# PROJECT

**Project Title :**

Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management

**Team Id:**

LTVIP2025TMID3

4061

**Team Members :**

- Tappita Jordan(Team Leader)
- Chanduri Jaswanth
- Thammineni Venkata Sravanthi
- Vasana Siva Sai Teja

# Phase-1 : Brainstorming & Ideation

## Objective:

To develop a transfer learning-based classification system for poultry diseases that enhances diagnostic accuracy, reduces manual analysis time, and supports poultry health management applications such as automated diagnostics, remote veterinary consultations, and agricultural education.

.

## Key Points:

1. **Problem Statement:**

    - Poultary Diseases, such as avian influenza, Newcastle disease, and coccidiosis, significantly impact poultry health and farm productivity. Early and accurate diagnosis is crucial but often relies on manual inspection, which is time-consuming, labor-intensive, and prone to human error.
    - A Transfer Learning Based Poultary Disease Classification System leverages deep learning models to automatically identify disease patterns from images (e.g., lesions, feces, or skin conditions). By integrating AI into poultry health monitoring, this system improves diagnostic accuracy, reduces response time, and supports effective disease management, remote consultations, and farmer education — ultimately enhancing poultry health outcomes and farm sustainability.

2. **Proposed Solution:**

    - An AI-powered An AI-powered application that uses transfer learning to accurately classify poultry diseases from visual data (such as images of lesions, droppings, or external symptoms) in real-time.
    - It assists veterinarians and farmers by providing quick, reliable disease identification, reducing manual inspection efforts, and improving the speed and accuracy of poultry health management.

3. **Target Users:**

    - Veterinarians and poultry health experts who need quick and accuratediagnosis of poultry diseases.
    - Poultry farms and agricultural enterprises aiming to automate disease detection and improve flock health management.
    - Veterinary professionals offering remote consultations who require fast and reliable diagnostic support.

**4.Expected Outcome:**

- A functional AI- powered application that accurately classifies poultry diseases from images using transfer learning.
- Provides fast,reliable diagnostic **support to enhance efficiency in poultry health management, disease control, and veterinary education.**

# Phase-2: Requirement Analysis

**Objective:**

Define the technical and functional requirements for the Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management application.

**Key Points:**

1. **Technical Requirements:**

**Programming Language**: Python
Python Packages: NumPy, Pandas, Scikit-learn, Matplotlib, SciPy, Seaborn, TensorFlow, OpenCV (for image processing), Flask
**Frameworks**:
TensorFlow – for implementing deep learning and transfer learning models
Flask – for building and integrating the web-based user interface
Pre-trained Model: VGG16 (or ResNet50/InceptionV3) – used for transfer learning to classify poultry diseases from images
Development Tools: Command Line Interface (CLI) using pip for installing dependencies
Image Input Format: JPEG, PNG (images of infected areas, droppings, or visible symptoms)

**2.Functional Requirements:**

- Ability to upload poultry disease-related images (e.g., lesions, droppings external symptoms) through a user-friendly web interface.
- Classify poultry diseases into categories such as avian influenza, Newcastle disease, coccidiosis, and fowlpox using a trained transfer learning model.
- Display classification results with prediction confidence scores to support informed decision-making.
- Provide an intuitive interface designed for use by veterinarians, farmers, veterinary students, and animal health professionals.

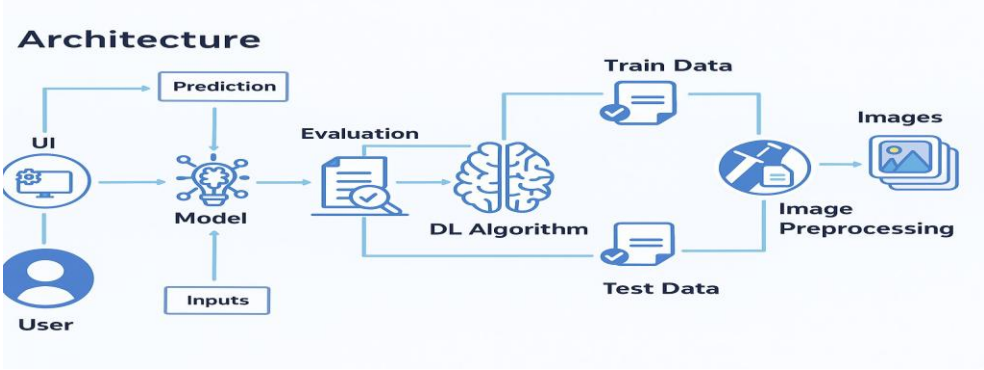
### 3.Constraints & Challenges:

- Handling imbalanced datasets across different poultry diseases, as some diseases may have significantly fewer labeled images than others.
- Dealing with low-quality or unclear images (e.g., poor lighting, motion blur, or background noise) which can reduce classification accuracy.
- Optimizing the model for real-time inference to ensure fast disease detection, especially on low-resource devices used in rural or field settings.
- Designing a responsive and easy-to-use web interface suitable for veterinarians, farmers, and non-technical users across various devices and screen sizes.

# Phase-3: Project Design

## Objective:

Develop the architecture and user flow of the Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management application.




## Key Points:

### 1.System Architecture:

- The user uploads a poultry disease-related image (e.g., skin lesion, droppings, or infected area) through the web interface (UI).
- The image is sent as input to the trained deep learning model hosted on the server.
- The model uses a pre-trained CNN (such as VGG16 or ResNet50) to process the image and predict the type of poultry disease.
- The prediction result, along with confidence levels, is evaluated and displayed back on the user interface for decision-making support.

- Initially, the model is trained using a labeled dataset of poultry disease images, which is preprocessed and split into training and testing sets.
- The training data is used to fine-tune the deep learning model via transfer learning, while the testing data is used to evaluate the model's accuracy and performance.

**2.User Flow:**

- Step 1: User opens the poultry disease classification web application.
- Step 2: Uploads an image showing visible symptoms (e.g., lesions, droppings, or infected areas of the bird).
- Step 3: The AI model processes the image and classifies the disease using transfer learning.
- Step 4: The predicted disease type and confidence score are displayed to the user.
  - Step 5: User can upload additional images or use the results for timely treatment, expert consultation, or record-keeping.

# Phase-4: Project Planning

**Objective:**

Breakdown development tasks for efficient completion.

| Sprint | Task | Priority | Duration | Deadline | Assigned to | Dependencies | Expected outcome |
|---|---|---|---|---|---|---|---|
| Sprint 1 | Environment Setup & Package Installation | 🔴 High | 3 hours | Day 1 | Member 1 | Anaconda , Python | Project environment ready |
| Sprint 1 | Dataset Collection & Preprocessing | 🔴 High | 4 hours | Day 1 | Member 2 | Dataset access | Clean,prepared image dataset |
| Sprint 2 | Model Building using Transfer Learning | 🔴 High | 5 hours | Day 2 | Member 3 | Preprocessed data, TensorFlow | Trained classification model |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Sprint 2 | Flask Web App Integration | 🟡 Medium | 3 hours | Day 2 | Member 1 & 4 | Trained Model,Flask installed | Working web interface |
| Sprint 3 | Testing & Debugging | 🟡 Medium | 2 hours | Day 2 | Member 2 & 3 | Complete System | Bug-free and responsive system |
| Sprint 3 | Final Presentation & Deployment | 🟢 Low | 1 hour | End of Day 2 | Entire Team | Working application | Project deployed and demo-ready |

## Sprint Planning with Priorities

### Sprint 1 - Setup & Preparation (Day 1)

#### 🔴 High Priority

- Set up the environment using Anaconda Navigator.
- Install all required Python packages (TensorFlow, Flask, etc.).
- Collect and preprocess the blood cell image dataset.

### Sprint 2 – Model Development & Integration (Day 2)

#### 🔴 High Priority

- Build and train the blood cell classification model using transfer learning (e.g., VGG16).
- Integrate the trained model with the Flask web application.

### Sprint 3 – Testing, Deployment & Submission (Day 3)

#### 🟡 Medium Priority

- Test the app functionality, fix bugs, and improve UI responsiveness.

#### 🟢 Low Priority

- Finalize deployment and prepare presentation/demo materials.

# Phase-5: Project Development

## Objective:

Implement the core features of the poultry disease classification application using transfer learning to identify diseases from poultry-related images.

### Key Points:

#### 1. Technology Stack Used:

- **Frontend:** HTML (via Flask templates)
- **Backend:** Flask Framework
- **Deep Learning:** TensorFlow with pre-trained VGG16 model
- **Programming Language**: Python

#### 2. Development Process:

- Built and trained a poultry disease classification model using transfer learning (VGG16).
- Preprocessed a dataset of labeled poultry disease images (e.g., lesions, infected skin, droppings).
- Integrated the trained model into a Flask-based web application.
- Developed a user interface to upload images and display disease classification results with prediction confidence.

#### 3. Challenges & Fixes:

- Challenge: Model overfitting on some poultry disease categories
- Fix: Applied data augmentation techniques and dropout regularization to improve generalization.
- Challenge: Large model size slowed down predictions.
- Fix: Optimized the model and saved it in .h5 format for faster loading and inference.
- Challenge: Variation in image quality due to real-world farm conditions.
- Fix: Added preprocessing steps like resizing, normalization, and noise reduction before prediction.

# Phase-6: Functional & Performance

**Objective:**

- Ensure that the Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management application performs accurately, reliably, and consistently across various test cases and environments.

| Test Case ID | Category | Test Scenario | Expected Outcome | Tester |
|---|---|---|---|---|
| TC-001 | Functional Testing | Upload image of eosinophils | Correct cell type identified with confidence score | Tester 1 |
| TC-003 | Performance Testing | Check model response time | Results displayed under 2 secods | Tester 3 |
| TC-004 | Bug Fix Validation | Image with poor lighting | System still makes reasonable prediction | Developer |
| TC-005 | UI Responsiveness | Test on mobile browser | Layout adjusts properly on mobile | Tester 2 |
| TC-006 | Deployment Testing | Hosted on local server and accessed remotely | App loads and predicts successfully online | DevOps |