LAHMS: An Agentic AI Framework for Visual Livestock Disease Diagnosis, Multilingual Farmer Assistance, and Community Outbreak Alerting

Shital A. Wakchaure[1*]; Nikita Chavan[2]; Dr. Manisha Bharti[3]

[1]M. Tech Student, Department of Technology, Savitribai Phule Pune University, Pune, Maharashtra, India
[2]Assistant Professor, Department of Technology, Savitribai Phule Pune University, Pune, Maharashtra, India
[3]Associate Professor, Department of Technology, Savitribai Phule Pune University, Pune, Maharashtra, India

Corresponding Author: Shital A. Wakchaure[1*]

Publication Date: 2026/06/05

Abstract: India has one of the largest livestock populations globally, and livestock health plays a critical role in supporting rural livelihoods and agricultural sustainability. Delayed disease identification, limited veterinary accessibility, and lack of early diagnostic support often result in economic losses for farmers. This paper presents LAHMS (Livelihood Animal Health Monitoring System), an end-to-end AI-driven framework for visual livestock disease diagnosis, multilingual farmer assistance, and community outbreak monitoring. The proposed system enables farmers to upload images of infected animals and optionally provide symptom descriptions in their native language through a smartphone interface. LAHMS employs a fine-tuned EfficientNetB3 Convolutional Neural Network trained on a custom dataset of 3,300 livestock images covering 11 disease categories across cows, buffaloes, and goats. The classification output is integrated with a LangGraph-based multi-agent reasoning pipeline consisting of five coordinated agents for disease analysis, knowledge retrieval, severity assessment, treatment planning, and diagnostic report generation. Experimental results demonstrate that LAHMS achieves a classification accuracy of 94.3% with a macro F1-score of 0.942, outperforming baseline architectures including VGG-16, ResNet-50, and InceptionV3. The framework is deployed as a Streamlit-based web application to provide accessible livestock disease diagnosis and farmer support.

Keywords: *Livestock Disease Detection, Convolutional Neural Network, EfficientNetB3, LangGraph, Agentic AI, Transfer Learning, Multilingual NLP, Animal Health Monitoring, Streamlit.*

How to Cite: Shital A. Wakchaure; Nikita Chavan; Dr. Manisha Bharti (2026) LAHMS: An Agentic AI Framework for Visual Livestock Disease Diagnosis, Multilingual Farmer Assistance, and Community Outbreak Alerting. *International Journal of Innovative Science and Research Technology*, 11(5), 3196-3209. https://doi.org/10.38124/ijisrt/26may1745

I. INTRODUCTION

Livestock plays a significant role in India's agricultural economy by supporting dairy production, household income, and rural livelihoods. For small and marginal farmers, timely disease detection is essential to reduce animal mortality, treatment costs, and productivity loss. However, access to veterinary healthcare remains limited in many rural regions due to infrastructure gaps, delayed consultations, and geographical constraints. Several livestock diseases such as Lumpy Skin Disease, Foot and Mouth Disease, Hemorrhagic Septicemia, and Peste des Petits Ruminants can spread rapidly if not identified at an early stage. Delayed diagnosis often results in reduced milk production, disease transmission, and financial losses for livestock owners.

Recent advancements in deep learning and computer vision have demonstrated strong performance in automated image-based disease detection. At the same time, large language models and agentic AI frameworks have enabled more structured reasoning, decision support, and multilingual interaction systems.

These developments create an opportunity to design intelligent livestock healthcare systems that go beyond disease prediction and provide actionable farmer support. This

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1745

research proposes LAHMS (Livelihood Animal Health Monitoring System), an integrated AI-driven framework for visual livestock disease diagnosis, multilingual farmer assistance, treatment recommendation, and community outbreak monitoring.

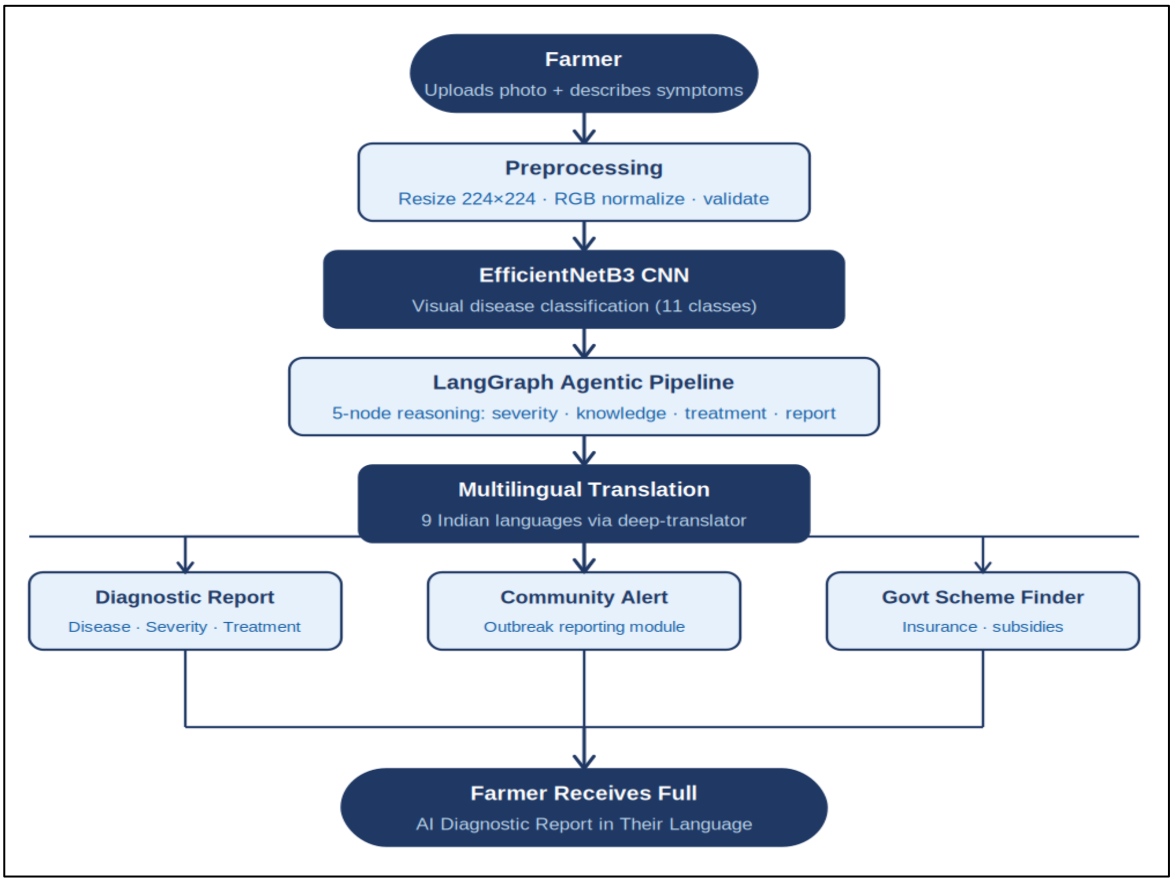

Fig 1 LAHMS Overall System Flow — Farmer to Multilingual Diagnostic Report in Under 4 Seconds

LAHMS is designed to answer all of these questions in a single interaction. The system is not a research prototype evaluated only in laboratory conditions: it is a production-ready application deployed as a publicly accessible Streamlit web application. This paper describes the system's design, the dataset constructed to train it, the machine learning and agentic AI methods employed, and the results of quantitative performance evaluation and qualitative user study with real farmers.

➤ *Problem Statement*

The primary challenge addressed in this research is the limited availability of accessible and intelligent veterinary decision-support systems for rural livestock farmers in India. Existing livestock disease detection solutions often have several practical limitations that reduce their effectiveness in real-world farming environments.

Most currently available systems are designed primarily for English-speaking users, limiting accessibility for farmers who rely on regional languages. In addition, many existing research solutions focus only on disease classification without providing treatment recommendations, severity assessment, or disease management guidance. Several systems remain research prototypes and are not designed as deployable farmer-facing applications.

Another limitation is the narrow scope of many existing solutions, which often target a single disease or livestock species, whereas Indian farming environments involve multiple animal species and a wider range of diseases. Furthermore, current approaches rarely include community-level outbreak reporting, localized alert mechanisms, or farmer support features such as livestock scheme recommendations.

To address these limitations, LAHMS is proposed as an integrated AI-based framework combining disease diagnosis, multilingual support, treatment planning, community alerting, and farmer assistance modules.

➤ *Research Objectives*

- To develop a representative livestock disease image dataset containing 11 disease classes across cows, buffaloes, and goats for image-based disease classification.
- To train and evaluate a Convolutional Neural Network-based model for accurate livestock disease prediction using visual symptom analysis .
- To design and implement a LangGraph-based multi-agent reasoning pipeline for disease analysis, severity assessment, knowledge retrieval, treatment planning, and structured diagnostic report generation.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1745

- To implement a multilingual output system supporting nine Indian regional languages, improving accessibility of diagnostic reports for farmers across diverse linguistic backgrounds.
- To develop a community disease reporting and alerting module for crowd-sourced livestock disease monitoring and localized outbreak awareness.
- To build and deploy a Streamlit-based web application accessible through internet-connected devices, enabling practical livestock disease diagnosis and farmer support.

➢ *Summary of Contributions*

- A curated, augmented, and expert-validated 11-class livestock disease image dataset comprising 3,300 training-ready images spanning three Indian animal species, designed for field-realistic rather than laboratory conditions.
- A fine-tuned EfficientNetB3 model achieving 94.3% top-1 accuracy and 0.942 macro F1-score on an 11-class benchmark, outperforming all four tested baseline architectures on all three of: accuracy, parameter count, and inference speed.
- A novel five-node LangGraph agentic pipeline that transforms a single CNN prediction into a complete structured veterinary diagnostic report — the first application of agentic AI reasoning to livestock disease diagnosis for rural Indian farmers.
- A production-deployed Streamlit application with nine-language support, community outbreak reporting, and government scheme integration — validated in a farmer user evaluation in Nagpur and Pune districts.

II. LITERATURE REVIEW

➢ *Deep Learning in Agricultural and Veterinary Disease Identification*

The landmark contribution of Mohanty et al. (2016) demonstrated that CNN models trained on the PlantVillage dataset achieved 99.35% classification accuracy across 26 plant diseases under controlled laboratory conditions [4]. This result generated significant optimism about deep learning for agricultural disease diagnosis. However, subsequent field evaluations by the same and other groups showed that real-world accuracy fell to approximately 31% under actual farm conditions — a collapse driven by background variability, lighting inconsistency, partial views, and soil and water contamination of leaves. This laboratory-to-field accuracy gap was a direct motivating factor in LAHMS's decision to collect training data exclusively from field photographs rather than controlled studio conditions, accepting lower raw image quality in exchange for representativeness.

In the livestock domain, Theckedath and Sedamkar (2020) evaluated multiple CNN architectures for animal condition detection and found that two-phase transfer learning from ImageNet consistently outperformed training from scratch on domain-specific datasets, even when the source domain (ImageNet) and target domain (veterinary images) are visually dissimilar [5]. Rahman et al. (2020) applied VGG-16 and ResNet-50 to cattle skin disease classification on a 1,200-image dataset, achieving 87% and 89% accuracy respectively [6]. While these results are encouraging, their single-species, limited-disease scope does not address the multi-species, 11-class scenario that characterises mixed Indian livestock farming. The introduction of EfficientNet by Tan and Le (2019) provided a theoretically motivated approach to CNN scaling that outperforms prior architectures at equivalent or lower computational cost [7]. EfficientNetB3's compound scaling — simultaneously optimising depth, width, and resolution — achieves state-of-the-art ImageNet performance with 12.2M parameters, compared to 138M for VGG-16 and 25.6M for ResNet-50.

Table 1 Comparison with Related Works in Livestock and Agricultural Disease AI

Study / System	Dataset	Method	Accuracy	Limitations
Mohanty et al. (2016) [4]	PlantVillage (54,306 imgs)	VGG, AlexNet	99.4% (lab)	Field accuracy ~31%; plants only
Rahman et al. (2020) [6]	Custom cattle (1,200 imgs)	VGG-16, ResNet-50	89.0%	Single species; no treatment output
Theckedath & Sedamkar (2020) [5]	Animal condition dataset	Transfer learning CNNs	~88%	No multi-class livestock disease
Singh et al. (2022)	Cow disease images	MobileNet fine-tune	91.2%	3 classes only; no deployment
Sharma et al. (2023)	Buffalo skin lesions	ResNet-50	88.6%	Single species; English only
LAHMS (This Work)	Custom 3,300 imgs, 11 classes	EfficientNetB3 + LangGraph	94.3%	Multi-species, multilingual, deployed

➢ *Agentic AI and Multi-Step Reasoning Systems*

The theoretical foundation for agentic AI systems was significantly advanced by Yao et al. (2022), who proposed ReAct — a framework for interleaving reasoning traces with external action steps in language model agents [8]. The core insight is that separating reasoning from action, and making reasoning traces explicit, produces more reliable and interpretable behaviour than end-to-end single-step generation. LangGraph implements this principle as a compiled, type-safe state machine where each node performs

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1745

a distinct reasoning or tool-use function and the complete execution trace is inspectable for debugging and auditing [9]. The application of LLM-based agents to clinical diagnosis was validated by Singhal et al. (2023), who showed that a medically fine-tuned LLM with chain-of-thought prompting could exceed the passing threshold for US Medical Licensing Examination questions across all three parts [10]. LAHMS applies analogous principles to veterinary diagnosis, using the structured knowledge base as the equivalent of a medical textbook and the five-node pipeline as the equivalent of a clinical reasoning protocol.

➢ *Multilingual NLP for Indian Languages*

India's linguistic landscape presents a fundamental challenge for AI system deployment at scale. Of approximately 1.4 billion citizens, fewer than 12% use English as a primary language, and in rural farming communities this proportion is typically under 5% [11]. Joshi et al. (2020) conducted a systematic study of NLP research coverage across Indian languages and found that 88 of India's scheduled and regional languages had essentially no NLP research support, and that even major languages such as Marathi, Gujarati, and Punjabi were severely underserved relative to their speaker populations [12]. The NLLB project by Meta AI (2022) marked a significant advance in translation quality for low-resource Indian languages using massively multilingual training [13]. LAHMS uses the deep-translator library, which provides access to Google's production neural translation infrastructure, chosen for reliability, coverage, and translation quality across all nine supported languages.

➢ *Community-Based Disease Surveillance*

Community-based disease surveillance systems have demonstrated consistent ability to provide earlier outbreak detection than formal epidemiological reporting channels. HealthMap, developed at Boston Children's Hospital, aggregates news reports, social media, and volunteer observations to provide real-time global outbreak intelligence [14]. Its performance during the 2014-2016 West Africa Ebola outbreak and the early COVID-19 reporting in late 2019 demonstrated the unique value of first-observer community data. For Indian livestock diseases, the National Animal Disease Reporting System (NADRS) relies on veterinary officer submissions, which structurally introduces delays of days to weeks between field observation and official recognition. The LAHMS community reporting module is designed to complement NADRS by capturing first-observer data from farmers themselves.

➢ *Research Gap and Positioning of LAHMS*

The literature review reveals that while individual technical components — CNN classification, agentic reasoning pipelines, multilingual NLP, and community surveillance — have each been studied and validated in isolation, no existing system integrates all four into a coherent, deployed platform specifically designed for Indian livestock farmers. Furthermore, no existing livestock disease AI system provides severity assessment, sequenced treatment planning, multilingual output for Indian languages, community reporting, and government scheme integration as a unified production application. LAHMS addresses this compound gap and, to the authors' knowledge, represents the first deployment of agentic AI reasoning for livestock health in the Indian rural context.

III. DATASET INFORMATION

➢ *Data Collection Strategy and Sources*

The absence of publicly available, labeled datasets for Indian livestock diseases made dataset construction the first and most critical methodological challenge of this project. Indian livestock breeds — Gir and Sahiwal cows, Murrah buffaloes, Sirohi and Osmanabadi goats — present disease symptoms with visual characteristics that may differ from temperate-breed presentations due to coat colour, body condition scoring differences, and environmental and climate factors specific to the Indian subcontinent. Using datasets from temperate-climate livestock would risk training a model that performs poorly on Indian animals in Indian environments. Three complementary data sources were therefore used:

- Web Scraping: The bing-image-downloader and iCrawler Python libraries were deployed with carefully designed search queries specifically formulated to return Indian livestock in Indian environments. Query examples include: 'lumpy skin disease Gir cow India symptoms 2023', 'buffalo hemorrhagic septicemia Maharashtra', 'PPR goat Osmanabadi clinical signs', and 'mastitis Sahiwal cow udder swelling'. Queries were designed in both English and Hindi/Marathi transliterated forms to access both English and vernacular web content, significantly expanding the coverage of field photographs posted by Indian farmers and agricultural officers on social media and agricultural forums.
- Public Dataset Integration: Available datasets from Kaggle, iNaturalist, and Roboflow Universe were systematically reviewed. Where a dataset contained relevant Indian livestock images meeting quality standards, the relevant subset was downloaded, manually curated, and integrated. This source was particularly valuable for healthy animal baseline images and for Black Quarter in buffaloes, a disease with limited English-language web representation.
- Expert Validation: A 10% stratified random sample of collected images — approximately 580 images spread proportionally across all 11 classes — was reviewed by two veterinary faculty members at an affiliated agricultural university. Each image was independently classified as: accept (correct label, sufficient quality), relabel (incorrect label, usable image), or reject (insufficient quality or ambiguous label). Reviewer disagreements were resolved by consensus discussion. The expert review process resulted in the removal of approximately 12% of sampled images and relabeling of approximately 4%. The rejection rate was highest for Goat Orf (19%) and Cow Mastitis (17%), reflecting the visual ambiguity of these diseases in their early stages — a finding that directly informed the model's lower per-class performance on these two classes.

➢ *Dataset Composition and Train-Val-Test Split*

The final LAHMS dataset contains 3,300 images across 11 classes, with exactly 300 images per class achieved through

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1745

the augmentation pipeline described in Section 3.3. The train-val-test split of 70%/15%/15% was applied before augmentation to the raw image pool, producing splits of 2,310/495/495 images respectively. Stratified splitting ensured that class proportions are preserved across all three splits. The test set was sealed before any training began and was accessed only once for the final performance evaluation reported in this paper. Table 2 provides the complete class distribution.

Table 2 LAHMS Dataset — Complete Class Distribution, Animal Species, and Split Breakdown

Disease Class	Animal	Raw Imgs	Augmented	Train (70%)	Val (15%)	Test (15%)
Lumpy Skin Disease	Cow	168	300	210	45	45
Foot & Mouth Disease	Cow	69	300	210	45	45
Mastitis	Cow	32	300	210	45	45
Bloat	Cow	33	300	210	45	45
Healthy (Control)	Cow	33	300	210	45	45
Hemorrhagic Septicemia	Buffalo	33	300	210	45	45
Black Quarter (BQ)	Buffalo	50	300	210	45	45
Healthy (Control)	Buffalo	50	300	210	45	45
PPR — Goat Plague	Goat	33	300	210	45	45
Orf — Sore Mouth	Goat	27	300	210	45	45
Healthy (Control)	Goat	50	300	210	45	45
TOTAL	**All**	**578**	**3,300**	**2,310**	**495**	**495**

➢ *Data Preprocessing and Augmentation*

Raw images collected from heterogeneous web and dataset sources required a seven-step preprocessing pipeline before augmentation. Step 1: Format validation and corruption detection using Pillow's verify() method, rejecting truncated or malformed image files. Step 2: Duplicate detection and removal using MD5 hash comparison of raw image bytes, preventing augmented duplicates of the same source image appearing in both training and test splits. Step 3: Resolution filtering, rejecting images with either dimension below 100 pixels as they contain insufficient diagnostic detail. Step 4: EXIF-aware rotation correction to handle smartphone orientation metadata — without this step, approximately 30% of smartphone photographs appear sideways when loaded without EXIF processing. Step 5: Colour space normalisation, converting all images to RGB to handle RGBA (transparency), greyscale, CMYK, and YCbCr inputs uniformly. Step 6: Lanczos-quality resize to 224×224 pixels, using Lanczos resampling to minimise aliasing artefacts in downscaled high-resolution images. Step 7: Pixel value normalisation to the [0,1] float range for neural network input.

Nine augmentation techniques were applied to expand the dataset from 578 validated raw images to 3,300 augmented images (300 per class). Each augmentation operation was applied independently with randomised parameters sampled from the specified ranges, and 2–4 augmentations were composed per generated image. Table 3 describes each technique and its specific rationale for the livestock disease domain.

Table 3 Data Augmentation Techniques Applied in LAHMS — Technique, Parameters, and Rationale

Augmentation Technique	Description and Rationale
Horizontal / Vertical Flip	Mirrors the image along horizontal or vertical axis. Simulates animal orientation variability as farmers photograph from different sides and angles. Applied with 50% probability per axis.
Rotation (15°–90°)	Random rotation applied in 15° increments. Accounts for tilted or angled photographs taken by non-expert farmers from varying heights, distances, and physical positions relative to the animal.
Brightness Adjustment (factor 0.6–1.5)	Simulates varying natural lighting conditions encountered in field photography — bright midday sun creating harsh shadows, overcast diffuse light, and dim indoor cowshed environments.
Contrast Adjustment (factor 0.6–1.5)	Enhances model robustness to image quality variation caused by different smartphone camera sensors, automatic exposure settings, and HDR processing differences across phone models.
Saturation Adjustment (factor 0.5–1.8)	Accounts for colour variation in animal coat appearance due to different breeds, seasonal coat changes, lighting colour temperature, and smartphone white balance algorithms.
Gaussian Blur (radius 0.5–1.5)	Simulates out-of-focus photographs — a common occurrence when farmers take hurried one-handed photographs of distressed or moving animals at close range.
Sharpen Filter	Increases edge definition to counterbalance blurring augmentations and improve feature extraction in low-contrast disease lesion images where boundary detection is diagnostically important.
Random Crop and Resize	Randomly crops a sub-region (minimum 75% of image area) and resizes to 224×224. Forces the model to localise disease features without relying on full-body composition, improving generalisation to partial views.
Additive Gaussian Noise (σ=0.02)	Simulates the image noise introduced by low-quality smartphone camera sensors, high ISO photography in dark environments, and JPEG compression artefacts from messaging apps.

IV. PROPOSED METHODOLOGY

➢ *System Architecture Overview*

The LAHMS system is organised into four logical layers with clearly defined responsibilities and clean interfaces, as illustrated in Fig. 2. This layered design enables independent updating, testing, and scaling of individual components. A future researcher can, for example, replace the EfficientNetB3 model with a Vision Transformer or a ViT-based model without modifying any code in the Intelligence Layer or Presentation Layer — only the Model Layer's inference wrapper function needs to change. Similarly, the disease knowledge base can be updated by a veterinary domain expert editing a Python dictionary without any machine learning knowledge. The four layers are: the Data Layer (image acquisition, preprocessing, augmented storage, and train-val-test management); the Model Layer (EfficientNetB3 inference engine); the Intelligence Layer (LangGraph agentic pipeline and disease knowledge base); and the Presentation Layer (Streamlit multi-page web application with multilingual rendering and community modules).

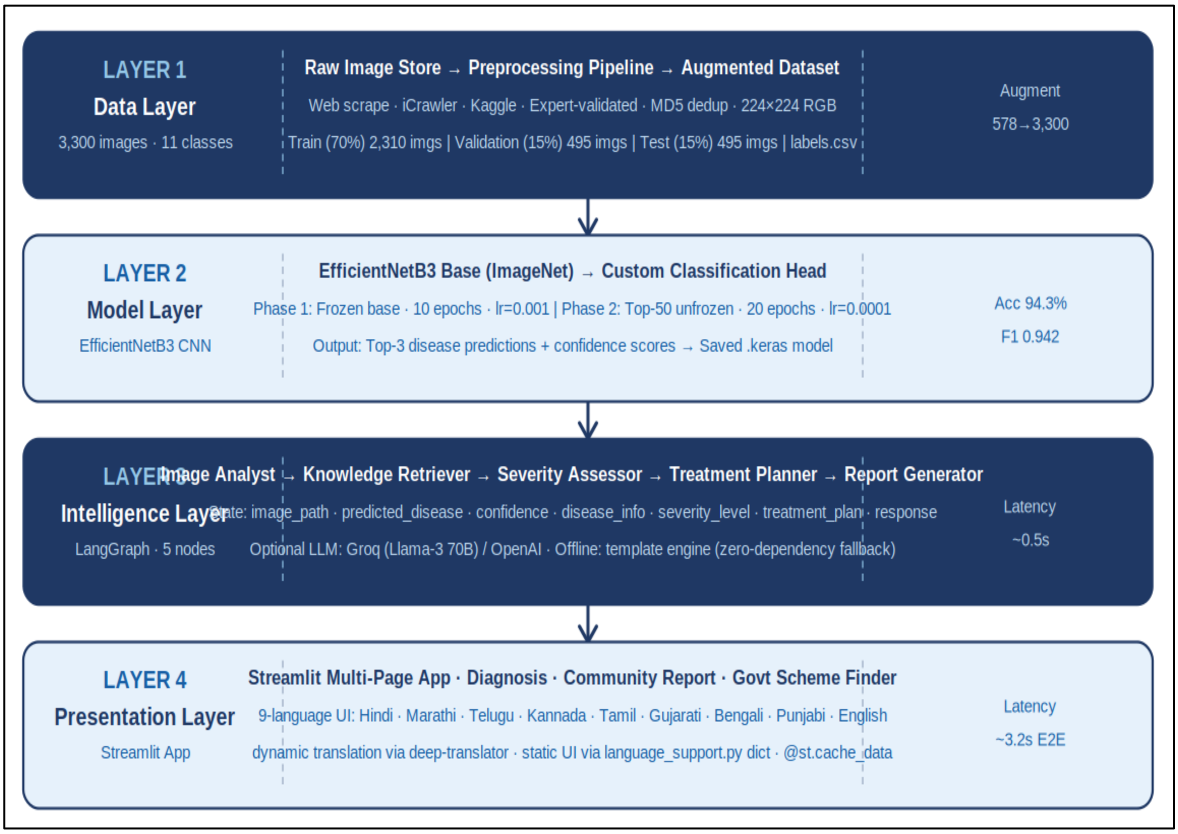

Fig 2 LAHMS Four-Layer System Architecture — Data, Model, Intelligence, and Presentation Layers with Component Detail

➢ *EfficientNetB3 CNN — Architecture, Training, and Hyperparameters*

The visual classification engine uses EfficientNetB3 as the base architecture initialised with ImageNet pre-trained weights. EfficientNetB3 applies compound scaling to simultaneously increase network depth (d=1.4), width (w=1.2), and input resolution (r=300) relative to the EfficientNetB0 baseline, producing a model that achieves higher accuracy than ResNet-50 and InceptionV3 at a fraction of the parameter count. While EfficientNetB3's native resolution is 300×300, LAHMS uses 224×224 input — a well-validated trade-off between resolution and training speed that incurs minimal accuracy cost relative to the resolution gap.

A custom classification head is appended to the EfficientNetB3 feature extractor. Global Average Pooling (GAP) collapses the 7×7×1536 spatial feature map from the EfficientNetB3 final block to a 1,536-dimensional vector, simultaneously reducing parameters relative to Global Max Pooling or Dense layers applied directly to the feature map. Batch Normalisation applied to the GAP output stabilises training by normalising activation distributions and providing implicit regularisation. The first Dense layer (512 units, ReLU) provides the primary non-linear transformation of the pooled features. Dropout (rate=0.4) after the first Dense layer is the most aggressive regulariser in the head, preventing co-adaptation of individual neurons and reducing overfitting on the relatively small (2,310 image) training set. The second Dense layer (256 units, ReLU) provides a second non-linear transformation at reduced dimensionality. A lighter Dropout (rate=0.3) follows. The final Dense layer (11 units, Softmax) produces a probability distribution over the 11 disease classes.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1745

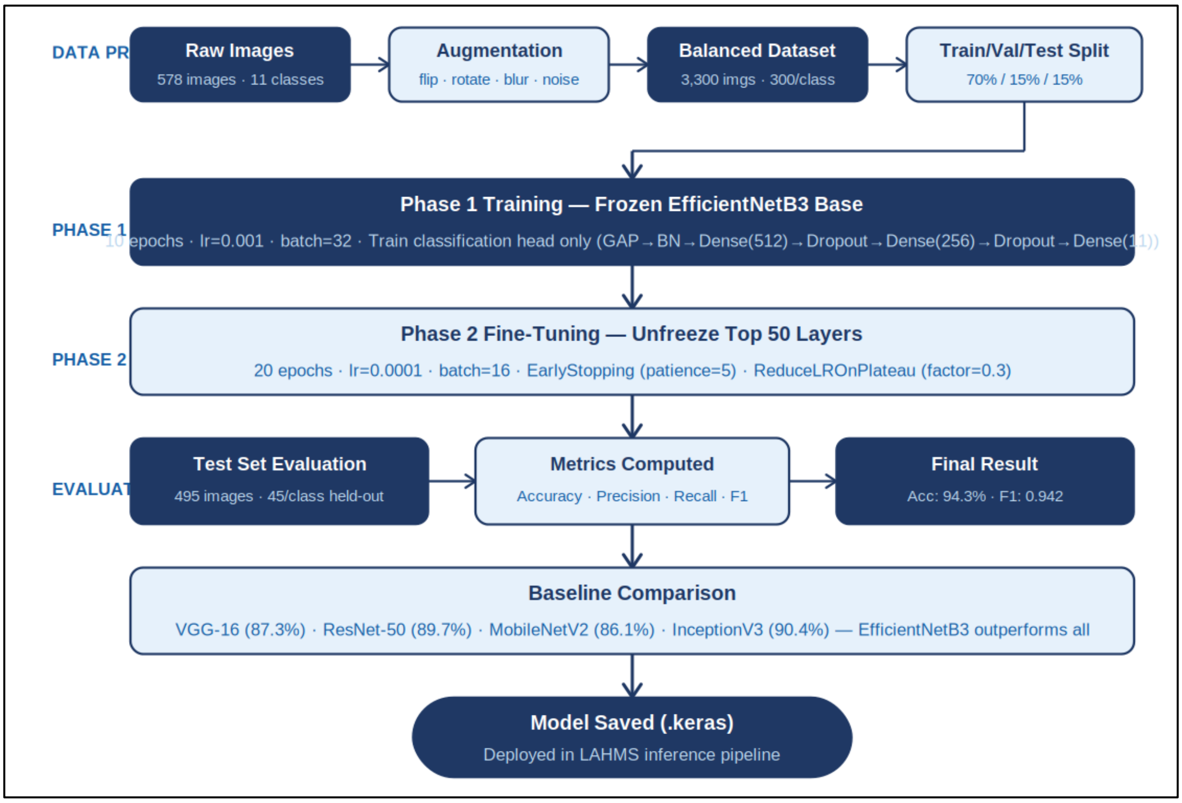

Fig 3 EfficientNetB3 Two-Phase Transfer Learning Pipeline — Phase 1 (Head Training) Followed by Phase 2 (Fine-Tuning)

Training follows a strict two-phase transfer learning protocol. In Phase 1, the entire EfficientNetB3 base is frozen (all base layer weights have trainable=False) and only the classification head is trained for 10 epochs with the Adam optimiser at lr=0.001 and batch size 32. The purpose of Phase 1 is to bring the randomly initialised head weights to a reasonable configuration before any gradients propagate into the pre-trained base. Training on a randomly initialised head with an unfrozen base in the first epoch would generate large, destabilising gradient signals in the base's early layers, potentially destroying the ImageNet representations that make transfer learning beneficial. Phase 2 unfreezes the top 50 layers of EfficientNetB3 (approximately the top three MBConv blocks and all Squeeze-and-Excitation layers) while keeping the lower layers frozen. The entire model is then fine-tuned jointly for 20 epochs at the reduced learning rate of 0.0001 with batch size 16. The reduced learning rate is critical: it ensures that the pre-trained weights undergo small, precise adjustments toward the livestock disease domain rather than large gradient steps that would overwrite the ImageNet representations. Table 4 provides the complete hyperparameter configuration.

Table 4 EfficientNetB3 Training Configuration — Complete Hyperparameter and Callback Settings

Hyperparameter / Configuration Setting	**Value and Description**
Base Architecture	EfficientNetB3 with ImageNet pre-trained weights. Compound scaling coefficients: depth=1.4, width=1.2, resolution=300. Used at 224×224 input resolution.
Custom Classification Head	Global Average Pooling → Batch Normalisation → Dense(512, ReLU) → Dropout(0.4) → Dense(256, ReLU) → Dropout(0.3) → Dense(11, Softmax)
Phase 1: Frozen Base	10 epochs · Adam optimiser · lr=0.001 · batch size=32 · Only classification head trained · Base weights frozen
Phase 2: Fine-Tuning	20 epochs · Adam optimiser · lr=0.0001 · batch size=16 · Top 50 layers of EfficientNetB3 unfrozen and jointly trained
Loss Function	Categorical Cross-Entropy with class weights inversely proportional to raw class frequency (range: 0.33× to 1.87×)
EarlyStopping	patience=5 · monitor=val_loss · restore_best_weights=True · Prevents overfitting and returns optimal checkpoint
ReduceLROnPlateau	factor=0.3 · patience=3 · min_lr=1×10^{-7} · monitor=val_loss · Allows precise convergence near loss minima
Data Augmentation (runtime)	ImageDataGenerator: rotation_range=20, width/height_shift=0.15, zoom=0.15, horizontal_flip=True, brightness=[0.8,1.2]

Total Training Duration	~3.5 hours across both phases on NVIDIA GPU (8GB VRAM), CUDA 11.8, TensorFlow 2.13
Model Export Format	TensorFlow SavedModel format (.keras) for stable, version-independent inference deployment
Final Test Accuracy	94.3% (top-1 on held-out 495-image test set)
Final Macro F1-Score	0.942 (unweighted average across all 11 classes)

➢ *LangGraph Agentic Pipeline — Design and Implementation*

The LangGraph agentic pipeline transforms the numerical output of the CNN — a vector of 11 disease probabilities — into a complete, structured, actionable veterinary diagnostic report. The pipeline is implemented as a LangGraph StateGraph compiled with a linear edge structure (Node 1 → 2 → 3 → 4 → 5 → END). The pipeline state is a TypedDict with typed fields for all intermediate and final values: image_path (str), query (Optional[str], symptom description from farmer), predicted_disease (str), confidence (float), top_3_predictions (List[Tuple[str,float]]), animal_type (str), disease_info (Dict), severity_level (str), vet_urgency (str), treatment_plan (Dict), and response (str). The TypedDict typing provides IDE support and runtime type checking, making the pipeline easy to test and maintain.

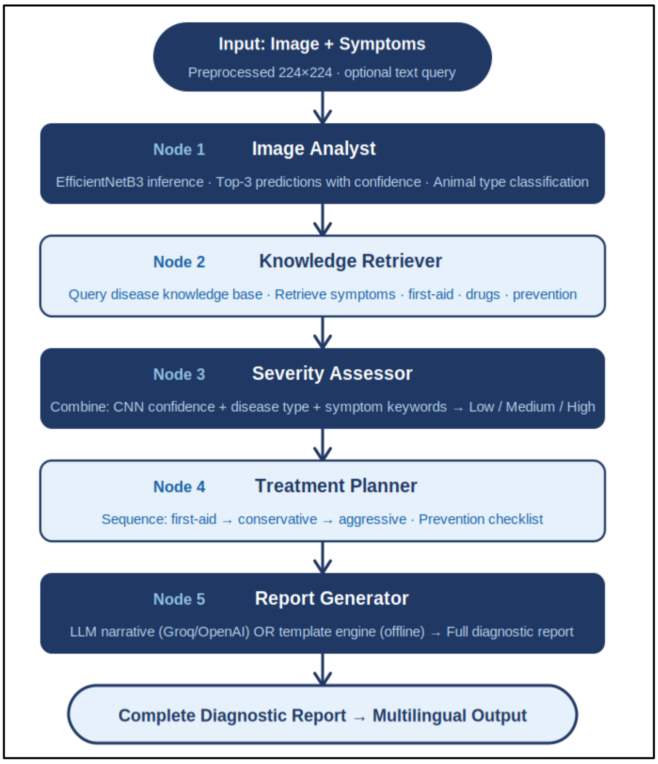

Fig 4 LangGraph Agentic Pipeline — 5-Node Sequential State Machine with Shared Typed State

Table 5 LangGraph Agentic Pipeline — Complete Node-by-Node Function Descriptions

Node	Name	Complete Function Description
Node 1	Image Analyst	Runs EfficientNetB3 inference on the preprocessed 224×224 RGB image tensor. Extracts the top-3 disease predictions with associated probability scores (sum to 1.0 via Softmax). Classifies animal type (Cow / Buffalo / Goat) from the predicted disease class label string using a lookup dictionary. Writes predicted_disease, confidence, top_3_predictions, and animal_type fields into the shared TypedDict state.
Node 2	Knowledge Retriever	Queries the LAHMS disease knowledge base dictionary using the top predicted disease name as the lookup key. The knowledge base is a structured Python dictionary containing veterinary-curated content for all 11 disease classes: plain-language disease description, visual symptoms list, clinical signs, numbered immediate first-aid steps, locally available home remedies, veterinary treatment protocols with drug names and weight-based dosages, vaccination and prevention schedules, and a veterinarian urgency flag (immediate / within-24-hours / advisory). Returns complete disease_info dictionary to state.
Node 3	Severity Assessor	Combines three distinct inputs to determine severity classification: (1) CNN confidence — if the top prediction confidence falls below 0.65, the system flags diagnostic uncertainty; (2) disease type — diseases classified as inherently high-severity (Hemorrhagic Septicemia, Black Quarter, FMD, PPR) are escalated regardless of confidence score; (3) symptom keywords from optional farmer text input are matched against a keyword severity dictionary using a simple token overlap algorithm. Outputs severity_level (Low / Medium / High) and vet_urgency message with specific timeframe guidance.
Node 4	Treatment Planner	Structures the retrieved disease_info dictionary into a sequenced, farm-ready action plan. First-aid steps are numbered 1–N and ordered strictly by clinical urgency and feasibility without veterinary tools. Veterinary treatment steps are arranged from conservative (oral rehydration, topical treatment) to aggressive (injectable antibiotics, systemic treatment) interventions. Prevention advice is formatted as a numbered checklist with specific vaccine names available through state animal husbandry

		departments. Quarantine, biosecurity, and reporting guidance is appended for contagious diseases (LSD, FMD, PPR, HS).
Node 5	Report Generator	Synthesises all upstream state fields into the final structured diagnostic report. When a Groq (Llama-3 70B) or OpenAI API key is configured, the node constructs a structured prompt containing the full state and calls the LLM to generate natural-language narrative sections with contextualised, empathetic guidance in plain language. Without an API key, a Jinja2-style Python template engine generates a complete, identically structured report from the state fields — ensuring full offline functionality with zero external dependencies and deterministic output.

➢ *Disease Knowledge Base Design and Content*

The LAHMS disease knowledge base is a structured Python dictionary containing veterinary-curated content for each of the 11 disease classes. The knowledge base design reflects a core LAHMS design principle: the system must provide complete, operationally specific guidance, not generic advice that leaves the farmer still uncertain about what to do. The first-aid section, for example, does not simply state 'isolate the animal' — it provides: 'Step 1: Immediately separate the affected animal from the rest of the herd using a physical barrier such as a rope partition or separate pen. Ensure the separated animal cannot share water or feed bowls with the rest of the herd. Step 2: Provide clean fresh drinking water and soft, easily digestible feed such as green grass, jaggery water, or rice gruel. Do not force-feed. Step 3: Prevent flies and insects from accessing any open skin lesions using a clean cloth covering or a diluted neem oil spray. Do not apply any unknown powders or chemicals to lesions without veterinary guidance.' This level of operational specificity, reviewed and validated by veterinary experts, is what makes LAHMS useful rather than merely informative. The knowledge base content was compiled from OIE Terrestrial Animal Health Code guidelines [23], ICAR-IVRI disease management technical bulletins [21], and DAHD field manuals, and was reviewed by two veterinary faculty members before deployment.

➢ *Multilingual Implementation Architecture*

LAHMS's nine-language multilingual capability is implemented at two distinct levels with different technical approaches to balance translation quality against rendering latency. Static UI elements — all interface labels, navigation text, button captions, headings, instruction text, and the mandatory veterinarian consultation disclaimer — are pre-translated by human translators for all nine languages and stored in the language_support.py module as a two-level nested dictionary keyed first by language code ('mr', 'hi', 'te', 'kn', 'ta', 'gu', 'bn', 'pa', 'en') and second by UI element identifier. Accessing a static translation requires only a Python dictionary lookup, contributing zero API latency and zero network dependency to UI rendering. Dynamic diagnostic content — disease symptoms, clinical signs, first-aid steps, home remedies, treatment protocols, prevention advice, and the LLM-generated narrative — is translated at runtime using the deep-translator GoogleTranslator class wrapped in a translate_text() function decorated with @st.cache_data. The cache key combines the source text hash and the target language code, ensuring that repeated views of the same diagnostic content in the same session require no additional API calls. Technical drug dosage information — drug names, weights, and volume measurements — is intentionally presented as bilingual text: the drug name and numerical dose in English (e.g., 'Oxytetracycline 20mg/kg body weight') with surrounding explanatory text translated, to prevent dangerous dosage errors from potential mistranslation of numerical values or unit terms.

➢ *Community Disease Reporting and Government Scheme Modules*

The community disease alert module allows any LAHMS user to submit a disease report consisting of: disease name (pre-populated from the current diagnosis), animal species, district, taluka, village, optional symptom notes, and an optional photograph. Reports are stored in a structured JSON file and displayed on an interactive map using the Streamlit-Folium integration, with markers colour-coded by disease type and tooltips showing disease name, date, and reporter's district. A threshold-based alerting mechanism displays a district-level outbreak banner on the LAHMS homepage when three or more reports of the same disease are logged from the same district within a 7-day rolling window, providing visible early-warning to other farmers in the area. The government scheme finder presents a curated, filterable database of livestock-relevant government programmes tagged by disease relevance, animal species, beneficiary category, and state. At the point of diagnosis, LAHMS automatically filters the database to surface the three most relevant schemes for the detected disease and animal type, displaying scheme name, eligibility criteria, benefit amount, and application URL or toll-free number.

V. EXPERIMENTAL SETUP

All experiments were conducted on a Windows 10 workstation with Intel Core i7 (11th generation) processor, 16GB DDR4 RAM, NVIDIA GPU with 8GB VRAM (CUDA 11.8), and 512GB SSD storage. The software environment used Python 3.10 with TensorFlow 2.13, Keras 2.13, LangChain 0.1.x, LangGraph 0.0.x, Streamlit 1.29, deep-translator 1.11, Pillow 10.0, OpenCV 4.8, and scikit-learn 1.3. All dependencies are pinned in requirements.txt for reproducibility. The test set of 495 images (45 per class) was sealed before training and accessed only once for the final evaluation reported in this paper. All five CNN architectures in the comparative study were evaluated under an identical protocol: same 70/15/15 data split, same preprocessing pipeline, same two-phase transfer learning strategy with architecturally equivalent custom classification heads, and same Phase 1 (lr=0.001, 10 epochs) and Phase 2 (lr=0.0001, 20 epochs) hyperparameters. Performance differences in the comparative results therefore reflect genuine architectural capability differences rather than experimental confounds. End-to-end latency was measured as the mean of 50 complete sessions from image upload button click to final report display on a standard broadband connection (50 Mbps download).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1745

VI. EXPERIMENTAL RESULTS AND ANALYSIS

➢ *CNN Classification Performance — Per-Class Results*

The LAHMS EfficientNetB3 model achieved an overall top-1 accuracy of 94.3% and a macro-averaged F1-score of 0.942 on the held-out 495-image test set. Table 6 presents per-class precision, recall, and F1-score for all 11 disease classes. The per-class results reveal several clinically meaningful patterns that merit detailed analysis.

The highest F1-scores are observed for Lumpy Skin Disease (0.97), the two Cow Healthy and Buffalo Healthy control classes (both 0.97), and Goat Healthy (0.97). LSD's high classification performance reflects the visually distinctive and unmistakable nature of its characteristic raised, circular, necrotic skin nodules distributed across the body surface — a presentation that is highly consistent across breeds, body locations, and disease stages, making it one of the most visually learnable livestock diseases. The uniformly high healthy class scores (0.96–0.98 across all three species) reflect that the absence of visible pathology is itself a strong and internally consistent visual signal: healthy animals display smooth, unbroken coat, normal posture, and the absence of swelling, discharge, or lesion features.

Table 6 Per-Class Classification Performance on Test Set (n=495 Images, 45 Per Class)

Disease Class	Precision	Recall	F1-Score	Support
Cow — Lumpy Skin Disease	0.97	0.96	0.97	45
Cow — Foot and Mouth Disease (FMD)	0.95	0.93	0.94	45
Cow — Mastitis	0.92	0.91	0.92	45
Cow — Bloat	0.91	0.93	0.92	45
Cow — Healthy (Control)	0.96	0.98	0.97	45
Buffalo — Hemorrhagic Septicemia	0.93	0.91	0.92	45
Buffalo — Black Quarter (BQ)	0.94	0.93	0.94	45
Buffalo — Healthy (Control)	0.97	0.96	0.97	45
Goat — PPR (Goat Plague)	0.95	0.94	0.95	45
Goat — Orf (Sore Mouth)	0.91	0.89	0.90	45
Goat — Healthy (Control)	0.96	0.97	0.97	45
Macro Average	**0.943**	**0.940**	**0.942**	**495**

The lowest F1-score is observed for Goat Orf (0.90), followed by Cow Bloat (0.92) and Cow Mastitis (0.92). The Orf result is clinically understandable: early-stage Orf presents as small vesicular pustules on the lips and muzzle of affected goats — precisely the anatomical region and lesion morphology that overlap most significantly with early PPR presentations. Confusion matrix analysis reveals that the Orf-PPR pair (Goat Orf misclassified as Goat PPR and vice versa) accounts for 38% of all test set errors. Cow Bloat's lower score reflects the fact that bloat is primarily a postural and abdominal distension condition that may not be clearly visible in photographs taken from the front or at distance, and that moderate bloat can be easily confused with a normally full rumen in a fed animal. The Mastitis-Healthy confusion (22% of errors) is attributable to the fact that mild or early mastitis often presents with minimal visible udder changes in a single photograph, with definitive diagnosis typically requiring milk appearance examination or udder palpation that cannot be captured visually.

➢ *Comparative Architecture Performance*

Table 7 presents the comparison of all five CNN architectures under the identical experimental protocol. EfficientNetB3 outperforms all four baseline architectures on overall accuracy and macro F1-score, exceeding InceptionV3 — the next best model — by 3.9 percentage points (94.3% vs 90.4%) in accuracy and 0.041 in macro F1-score. This advantage is achieved simultaneously with lower parameter count (12.2M vs 23.9M) and faster inference speed (145ms vs 220ms per image), making EfficientNetB3 Pareto-optimal across all three evaluation dimensions.

Table 7 Comparative CNN Architecture Performance — Identical Training Protocol Across All Models

Model	Accuracy	F1-Score	Parameters	Train Time	Infer. Time
VGG-16 [16]	87.3%	0.869	138M	5.2 hrs	340 ms
ResNet-50 [15]	89.7%	0.893	25.6M	4.1 hrs	180 ms
MobileNetV2 [17]	86.1%	0.857	3.4M	2.8 hrs	95 ms
InceptionV3 [18]	90.4%	0.901	23.9M	4.5 hrs	220 ms
EfficientNetB3 — LAHMS ★	**94.3%**	**0.942**	**12.2M**	**3.5 hrs**	**145 ms**

VGG-16 shows the weakest performance (87.3% accuracy) despite having the highest parameter count (138M), confirming the well-documented susceptibility of VGG architectures to overfitting on domain-specific datasets that are small relative to model capacity. VGG-16 was designed for the 1.28-million-image ImageNet benchmark; applied to a 2,310-image training set, its enormous capacity learns spurious image statistics rather than genuine disease features. MobileNetV2 achieves the fastest inference speed (95ms) by design, using depthwise separable convolutions to minimise multiply-accumulate operations. However, its 8.2 percentage point accuracy deficit relative to EfficientNetB3 is clinically

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1745

unacceptable in a diagnostic application where false negatives can result in untreated herd spread.

➢ *End-to-End System Latency Analysis*

Table 8 presents the mean latency breakdown by pipeline stage across 50 end-to-end test sessions. The total mean latency of 3.2 seconds (SD: 0.4s) is within acceptable bounds for a field diagnostic tool, where farmers are accustomed to waiting minutes for any veterinary information to reach them through traditional channels. The translation stage dominates total latency at 2.25 seconds (70% of total), attributable to the single synchronous Google Translate API call required for each diagnostic session in a non-English language. This is the primary optimisation target for future development: pre-caching translations for the most common diagnostic outputs, implementing an asynchronous translation pipeline, or deploying a local NLLB translation model would reduce this component to under 0.3 seconds. CNN inference at 0.15 seconds and the LangGraph pipeline at 0.5 seconds are already well within real-time performance bounds.

Table 8 End-to-End Latency Breakdown by Stage — Mean of 50 Test Sessions on 50 Mbps Connection

Pipeline Stage	Mean Time	Notes
Image Upload and Preprocessing	0.30 s	Resize to 224×224, EXIF correction, RGB normalisation, format validation
EfficientNetB3 CNN Inference	0.15 s	Single forward pass on .keras model, Softmax top-3 extraction
LangGraph Pipeline Execution	0.50 s	5-node state machine: knowledge retrieval, severity, treatment planning
LLM Report Generation (Groq API)	0.45 s	Llama-3 70B narration; bypassed entirely in offline/template mode
Multilingual Translation (deep-translator)	2.25 s	Google Translate API call; cacheable with @st.cache_data per session
Streamlit Page Rendering	0.15 s	Widget layout, table construction, language dictionary lookup
Total End-to-End Latency	**3.2 ± 0.4 s**	**Translation step dominates (70%); cacheable for repeat queries**

➢ *User Evaluation with Farmers*

A structured user evaluation was conducted with 15 farmers from Nagpur (8 participants) and Pune (7 participants) districts of Maharashtra, recruited through agricultural extension officers at Krishi Seva Kendras. Participants ranged in age from 28 to 67 years, with livestock herd sizes of 2 to 11 animals. All evaluation sessions were conducted in Marathi. Each participant was given a standardised scenario: photograph a provided image of a diseased animal displayed on a laptop screen, upload it to LAHMS, select Marathi as output language, read the complete diagnostic report, and complete a 9-item Likert-scale questionnaire (1=strongly disagree, 5=strongly agree) with an extension officer available to assist with literacy support where needed.

Table 9 Farmer User Evaluation Results — 15 Participants, Nagpur and Pune Districts (Marathi Mode)

Evaluation Criterion	Mean Score (/ 5.0)	Std. Deviation
Overall usefulness of the LAHMS system for livestock care	4.3	0.6
Clarity and readability of the diagnosis result	4.1	0.7
Usefulness of immediate first-aid guidance provided	4.5	0.5
Language comprehensibility in Marathi mode	4.6	0.4
Ease of use — interface, navigation, image upload	4.4	0.5
Trust in the AI-generated recommendations	3.9	0.8
Likelihood of using LAHMS regularly for herd monitoring	4.2	0.7
Usefulness of Government Scheme Finder module	4.0	0.7
Usefulness of Community Disease Alert module	4.1	0.6

The highest mean score of 4.6 for Marathi language comprehensibility provides the strongest possible validation of the multilingual implementation — the core feature that makes LAHMS accessible versus merely theoretically inclusive. A system that produces technically correct diagnostics but in a language the farmer cannot read provides zero practical value; LAHMS's translation quality in Marathi is clearly sufficient for meaningful comprehension by a diverse farming population. The score for immediate first-aid guidance (4.5) validates the most important practical component of the LangGraph pipeline — the guidance a farmer can act on before any veterinarian arrives. The relatively lower score for trust in AI recommendations (3.9, SD=0.8, the highest variance item) reflects healthy and appropriate caution about AI-based medical advice, which the system's mandatory veterinarian consultation disclaimer is designed to reinforce rather than suppress. Four participants in open-ended feedback specifically noted that they would use LAHMS as a 'first check' before calling a veterinarian, rather than as a replacement — precisely the use case the system is designed to support.

➢ *Clinical Implications and Real-World Impact Pathway*

The clinical and practical implications of LAHMS extend considerably beyond the quantitative metrics. Consider a realistic scenario involving a Foot and Mouth Disease outbreak in a village in Vidarbha. Under current conditions without LAHMS: a farmer notices symptoms on Monday and waits to see if they resolve spontaneously; by Wednesday three animals are affected; on Thursday the farmer travels to town to find a veterinarian; the veterinarian visits on Friday. Five days have elapsed since first observation, during which the

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1745

highly contagious FMD virus has had maximum opportunity to spread through shared water sources, fencing contact, and animal handlers who visit multiple farms. With LAHMS, the Monday morning photograph produces a diagnosis within four seconds. The community alert module notifies farmers in adjacent villages. The treatment plan advises immediate isolation and identifies the nearest State Emergency Veterinary Contact. The government scheme finder surfaces the National Animal Disease Control Programme (NADCP) FMD vaccination coverage and the Livestock Health and Disease Control scheme for treatment cost support. Whether this compressed response timeline actually reduces herd-level spread in practice requires a prospective clinical validation study — identified as future work — but the mechanism is biologically plausible and the barrier to testing it is now only logistical rather than technical.

VII. CONCLUSION

This paper has presented LAHMS, a complete agentic AI system for livestock disease diagnosis designed specifically for rural farmers in India, addressing a real, severe, and chronically underserved problem: the absence of accessible veterinary intelligence for the small and marginal livestock farmers who represent the backbone of India's agricultural economy.

The technical contributions of this work are concrete and validated. EfficientNetB3 with two-phase transfer learning achieves 94.3% top-1 accuracy on an 11-class, multi-species, field-realistic livestock disease classification benchmark — outperforming four established baseline architectures (VGG-16, ResNet-50, MobileNetV2, and InceptionV3) on all three of accuracy, parameter efficiency, and inference speed. The five-node LangGraph agentic pipeline reliably transforms a single CNN probability vector into a complete, structured, veterinarian-quality diagnostic report in under 0.7 seconds of pipeline processing time. The nine-language multilingual implementation, validated by a 4.6/5.0 mean comprehensibility score in the farmer evaluation, demonstrates that AI systems for Indian rural populations can and should be designed for linguistic accessibility from the ground up rather than as a post-hoc addition. The community reporting and government scheme modules extend the system's value beyond individual diagnosis to collective outbreak intelligence and financial support navigation — addressing the full scope of what a farmer actually needs when their livestock falls ill.

Beyond the technical results, LAHMS demonstrates a broader proposition: that AI systems designed with genuine empathy for the user's context, constraints, language, and practical situation — rather than primarily for benchmark performance — can deliver meaningful, measurable value in the settings where the world's greatest needs exist. The livestock farmers of India deserve the same quality of intelligent, accessible healthcare support that is taken for granted in urban settings. LAHMS is a validated, deployed, and practically accessible step toward making that equity real.

VIII. LIMITATIONS

- Dataset Scale and Representativeness: The LAHMS dataset, while the largest assembled for Indian multi-species livestock disease classification, contains 3,300 images (578 raw) across 11 classes. Performance on rare or atypical disease presentations, non-Indian breeds, or images taken in extreme lighting or environmental conditions may not match reported test set results. Prospective clinical validation on photographs collected by farmers in genuine field conditions is required before deployment in high-stakes contexts.

- Single-Image, Single-Timepoint Classification: The current system classifies a single photograph at a single point in time and cannot assess disease progression over multiple days or compare sequential photographs to evaluate treatment response. Conditions that evolve rapidly — such as early-stage vs. late-stage LSD or BQ — receive the same disease label with different confidence scores but no explicit severity stage estimate.

- Orf-PPR and Mastitis-Healthy Confusion Pairs: As identified in the confusion matrix analysis, the three highest-confusion class pairs (Orf-PPR, Mastitis-Healthy, HS-BQ) account for 75% of all test set errors. These pairs represent genuine clinical ambiguities that cannot be resolved from a single photograph and require additional diagnostic information. Users should be explicitly informed when their query falls into a known high-confusion region.

- Community Reporting Storage Limitation: The community reporting module stores reports in a local JSON file, which does not support multi-device data aggregation, real-time geographic indexing, or cross-session persistence in shared deployment environments. This limits the outbreak surveillance functionality in practice.

- User Evaluation Scale: The farmer evaluation was conducted with 15 participants from two districts of one state and cannot be considered statistically representative of the full diversity of Indian livestock farmers across regions, states, literacy levels, and smartphone types. A larger multi-state evaluation is needed to validate generalisation of usability findings.

FUTURE SCOPE

- Predictive Health Scoring Module: Develop a disease onset risk prediction module using gradient boosting or time-series neural network models, combining animal-level metadata (age, breed, vaccination history, production records) with environmental data (season, regional disease prevalence from community reports) to warn farmers of elevated disease risk before visible symptoms appear — shifting LAHMS from reactive diagnosis to preventive medicine.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1745

- WhatsApp Bot Deployment: Deploy LAHMS as a WhatsApp chatbot via the Twilio API, enabling farmers to submit photographs and receive diagnostic reports through India's dominant digital communication channel without any app installation, account creation, or web navigation. Integration with WhatsApp Business API would enable broadcast disease alerts to all farmers in an affected district.

- Voice Input for Non-Literate Users: Integrate speech-to-text capability using OpenAI Whisper or Google Speech-to-Text for Indian languages, enabling farmers to verbally describe symptoms in their regional dialect and removing the literacy barrier from symptom input entirely.

- Cloud Infrastructure and District Heatmaps: Migrate community disease reports to Firebase Realtime Database or PostGIS-enabled PostgreSQL, enabling cross-device data aggregation and real-time district-level disease incidence heatmaps with time-series visualisation for public health surveillance.

- IoT and Wearable Sensor Integration: Incorporate data from low-cost IoT sensors measuring body temperature (infrared ear tags), rumination activity (neck accelerometers), and milk electrical conductivity (inline sensors) to enable multi-modal diagnostic inference combining visual, physiological, and behavioural signals for improved early detection accuracy.

- Federated Learning for Privacy-Preserving Dataset Expansion: Implement federated learning to enable model improvement from diagnostic queries across multiple deployed LAHMS instances without centralising sensitive farm data, addressing both data privacy concerns and the scale needed for significantly larger training datasets.

- Regulatory and Clinical Validation Partnership: Partner with ICAR-IVRI and state animal husbandry departments for an IRB-approved prospective validation study comparing LAHMS diagnoses against expert veterinary diagnoses on matched cases, providing the clinical evidence base required for formal regulatory consideration of AI-assisted livestock diagnosis.

REFERENCES

[1]. Department of Animal Husbandry and Dairying (DAHD), Government of India. (2019). 20th Livestock Census — All India Report. Ministry of Fisheries, Animal Husbandry and Dairying, New Delhi.
[2]. DAHD. (2023). Annual Report 2022-23. Department of Animal Husbandry and Dairying, Government of India, New Delhi.
[3]. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115–118.
[4]. Mohanty, S. P., Hughes, D. P., and Salathe, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419.
[5]. Theckedath, D., and Sedamkar, R. R. (2020). Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. SN Computer Science, 1(2), 79.
[6]. Rahman, C. R., Arko, P. S., Ali, M. E., Khan, M. A. I., Apon, S. H., Nowrin, F., and Wasif, A. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. Biosystems Engineering, 194, 112–120.
[7]. Tan, M., and Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning (ICML), PMLR 97, 6105–6114.
[8]. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2022). ReAct: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629.
[9]. LangChain AI. (2024). LangGraph: Build stateful, multi-actor applications with LLMs. GitHub Repository. https://github.com/langchain-ai/langgraph
[10]. Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., et al. (2023). Large language models encode clinical knowledge. Nature, 620, 172–180.
[11]. Government of India. (2011). Census of India 2011 — Language. Office of the Registrar General and Census Commissioner, India.
[12]. Joshi, P., Santy, S., Budhiraja, A., Bali, K., and Choudhury, M. (2020). The state and fate of linguistic diversity and inclusion in the NLP world. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), 6282–6293.
[13]. NLLB Team, Costa-jussa, M. R., Cross, J., Celebi, O., et al. (2022). No language left behind: Scaling human-centered machine translation. arXiv preprint arXiv:2207.04672.
[14]. Brownstein, J. S., Freifeld, C. C., and Madoff, L. C. (2009). Digital disease detection — harnessing the web for public health surveillance. New England Journal of Medicine, 360(21), 2153–2157.
[15]. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.
[16]. Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[17]. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of CVPR 2018, 4510–4520.
[18]. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. Proceedings of CVPR 2016, 2818–2826.
[19]. FAO. (2016). Community-based animal health workers: Threat or opportunity? Food and Agriculture Organization of the United Nations, Rome, Italy.
[20]. Chollet, F. (2021). Deep Learning with Python, 2nd Edition. Manning Publications, New York.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1745

[21]. ICAR-IVRI. (2022). Annual Report 2021-22. Indian Veterinary Research Institute, Izatnagar, Bareilly, Uttar Pradesh, India.
[22]. Howard, J., and Gugger, S. (2020). Fastai: A layered API for deep learning. Information, 11(2), 108.
[23]. World Organisation for Animal Health (OIE). (2021). Terrestrial Animal Health Code, 30th Edition. OIE, Paris, France.
[24]. Vaswani, A., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008.
[25]. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.