

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

### Project Overview

This project aims to develop a Transfer Learning-based system for classifying poultry diseases into four key categories:

Salmonella

New Castle Disease

Coccidiosis

Healthy

The core of the system is a deep learning model built using Transfer Learning techniques, specifically optimized for poultry image classification. The model is integrated into a Flask web application (with future scope for mobile deployment), allowing users – especially farmers – to diagnose poultry diseases efficiently.

The system empowers users to upload poultry images and, optionally, input data such as symptoms and environmental conditions to receive:

Real-time disease classification

Recommended treatments

Actionable health management advice

### Project Objectives

Develop an accurate poultry disease classifier using deep learning.

Use transfer learning (MobileNetV2 or equivalent) to improve model efficiency on limited datasets.

Deploy the trained model in a user-friendly web/mobile app.

Enhance poultry health management through timely and reliable disease detection.

### Technologies Used

Component	Technology
Programming	Python
Deep Learning	TensorFlow, Keras
Web Framework	Flask
UI Development	HTML, CSS, JavaScript
Visualization	Matplotlib, Seaborn
Data Manipulation	Pandas, NumPy

### System Architecture

Data Collection

Images categorized into four classes: Healthy, Salmonella, New Castle Disease, Coccidiosis

Preprocessing & Augmentation

Resize, normalize, and augment the data for better generalization

Model Training

Use of pre-trained MobileNetV2 model

Fine-tuning with poultry dataset

Evaluation using accuracy and loss metrics

Model Saving

Trained model saved in .h5 format for inference

Web Application (Flask)

Allows users to upload poultry images

Model predicts disease

Displays result with treatment suggestions

### Real-World Scenarios

#### Scenario 1: Outbreak in a Rural Community

A rural village sees an outbreak of sick birds. Farmers use the app to upload images and input symptoms (e.g., diarrhea, lethargy). The system classifies the disease as Coccidiosis and recommends immediate treatment steps, enabling the farmers to control the outbreak and reduce loss.

#### Scenario 2: Commercial Poultry Farm Management


A commercial poultry enterprise integrates this system into their daily health checks. When symptoms of New Castle Disease are detected, the app flags the area, and the farm managers quarantine the affected flock early, preventing a costly spread.


#### Scenario 3: Research and Training for Veterinary Students


Veterinary students use the app to simulate diagnostics from real-world scenarios. It helps them identify diseases like Salmonella and Coccidiosis, and offers insights into clinical symptoms and treatments – making learning interactive and practice-oriented.

### System Interface Features

 Upload poultry image

 Receive instant classification

 View disease information and treatment tips

 Navigate to blog and portfolio sections

### Project Directory Structure

bash

Copy

Edit

```
poultry_disease_classifier/
├── app.py
├── healthy_vs_rotten.h5
├── Readme.txt
├── ipython.html
├── static/
│   ├── uploads/
│   ├── assets/style.css
│   └── forms/form.js
├── templates/
│   ├── index.html
│   ├── blog-single.html
│   ├── blog.html
│   └── portfolio-details.html
```

### How to Run the Project

Install Required Packages

Open Anaconda Prompt:

bash

Copy

Edit

```
pip install numpy pandas scikit-learn matplotlib scipy seaborn tensorflow flask
```

Ensure Files Are in Place

app.py and healthy\_vs\_rotten.h5 in root folder

Templates and static files in respective folders

Start Flask App

bash

Copy

Edit

python app.py

Open in Browser

<http://127.0.0.1:5000>



#### Limitations

Model performance depends on training data size and variety

May misclassify if image is blurry, unrelated, or non-poultry

Current interface is web-based; mobile deployment is future scope



#### Future Enhancements

Develop a fully offline-capable Android application

Expand dataset for better generalization

Add text-based symptom input for improved diagnostic power

Integrate expert system for treatment recommendation

Enable API for third-party vet tools and monitoring platforms



#### Contributors

Vinay - Model training, Flask development, and integration

OpenAI ChatGPT - Documentation and code generation assistance