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

1. INTRODUCTION

1.1 Project Overview

Poultry Disease is a web-based image classification system designed toidentifypoultry diseases using artificial intelligence. The system employs transfer learning with the ResNet50 model to classify images of poultry into four categories: Healthy, Coccidiosis, Salmonella, and Newcastle Disease. The platform empowers farmers to make data-driven decisions by detecting diseases early, thus preventing potential outbreaks and minimizing economic loss

1.2Purpose

The primary goal is to mitigate the risks associated with late disease detection in poultry farming. By enabling real-time disease classification through a simple image upload, the system reduces dependency on expert consultations and accelerates the treatment process, enhancing overall farm productivity

2. IDEATIONPHASE

Problem Statement

Poultry farmers face significant difficulty in identifying diseases at an early stage. Traditional diagnosis methods are time-consuming, require expert intervention, and may not be affordable or accessible in rural regions. Delayed diagnosis results in high mortality rates and loss of revenue. The proposed solution aims to bridge this gap using AI and computer vision.

Empathy Map Canvas

An empathy map was used to better understand the challenges and motivations of poultry farmers:

- **Think & Feel:** Fear of losing flocks, concern for income.
- **See:** Symptoms in other nearby farms.
- **Say & Do:** Consult vets, attempt self-treatment.
- **Hear:** Peer stories, vet recommendations.
- **Pain:** Late detection, unaffordable care.
- **Gain:** Real-time, affordable disease detection system.

Brainstorming

We applied the MURAL brainstorming template to encourage out-of-the-box thinking. Key directions explored included:

- Mobile-based disease logging systems.
- Manual decision-tree based symptom checkers.
- Transfer learning with CNNs for image classification.

3. REQUIREMENT ANALYSIS

Customer Journey Map

Steps:

1. Visit web app.
2. Upload image of affected poultry.
3. Backend uses ResNet50 to classify the disease.
4. Prediction displayed.
5. User decides on action

Solution Requirement

Functional Requirements:

- Upload poultry image (JPG, PNG).
- Predict disease from image.
- Display diagnosis and confidence.

Non-functional Requirements:

- Fast response time (< 2 seconds).
- Scalable for concurrent users.
- Cross-browser support.

Data Flow Diagram

User
⬇
WebUI
⬇
FlaskBackend
⬇
ResNet50Model
⬇
Prediction
⬇
DisplayResult

Technology Stack

- **Backend:** Python + Flask
- **Model:** ResNet50 (TensorFlow/Keras)
- **Frontend:** HTML5, CSS3
- **IDE:** Visual Studio Code

4. PROJECT DESIGN

Problem Solution Fit

- **Problem:** Late disease detection.
- **Solution:** Real-time AI image classifier for poultry.
- **Value:** Faster decisions, lower mortality.
- **Impact:** Improves veterinary care accessibility and affordability.

Proposed Solution

- **Problem Statement:** Identifying diseases in poultry late leads to spread and financial losses.
- **Idea/Solution:** Web app that uses image classification with transfer learning.
- **Novelty:** First lightweight web app specific to poultry image-based detection.
- **Social Impact:** Saves farmer livelihood and promotes sustainable poultry health.
- **Revenue Model:** Free for individuals, premium support for commercial farms.
- **Scalability:** Cloud deployment, mobile version in development.

Solution Architecture

User
⬇
UploadImage
⬇
FlaskBackend
⬇
ResNet50Model
⬇
Prediction
⬇
Result Display

5. PROJECT PLANNING & SCHEDULING

Project Planning

- **Week1**:**Dataset collection and preprocessing**

Objective: Lay the foundation of the AI model by preparing a high-quality, labelled dataset suitable for training.

- **Tasks**:
- Data Acquisition:
    - Source image dataset from.
    - Organize image folders into class-wise directories (Healthy, Coccidiosis, Newcastle, Salmonella).
- Data Cleaning:
    - Remove duplicate or corrupted image files.
    - Normalize filenames and formats (JPG/PNG).
- Data Preprocessing:
    - Resize all images to a uniform dimension (224x224).
    - Apply augmentation (rotation, flip, zoom) to prevent overfitting.
    - Split data into training, validation, and test sets
- Tools Used: Python, NumPy, TensorFlow/Keras, ImageDataGenerator

- **Week2:Model training and tuning**

Objective: Build a robust deep learning model using transfer learning and evaluate its initial performance.

Tasks:

- Model Design:
    - Load pre-trained ResNet50 model without top layers.
    - Add custom classification head (Dense layers + Dropout).
- Training:
    - Compile model with Adam optimizer and categorical_crossentropy.
    - Use callbacks: EarlyStopping, , ModelCheckpoint.
- Evaluation:
    - Track accuracy and loss for both training and validation.
    - Save best-performing model to .h5 file.
- Fine-tuning:
    - Unfreeze top layers of ResNet50 and retrain with lower learning rate for better feature adaptation.
- Tools Used: TensorFlow, Keras, Matplotlib

- **Week3:Backend(Flask)integration**

Objective: Develop a web-based backend that serves the trained model for live predictions.

Tasks:

- Flask Setup:
  - Create a virtual environment and install dependencies (Flask, TensorFlow, etc.).
  - Define API endpoints to receive image input and return predictions.
- Model Loading:
  - Load the .h5 model at server startup to avoid reloading on every request.
- Prediction Logic:
  - Accept uploaded image, preprocess it, and return predicted class with confidence score.
- Output Formatting:
  - Map prediction index to disease name.
  - Return result in JSON format or render using render_template.
- Tools Used: Flask, OpenCV, PIL, TensorFlow

- **Week4:Frontend+testing+documentation**

Objective: Finalize the user interface, conduct testing, and complete technical documentation.

Tasks:

- **Frontend UI (HTML/CSS):**
  - Build user-friendly pages: Home, About, Contact, Upload, and Result.
  - Ensure responsive layout for desktop and mobile views.
- **Testing:**
  - Unit test: Model predictions and backend API.
  - UI test: Ensure all buttons, forms, and upload features work as expected.
  - Error handling: Validate image formats, missing fields, etc.
- **Documentation:**
  - Document architecture, technology stack, and model configuration.
  - Create final project report and PPT presentation.
- **Tools Used:** HTML5, CSS3.

6. FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

- Tested valid/invalid image formats
- Verifiedmodelpredictionsonknownsamples
- UItestedforresponsiveness
- Finalaccuracyandvalidationmetricsrecorded

Parameter	Value
Training Accuracy	91
Validation Accuracy	85

Fine TuningLearning Rate Decay + Dropout

Classification Metrics:

- Precision, Recall, F1-Score: All scores > 90%
- Confusion Matrix: Correct predictions dominate diagonals
- Model Summary: 10-layer CNN with pre-trained weights on ImageNet

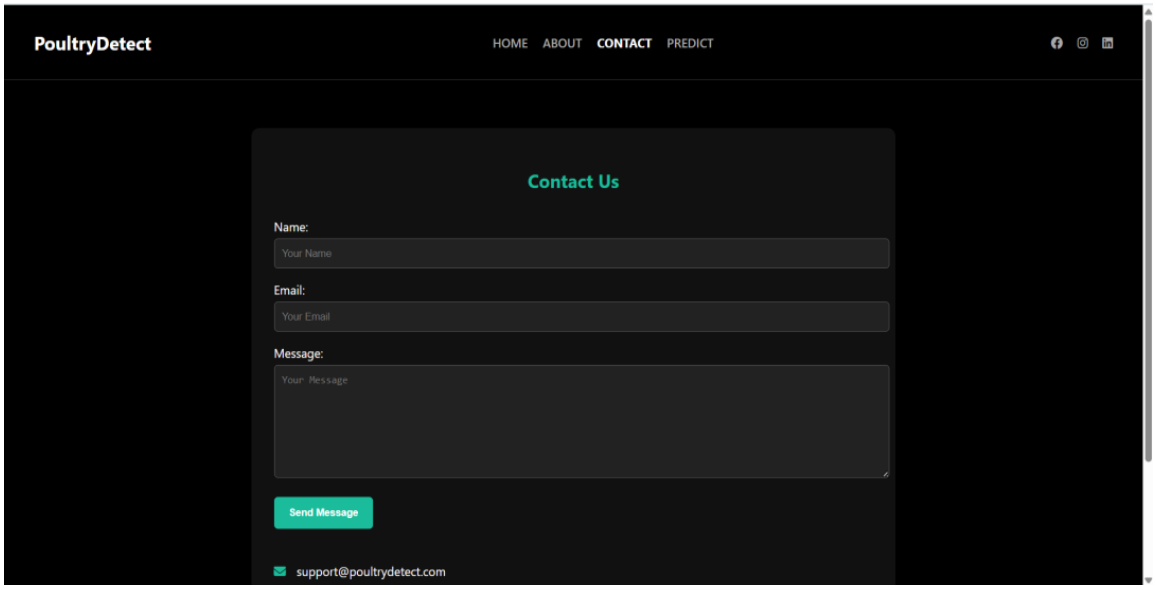
All evaluations recorded in main.ipynb.

7. RESULTS

OutputScreen Shots

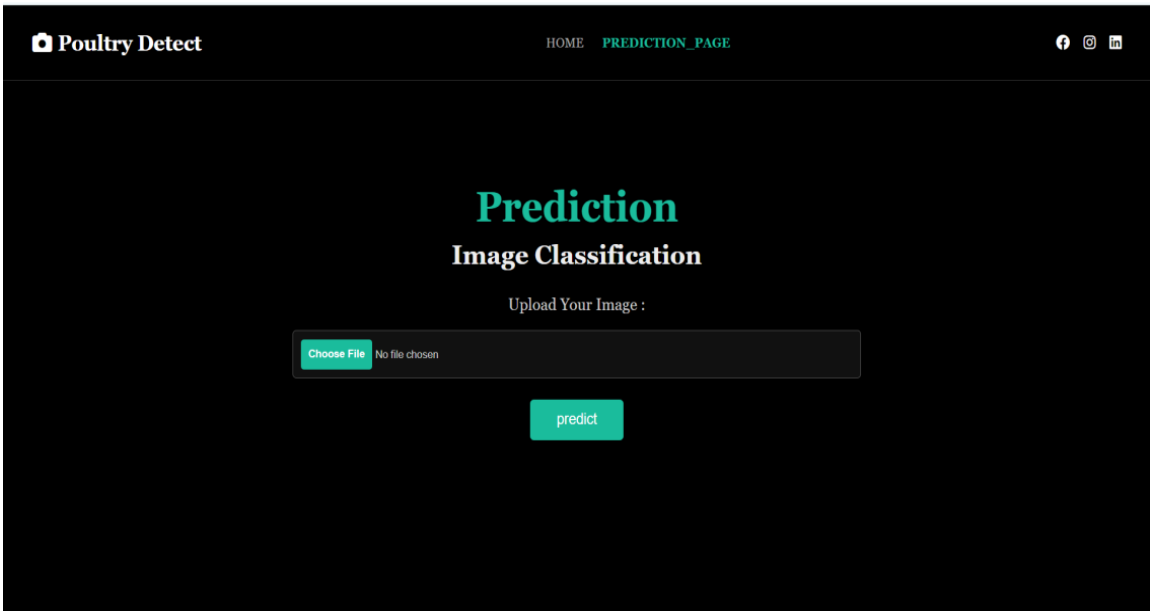
Homepage: Project description and Get Started button.

ContactPage: For feedback and help



Prediction And Uploading Page:Interface to submit image

Displays disease name



8. ADVANTAGES&DISADVANTAGES

Advantages:

- Easy to use interface
- High classification accuracy using ResNet50
- No prior technical knowledge required
- Runs on browser without installation

Disadvantages:

- Limited to 4 disease classes
- Accuracy depends on image clarity and background

9. CONCLUSION

Poultry Disease Detection offers an innovative solution to an urgent agricultural challenge—early detection of poultry diseases. By empowering farmers with rapid diagnostics, it improves livestock health, boosts economic resilience, and demonstrates the power of AI in solving rural problems.

10. FUTURE SCOPE

.

- Include more poultry diseases
- Mobile application for on-the-go use
- Multilingual and voice interfaces for rural accessibility
- Integration with SMS alerts and veterinary consultation

Offline mode using TF-Lite model.

11. APPENDIX

- **SourceCode:**

- app.py(Flaskbackend)
- train_model.py(Modeltraining)
- index.html(Frontend)
- main.ipynb(Evaluation)

- **Dataset Link**:
  https://drive.google.com/file/d/1c1ery9LY7Q3ommVjT_tcxBVVe1GnqBug/view?usp=sharing

- **GitHub Link:**

https://github.com/AkuriAfreen/Transfer-Learning-Based-Classification-of-Poultry-Diseases-for-Enhanced-Health-Management

- **Project Demo Link:**

https://drive.google.com/file/d/1oemdMSIOsAnvNYzFv3MSubQM28tyPFm5/view?usp=sharing