**Transfer Learning-Based Classification of Poultry Diseases**

**for Enhanced Health Management**

# 1. Introduction

**Project Title**: Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management

**Team ID**: LTVIP2025TMID37158

**Team size**: 4

🔹 **Team Leader: L Yuva shree**

Project Coordination, Milestone Tracking, Model Building, Final Validation

🔹 **Team Member: N Charantejash**

* Data Collection, Cleaning, Preprocessing Pipeline

🔹 **Team Member: Rodda Vamsi Krishna Reddy**

* Problem Definition, Problem Understanding

🔹 **Team Member: Reddem Ganesh Reddy**

* ML Model Development & Flask Integration, UI/UX Design, Documentation

# 2. Project Overview

**2.1 Purpose**:

To develop an AI-driven system using transfer learning to identify common poultry diseases such as:

* Salmonella
* New Castle Disease
* Coccidiosis
* Healthy (no disease)

**2.2 Features**:

* Real-time image classification
* Symptom-based prediction
* Treatment suggestions
* Offline usability for rural farmers
* Educational resource for veterinary students

# 3. Architecture

**3.1. Frontend**

* Developed using **HTML5**, **CSS3**, and **Jinja2 templates** (via Flask)
* HTML form (index.html) allows users to **upload poultry images for prediction**
* UI is styled using embedded CSS and media served from the /static directory
* Clean and intuitive design for farmers and end-users
* Displays model prediction results (disease name) after image submission

**3.2. Backend**

* Backend logic implemented using **Python and Flask**
* **Model Training Pipeline:**

 Training, validation, and model export performed in **Google Colab**

 Data preprocessing, augmentation, and model training scripts run in Colab

 Final trained model exported as a .h5 file (e.g., healthy vs rotten.h5)

* **Model:**

A deep learning model using **Transfer Learning** (VGG16/VGG19/ResNet50)

Model artifacts include:

 poultry\_model.h5– Trained model file

 Preprocessing logic embedded in the Flask app for real-time image handling

**3.3. Data Preprocessing**

* **Image resizing** (typically to 224x224 for VGG/ResNet compatibility)
* **Normalization** – Scaling pixel values to the range [0, 1]
* **Augmentation** – Flipping, rotating, and zooming during training to enhance performance
* Preprocessing logic handled during training and reused during inference in Flask backend

**3.4. Database**

* **No persistent database** is used
* All predictions are handled **in-memory**
* Model receives image input directly, processes it, and returns output without storage
* Optionally, uploaded images and results can be logged locally or stored in cloud storage if needed

# 4. Setup Instructions

**4.1. Prerequisites**:

1. Python 3.10+ or anaconda
2. Flask
3. TensorFlow or PyTorch
4. Google Colob

**4.2. Installation**

Git clone https://github.com/TrishaKanderi/Transfer-Learning-Based-Classification-of-Poultry-Diseases-forEnhanced-Health-Management cd poultry\_disease\_detection/Flask

pip install -r requirements.txt python app.py

1. **Folder Structure**

**5.1. Client (Flask Frontend)**

The frontend is handled via **HTML templates** and **static assets**:

├── templates/

│ └── index.html # Main HTML page for image upload and prediction

│

├── static/

│ └── images/

│ └── poultry-bg.jpg # Background image

│ └── healthy.png # Image used for healthy prediction

│ └── infected.png # Image used for infected result

* index.html contains a **form to upload poultry images** and a **predict** button.
* **images** used for UI styling are stored under the static/ folder.

**5.2. Server (Flask Backend)**

The backend is developed using **Python and Flask**:

├── app.py # Main Flask application file (routing & prediction)

├── poultry\_model.h5 # Trained CNN model (VGG16/VGG19/ResNet50)

├── preprocess.py # (Optional) Image preprocessing helper functions

├── train\_model.ipynb # Google Colab notebook for model training

├── tensorflow.txt # Python package dependencies

* app.py handles:

 Routing (/ and /predict)

 Loading the .h5 model

 Image preprocessing and prediction logic

* poultry\_model.h5 is the **trained deep learning model** saved from Colab.
* train\_model.ipynb includes:

 Dataset loading

 Data preprocessing and augmentation

 Model training and evaluation

# 6. Running the Application

* Start the app:

cd Flask python app.py

* Open in browser at: http://127.0.0.1:5000/

# 7. API Documentation

This is a form-based web application. The /predict route accepts POST requests from the image upload form and returns an HTML page with the prediction result.

# POST /

**Route:** / (root URL) **Method:** POST

**Input:**

 User uploads a poultry image (e.g., image of a sick chicken) via the HTML form  File input field: name="file"

**Processing:**

1. The uploaded image is:
   * Loaded and resized (typically to 224x224) o Normalized (pixel values scaled to [0,1])
   * Converted into an array format compatible with the model
2. The processed image is passed to the trained model (poultry\_model.h5) for classification.
3. The model predicts the type of poultry disease (e.g., Newcastle Disease, Avian Influenza, etc.)

**Output:**

 Rendered index.html page displaying:

* + Prediction result: e.g., "Predicted Disease: Newcastle Disease"
  + Visual feedback: Image preview and result text

**Error Handling:**

 Displays a friendly error message if: o No image is uploaded o An unsupported file type is uploaded o File is too large or unreadable

 Flask app includes basic validations to ensure smooth user experience

# 8. Authentication

Not applicable – This application does not implement user login or authentication as it's a prototype for medical prediction.

# 9. User Interface

 Simple form to upload poultry bird images for disease prediction

 Clean UI with a semi-transparent poultry-themed background

 Responsive layout with easy-to-use buttons  Image preview with the result

# 10. Testing

 Model accuracy validated using evaluation metrics (accuracy, precision, recall) in Google Colab

 Manual testing of image upload form and prediction flow via Flask UI

 Model tested on a holdout test set (80/20 train-test split) to ensure generalization

 Verified predictions across different poultry disease classes for consistency

# 11. Screenshots or Demo



**Demo video link**:

https://drive.google.com/file/d/1TWK9cS-3sLZSRlMsyHsPNICjiTRjSVwn/view?usp=drivesdk

# 12 Known Issues

Limited dataset impacts prediction accuracy

No multi-language support (planned)

Poor image quality reduces model performance

# 13 Future Enhancements

Add more poultry diseases (e.g., Avian Influenza)

Integrate chatbot support for farmers

IoT sensor-based environment tracking

Admin dashboard for farm-wide monitoring