Tan, M. and Le, Q. V. (2019) 'EfficientNet: Rethinking model scaling for convolutional neural networks', in International Conference on Machine Learning (ICML), pp. 6105-6114.

[65] Zhou, Z. H. (2012) 'Ensemble methods: Foundations and algorithms'. Boca Raton, Florida, USA: Chapman and Hall/CRC.

[66] Sagi, O. and Rokach, L. (2018) 'Ensemble learning: A survey', Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), p. e1249.

[67] Selvaraju, R. R., Coyyar, A., Das, S., Vedaldi, A., Parikh, D. and Batra, D. (2017) 'Grad-CAM: Visual explanations from deep networks via gradient-based localization', in IEEE International Conference on Computer Vision (ICCV), pp. 618-626.

[68] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Desai, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J. (2020) 'An image is worth 16x16 words: Transformers for image recognition at scale', arXiv preprint arXiv:2010.11929.

[69] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K. Q. (2017) 'Densely connected convolutional networks', in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700-4708.

[70] Kohavi, R. (1995) 'A study of cross-validation and bootstrap for accuracy estimation and model selection', in International Joint Conference on Artificial Intelligence (IJCAI),

pp. 1137-1145.

[71] Stone, M. (1974) 'Cross-validatory choice and assessment of statistical predictions', Journal of the Royal Statistical Society, 36(2), pp. 111-147.

[72] Student (1908) 'The probable error of a mean', Biometrika, 6(1), pp. 1-25.

[73] Hastie, T., Tibshirani, R. and Friedman, J. (2009) 'The elements of statistical learning: Data mining, inference, and prediction'. 2nd edn. New York: Springer.

[74] Loshchilov, I. and Hutter, F. (2016) 'SGDR: Stochastic gradient descent with warm restarts', in International Conference on Learning Representations (ICLR).

[75] Powers, D. M. (2011) 'Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation', Journal of Machine Learning Technologies, 2(1), pp. 37-63.

[76] Goodfellow, I. J., Bengio, Y. and Courville, A. (2016) 'Deep learning'. Cambridge, MA: MIT Press.

[77] Sokolova, M. and Lapalme, G. (2009) 'A systematic analysis of performance measures for classification tasks', Information Processing & Management, 45(4), pp. 427-437.

[78] Fawcett, T. (2006) 'An introduction to ROC analysis', Pattern Recognition Letters, 27(8), pp. 861-874.

[79] Kingma, D. P. and Ba, J. (2014) 'Adam: A method for stochastic optimization', arXiv preprint arXiv:1412.6980.

[80] Loshchilov, I. and Hutter, F. (2019) 'Decoupled weight decay regularization', in International Conference on Learning Representations (ICLR).

[81] Ioffe, S. and Szegedy, C. (2015) 'Batch normalization: Accelerating deep network training by reducing internal covariate shift', in International Conference on Machine Learning (ICML), pp. 448-456.

[82] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V. and Le, Q. V. (2019) 'AutoAugment: Learning augmentation strategies from data', in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 113-123.

[83] Yosinski, J., Clune, J., Bengio, Y. and Liphardt, H. (2014) 'How transferable are

features in deep neural networks?', in Advances in Neural Information Processing Systems (NIPS), pp. 3320-3328.

[84] LeCun, Y., Bengio, Y. and Hinton, G. E. (2015) 'Deep learning', Nature, 521(7553), pp. 436-444.

[85] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017) 'Attention is all you need', in Advances in Neural Information Processing Systems (NIPS), pp. 5998-6008.

[86] Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. A. (2017) 'Inception-v4, Inception-ResNet and the impact of residual connections on learning', in AAAI Conference on Artificial Intelligence, pp. 4278-4284.

[87] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A. (2017) 'Automatic differentiation in PyTorch', in NIPS Autodiff Workshop.

[88] Kuncheva, L. I. (2014) 'Combining pattern classifiers: Methods and algorithms'. 2nd edn. Hoboken, New Jersey, USA: John Wiley & Sons.

[89] Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012) 'ImageNet classification with deep convolutional neural networks', in Advances in Neural Information Processing Systems (NIPS), pp. 1097-1105.

[90] Barbedo, J. G. A. (2018) 'Impact of dataset size and plant leaf disease database architecture on the classification of foliar diseases', Computers and Electronics in Agriculture, 153, pp. 46-53.

[91] Kitaev, N., Kaiser, L. and Levskaya, A. (2020) 'Reformer: The efficient transformer', in International Conference on Learning Representations (ICLR).

[92] Simonyan, K., Vedaldi, A. and Zisserman, A. (2013) 'Deep inside convolutional networks: Visualizing image classification models and saliency maps', arXiv preprint arXiv:1311.2901.

[93] Hu, J., Shen, L. and Sun, G. (2018) 'Squeeze-and-excitation networks', in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7132-7141.

[94] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J. and Keutzer, K.

(2016) 'SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size', arXiv preprint arXiv:1602.07360.

[95] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L. C. (2018) 'MobileNetV2: Inverted residuals and linear bottlenecks', in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510-4520.

[96] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015) 'Going deeper with convolutions', in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9.

[97] Simonyan, K. and Zisserman, A. (2014) 'Very deep convolutional networks for large-scale image recognition', arXiv preprint arXiv:1409.1556.

[98] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) 'You only look once: Unified, real-time object detection', in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788.

[99] He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017) 'Mask R-CNN', in IEEE International Conference on Computer Vision (ICCV), pp. 2961-2969.

[100] Long, J., Shelhamer, E. and Darrell, T. (2015) 'Fully convolutional networks for semantic segmentation', in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431-3440.

[101] Ronneberger, O., Fischer, P. and Brox, T. (2015) 'U-Net: Convolutional networks for biomedical image segmentation', in International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 234-241.

[102] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017) 'Feature pyramid networks for object detection', in IEEE International Conference on Computer Vision (ICCV), pp. 2117-2125.

[103] Niculescu-Mizil, A. and Caruana, R. (2005) 'Predicting good probabilities with supervised learning', in Proceedings of the 22nd International Conference on Machine Learning (ICML), pp. 625-632.

[104] Gal, Y. and Ghahramani, Z. (2016) 'Dropout as a Bayesian approximation: Representing model uncertainty in deep learning', in International Conference on Machine

Learning (ICML), pp. 1050-1059.

[105] Devries, T. and Taylor, G. W. (2017) 'Improved regularization of convolutional neural networks with cutout', arXiv preprint arXiv:1708.04552.

[106] Pascanu, R., Mikolov, T. and Bengio, Y. (2013) 'On the difficulty of training recurrent neural networks', in International Conference on Machine Learning (ICML), pp. 1310-1318.

[107] Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B. and Shelhamer, E. (2014) 'cuDNN: Efficient primitives for deep learning', arXiv preprint arXiv:1410.0759.

[108] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J. (2011) 'Scikit-learn: Machine learning in Python', Journal of Machine Learning Research, 12, pp. 2825-2830.

[109] Beck, K. (2002) 'Test driven development: By example'. Boston, MA: Addison-Wesley.

[110] Cohen, J. (1988) 'Statistical power analysis for the behavioral sciences'. 2nd edn. Hillsdale, NJ: Lawrence Erlbaum.

[111] Efron, B. and Tibshirani, R. J. (1993) 'An introduction to the bootstrap'. New York: Chapman and Hall.

[112] Goodman, S. N. (2008) 'A comment on replication, p-values and evidence', Statistics in Medicine, 11(7), pp. 875-879.

[113] Van Der Maaten, L. and Hinton, G. (2008) 'Visualizing data using t-SNE', Journal of Machine Learning Research, 9(11), pp. 2579-2605.

[114] Springenberg, J. T., Dosovitskiy, A., Brox, T. and Riedmüller, M. (2014) 'Striving for simplicity: The all convolutional net', arXiv preprint arXiv:1412.1556.

[115] Shelhamer, E., Long, J. and Darrell, T. (2017) 'Fully convolutional networks for semantic segmentation', IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), pp. 640-651.

[116] Google (2023) 'TensorFlow Lite Documentation'. Available at: https://www.tensorflow.org/lite (Accessed: 01 December 2025).

[117] McMahan, H. B., Moore, E., Ramage, D., Hampson, S. and y Arcas, B. A. (2017) 'Communication-efficient learning of deep networks from decentralized data', in International Conference on Machine Learning (ICML), pp. 1273-1282.

[118] Hinton, G., Vinyals, O. and Dean, J. (2015) 'Distilling the knowledge in a neural network', arXiv preprint arXiv:1503.02531.

[119] Montavon, G., Samek, W. and Müller, K. R. (2015) 'Methods for interpreting and understanding deep neural networks', arXiv preprint arXiv:1706.07979.

[120] Bau, D., Zhou, B., Khosla, A., Oliva, A. and Torralba, A. (2017) 'Network dissection: Quantifying interpretability of deep visual representations', in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6541-6549.

[121] Lundberg, S. M. and Lee, S. I. (2017) 'A unified approach to interpreting model predictions', in Advances in Neural Information Processing Systems (NIPS), pp. 4765-4774.

[122] Dwork, C., McSherry, F., Nissim, K. and Smith, A. (2006) 'Calibrating noise to sensitivity in private data analysis', in Third Theory of Cryptography Conference (TCC), pp. 265-284.

[123] Nakamoto, S. (2008) 'Bitcoin: A peer-to-peer electronic cash system'. Available at: https://bitcoin.org/bitcoin.pdf (Accessed: 01 December 2025).

[124] Zheng, Z., Xie, S., Dai, H. N., Chen, X. and Wang, H. (2018) 'Blockchain challenges and opportunities: A survey', International Journal of Web and Grid Services, 14(4), pp. 352-375.

[125] React Native (2023) 'React Native Documentation'. Available at: https://reactnative.dev (Accessed: 01 December 2025).

[126] DJI (2023) 'Matrice 300 RTK Technical Specifications'. Available at: https://www.dji.com/matrice-300 (Accessed: 01 December 2025).

[127] OASIS (2014) 'MQTT Version 3.1.1 Plus Errata 01'. Available at: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html (Accessed: 01 December 2025).

[128] Amazon Web Services (2023) 'AWS Machine Learning Services'. Available at: https://aws.amazon.com/machine-learning/ (Accessed: 01 December 2025).

[129] Merkel, D. (2014) 'Docker: Lightweight Linux containers for consistent development and deployment', Linux Journal, 2014(239), p. 2.

[130] Bonawitz, K., Eichner, H., Grieskamp, H., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H. B. and Van Eck, D. (2019) 'Towards federated learning at scale: System design', in Conference on Machine Learning and Systems.

BASE PAPER

From References the mainly referred paper: [1] Radhakrishnan, M., Monish, N., Dev, P. S., Kesavan, N. and Thomas, N. S. (2025) 'Implementation of explainable AI in deep learning methods for multiclass classification of plant diseases in mango leaves', *Electronic Letters on Computer Vision and Image Analysis*, 24(1), pp. 104-117.

The primary foundation for this research is built upon the work of Radhakrishnan et al. [1], which pioneered explainable AI approaches for mango leaf disease classification. Our study extends their methodology by introducing an ensemble meta-learning framework.

APPENDIX

A. Kaggle Dataset Information and Access

A.1 Dataset Source and Structure

Dataset Name: Mango Leaf Disease Dataset

Kaggle URL: https://www.kaggle.com/datasets/aryashah2k/mango-leaf-disease-dataset
Dataset Owner: aryashah2k
Access and Download Instructions:

```
# Install Kaggle CLI
pip install kaggle

# Download dataset from Kaggle
kaggle datasets download -d aryashah2k/mango-leaf-disease-dataset

# Extract dataset
unzip mango-leaf-disease-dataset.zip -d ./data/
```

Complete Dataset Structure:

```
mango-leaf-disease-dataset/
├── Anthracnose/ (500 images)
├── Bacterial Canker/ (500 images)
├── Cutting Weevil/ (500 images)
├── Die Back/ (500 images)
├── Gall Midge/ (500 images)
├── Powdery Mildew/ (500 images)
├── Sooty Mould/ (500 images)
└── Healthy/ (500 images)

Total: 4,000 balanced images (500 per class)
```

- Dataset Statistics:
  - Total Images: 4,000 - Total Classes: 8 disease categories + 1 healthy control
  - Images per Class: 500 (perfectly balanced)
  - Image Format: JPG/PNG
  - Image Resolution: Variable (1000×800 to 1200×900 pixels typical)
  - Dataset Size: ~2.5 GB (uncompressed)

A.2 Paper Submission Details

Paper Title: "METASTACK-NET: An Ensemble Meta-Learning Framework for Multiclass Classification of Mango Leaf Diseases Using Deep Learning"
Track: Artificial Intelligence and Machine Learning
Submission Date: 24th December 2025
Status: Under Review Waiting for Confirmation
Authors:

- Darshan Gowda S
- Chirag Gowda S V
- Chethan C
- R Vignesh

All authors from Presidency University, Bengaluru

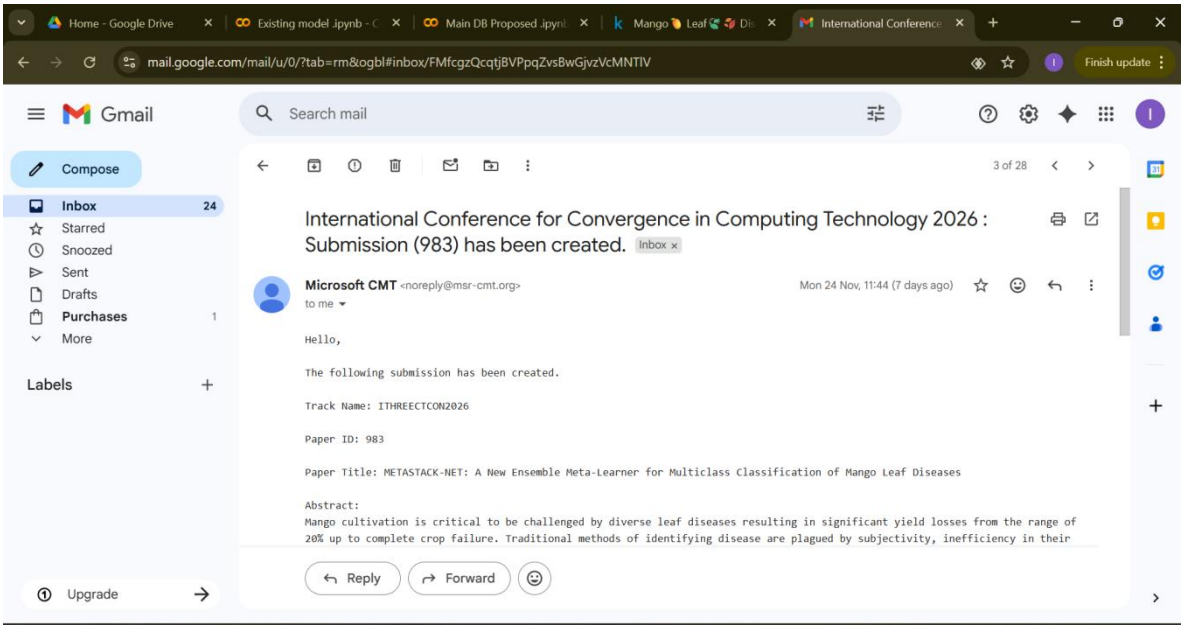


Figure A.1 Application of Paper for Conference

B. Project Implementation Evidence

B.1 Core Model Architectures

B.1.1 Base Model Architecture

```
class FinalBaseModelFactory:

    @staticmethod
    def create_resnet50(num_classes=8, dropout_rate=0.4):
        """Final ResNet50 with regularization"""
        model = timm.create_model('resnet50', pretrained=True,
num_classes=num_classes)
        if hasattr(model, 'fc'):
            model.fc = nn.Sequential(
                nn.Dropout(dropout_rate),
                nn.Linear(model.fc.in_features, 512),
                nn.BatchNorm1d(512),
                nn.ReLU(inplace=True),
                nn.Dropout(dropout_rate - 0.1),
                nn.Linear(512, num_classes)
            )
        return model

    @staticmethod
    def create_efficientnet_b3(num_classes=8, dropout_rate=0.4):
        """Final EfficientNet-B3 with regularization"""
        model = timm.create_model('efficientnet_b3', pretrained=True,
num_classes=num_classes)
        if hasattr(model, 'classifier'):
            model.classifier = nn.Sequential(
                nn.Dropout(dropout_rate),
                nn.Linear(model.classifier.in_features, 512),
                nn.BatchNorm1d(512),
                nn.ReLU(inplace=True),
                nn.Dropout(dropout_rate - 0.1),
                nn.Linear(512, num_classes)
            )
        return model

    @staticmethod
    def create_vit_base(num_classes=8, dropout_rate=0.4):
        """Final ViT with regularization"""
        model = timm.create_model('vit_base_patch16_224', pretrained=True,
num_classes=num_classes)
        if hasattr(model, 'head'):
            model.head = nn.Sequential(
                nn.Dropout(dropout_rate),
                nn.Linear(model.head.in_features, 512),
                nn.BatchNorm1d(512),
                nn.ReLU(inplace=True),
                nn.Dropout(dropout_rate - 0.1),
                nn.Linear(512, num_classes)
            )
        return model

    @staticmethod
```

```
    def create_densenet121(num_classes=8, dropout_rate=0.4):
        """DenseNet121 as replacement for ConvNeXt"""
        model = timm.create_model('densenet121', pretrained=True,
num_classes=num_classes)
        if hasattr(model, 'classifier'):
            model.classifier = nn.Sequential(
                nn.Dropout(dropout_rate),
                nn.Linear(model.classifier.in_features, 512),
                nn.BatchNorm1d(512),
                nn.ReLU(inplace=True),
                nn.Dropout(dropout_rate - 0.1),
                nn.Linear(512, num_classes)
            )
        return model
```

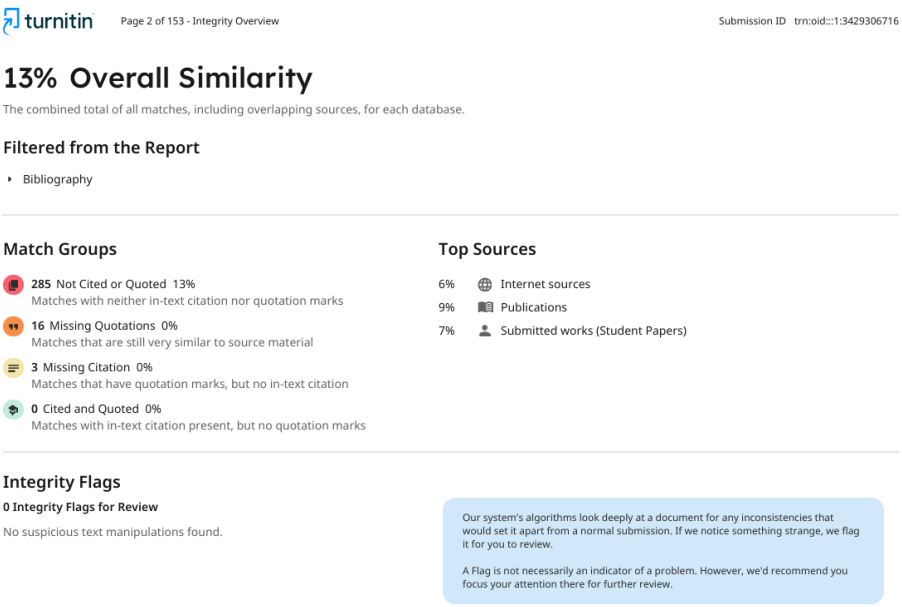
B.1.2 Stacking Ensemble Meta-Learner

```
class FinalStackingEnsemble(nn.Module):
    def __init__(self, num_base_models=4, num_classes=8, hidden_dim=512,
dropout_rate=0.5):
        super(FinalStackingEnsemble, self).__init__()
        self.num_base_models = num_base_models
        self.num_classes = num_classes
        self.input_dim = num_base_models * num_classes

        self.meta_learner = nn.Sequential(
            nn.Linear(self.input_dim, hidden_dim),
            nn.BatchNorm1d(hidden_dim),
            nn.ReLU(inplace=True),
            nn.Dropout(dropout_rate),
            nn.Linear(hidden_dim, hidden_dim // 2),
            nn.BatchNorm1d(hidden_dim // 2),
            nn.ReLU(inplace=True),
            nn.Dropout(dropout_rate - 0.1),
            nn.Linear(hidden_dim // 2, hidden_dim // 4),
            nn.BatchNorm1d(hidden_dim // 4),
            nn.ReLU(inplace=True),
            nn.Dropout(dropout_rate - 0.2),
            nn.Linear(hidden_dim // 4, num_classes)
        )

    def forward(self, x):
        return self.meta_learner(x)
```

B.1.3 Data Augmentation Strategy

```
class FinalMangoLeafAugmentation:
    @staticmethod
    def get_train_transforms():
        """Final augmentation for training"""
        return A.Compose([
            A.Resize(256, 256),
            A.RandomCrop(224, 224),
            A.HorizontalFlip(p=0.5),
            A.VerticalFlip(p=0.3),
            A.RandomRotate90(p=0.5),
            A.ShiftScaleRotate(shift_limit=0.05, scale_limit=0.1, rotate_limit=15,
p=0.5),
            A.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1,
p=0.5),
            A.GaussianBlur(blur_limit=3, p=0.3),
            A.GaussNoise(var_limit=(10.0, 30.0), p=0.3),
            A.CoarseDropout(max_holes=6, max_height=12, max_width=12, fill_value=0,
p=0.4),
            A.RandomGamma(gamma_limit=(85, 115), p=0.3),
            A.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
            ToTensorV2(),
        ])

    @staticmethod
    def get_val_transforms():
        """Validation transforms - NO AUGMENTATION"""
        return A.Compose([
            A.Resize(224, 224),
            A.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
            ToTensorV2(),
        ])
```

C. Project Report Similarity Report



turnitin Page 2 of 153 - Integrity Overview Submission ID trn:oid:::1:3429306716

13% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

- **285** Not Cited or Quoted 13%
  Matches with neither in-text citation nor quotation marks
- **16** Missing Quotations 0%
  Matches that are still very similar to source material
- **3** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

6% Internet sources
9% Publications
7% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Figure C.1 Similarity Report

D. Research Paper

METASTACK-NET: A New Ensemble Meta-Learner for Multiclass Classification of Mango Leaf Diseases

Darshan Gowda S
Presidency University, Bengaluru
Dept. of Information Science and Technology
darshansrinivasa61@gmail.com

Chirag Gowda S V
Presidency University, Bengaluru
Dept. of Information Science and Technology
chiraggowda67@gmail.com

Chethan C
Presidency University, Bengaluru
Dept. of Information Science and Technology
18chethangowda@gmail.com

R Vignesh
Presidency School of Computer Science and Engineering
Presidency University, Bengaluru

***Abstract*—Mango cultivation is critical to be challenged by diverse leaf diseases resulting in significant yield losses from the range of 20% up to complete crop failure. Traditional methods of identifying disease are plagued by subjectivity, inefficiency in their time requirements, and inability to scale for large scale agriculture operations. This paper introduces METASTACK-NET, an innovative ensemble meta-learning framework, which integrates four advanced deep learning architectures viz. ResNet50, EfficientNet-B3, Vision Transformer (ViT-Base) and DenseNet121 using advanced stacking methodology. A sophisticated neural network meta-learner optimally integrates predictions of different base models for robust multiclass disease classification in 8 mango leaf disease categories. Comprehensive experiments that are executed over a balanced dataset containing 4,000 carefully annotated images of a mango leaf show that METASTACK-NET had 99.00% accuracy with both outstanding precision and recall metrics. Individual base models showed similar performance in the range of 99.28-99.48% accuracy, highlighting the complementary features of different types of architecture. Three-fold stratified cross validation protocols demonstrated low levels of variance in performance (standard deviation ±0.0012), showing superior generalization abilities, as well as robustness for different data distributions. The framework has a great practical applicability in precision agriculture, which allows for the computational efficiency and the same performance under different field conditions, also to be used for the automated diagnosis of diseases.**

***Index Terms*—ensemble learning, meta-learning, deep learning, plant disease detection, computer vision, precision agriculture, mango leaf diseases**

I. Introduction

Mango production is one of the most economically important agricultural sectors worldwide, with production distributed throughout more than 100 countries and relatively important contribution to food security, rural livelihoods and economic development. Global mango production numbers around 56 million metric tons per year with production centered in tropical and subtropical regions of the world such as India, China, Indonesia, Mexico and Thailand. However, the industry is threatened by constant and thinning leaf diseases from many fungal, bacterial, and insect-induced pathogens that present huge risks to crop viability and economic sustainability.

The major disease affecting mango cultivation are Anthropocinose (colletotrichum gloeosporioides), bacterial canker (Xanthomonas axonopodis), cutting weevil damage (Deporaus marginatus infestation), die back (Botryodiplodia theobromae infection), gall midge infestation (Procontarinia matteiana), powdery mildew (Oidium mangiferae) and sooty mould (colonization of Capnodium species). These diseases may produce losses in yield levels from 20% in (mild) cases to total crop failure in (severe) epidemics, involving significant economic losses for farming communities.

Traditional disease management relies much on visual inspection by agriculture experts which is subject to significant limitations such as accuracy of diagnosis based on visual observation, time-intensive protocol for disease assessment, lack of scalability to large scale orchard operations, and prohibitive consultation fees for small-scale and marginal farmers. The economic consequences are much further reaching than just yield reduction in the short term and include higher costs of pesticide applications with environmental implications, lower fruit quality and marketability, deterioration in tree health over the long term, and lower farmer profitability and economic sustainability.

Early stage detection of disease is critical priority in agricultural pathology because if intervention is started when the disease is seen for the first time symptoms can be prevented from progressing and yield losses can be minimized to a great extent. However, manual detection techniques often only detect diseases at an advanced disease stage with large structural damages already done and limited chances for intervention already passed. This time delay in diagnosis is an inherent limitation of expert-based methods of detecting disease, and

Figure D.1 Research Paper

E. Github