

Artificial Intelligence And Machine Learning

Project: 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 Members:

1. Chitturi Jaya Sree
2. Chennakesavula Yuva Ramchandra
3. Chembeti Sravanthi
4. Chikkavarapu Harika

2. Project Overview

Purpose:

The purpose of this project is to provide poultry farmers and veterinarians with an intelligent web-based diagnostic tool that can quickly and accurately identify poultry diseases from images. Leveraging the power of deep learning and transfer learning with the ResNet50 architecture, the system enables early disease detection to minimize mortality and economic loss.

Features:

- Upload poultry images for automated disease classification
- Real-time prediction using a pre-trained deep learning model
- User-friendly interface for seamless navigation
- Informative About and Contact pages
- Mobile-responsive design for ease of use on various devices

3. Architecture

Frontend:

The frontend is developed using HTML and CSS templated through Flask's Jinja engine. It offers a clean and intuitive interface for image uploads, result viewing, and basic navigation across Home, About, and Contact sections.

Backend:

The backend is powered by Python's Flask framework. It handles image processing, interaction with the trained ResNet50 model, and routing between pages. It ensures fast and reliable predictions by preprocessing input images and passing them through the model.

Database:

Currently, the system operates without a database. However, it can be extended in the future to include a MongoDB or SQLite database for storing user data, feedback, and prediction history.

4. Setup Instructions

Prerequisites:

- Python 3.x
- Flask
- TensorFlow / Keras
- NumPy, PIL

Installation & Execution:

```
git clone https://github.com/Jaya1718/Transfer-Learning-Based-
Classification-of-Poultry-Diseases-for-Enhanced-Health-Management
cd project-folder
pip install -r requirements.txt(installations)
python app.py(execution)
```

5. Folder Structure

Client Side (Templates and Static Files):

```
/templates
└── index.html
```

- └── about.html
- └── contact.html
- └── predict.html

/static/uploads
/static/css (optional)

Server Side:

app.py # Main Flask backend
best_model.h5 # Pre-trained ResNet50 model file

6. Running the Application

Launch the Flask server by running:

python app.py

This will serve both frontend and backend through the same Flask application.

7. API Documentation

Endpoint: /predict

Method: POST

Request Parameters:

- pc_image – Uploaded poultry image file

Response:

- Renders predict.html displaying the uploaded image and predicted disease class.

8. Authentication

Current State:

No authentication or authorization is implemented.

Future Scope:

Authentication mechanisms such as admin login or user registration can be integrated using JWT tokens or session-based login systems.

9. User Interface

The application includes a clean and responsive user interface designed for accessibility and ease of use. Users can:

- Upload an image for analysis

- View real-time predictions
- Navigate across informative pages (About, Contact)

The interface supports both desktop and mobile browsers.

10. Testing

The application underwent the following testing processes:

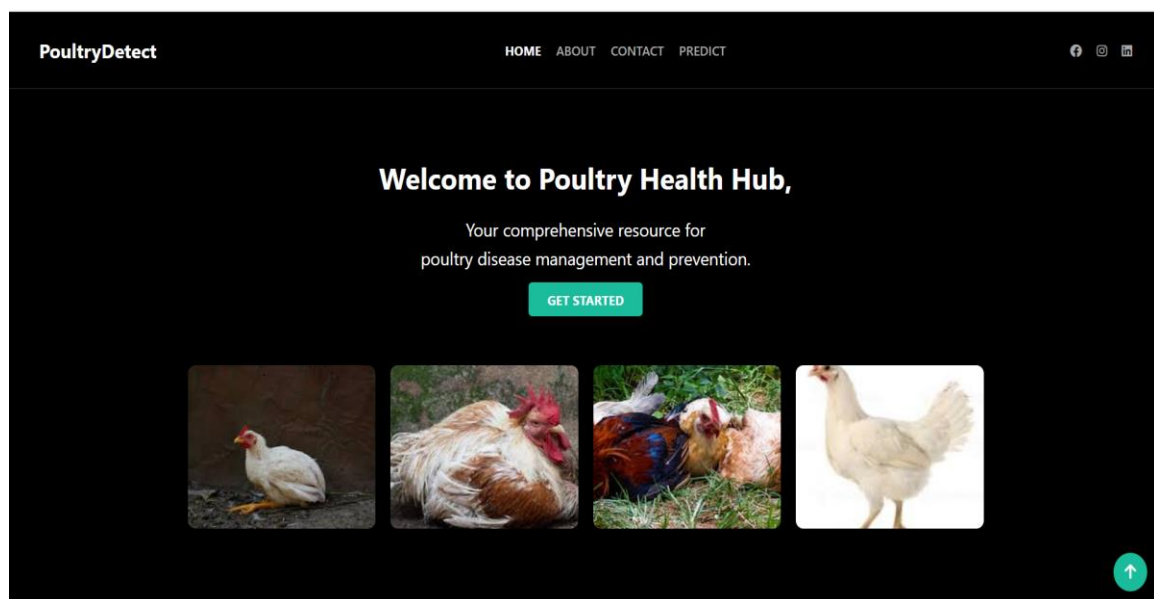
- Image Validity Testing – Ensured the system handles valid and invalid image formats gracefully.
- Model Verification – Cross-checked predictions on known labeled samples.
- UI Responsiveness – Verified compatibility across different devices and screen sizes.

11. Screenshots or Demo

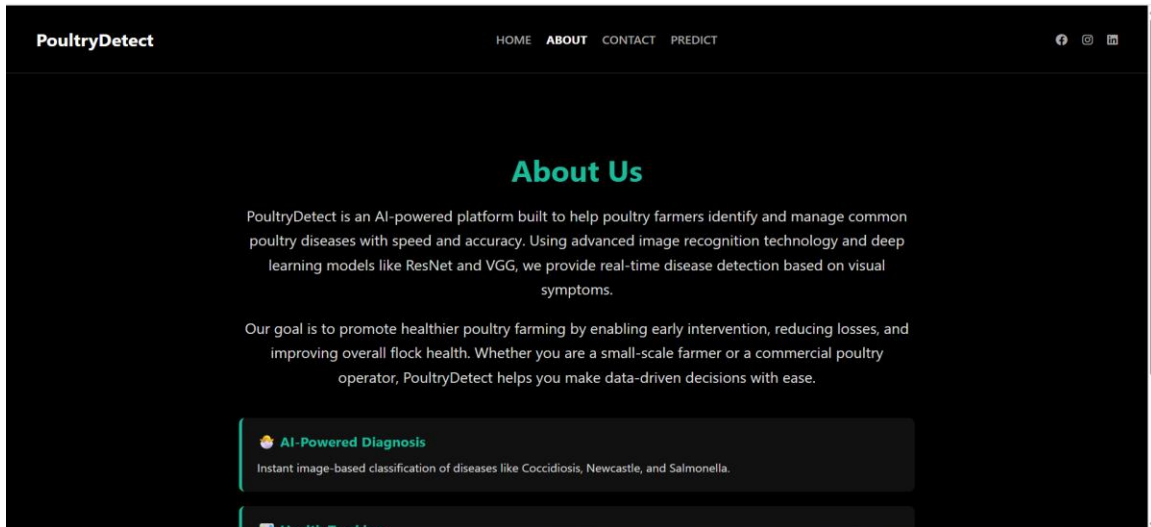
Key interface screenshots include:

- Homepage
- About Page
- Contact Form
- Prediction Output Page

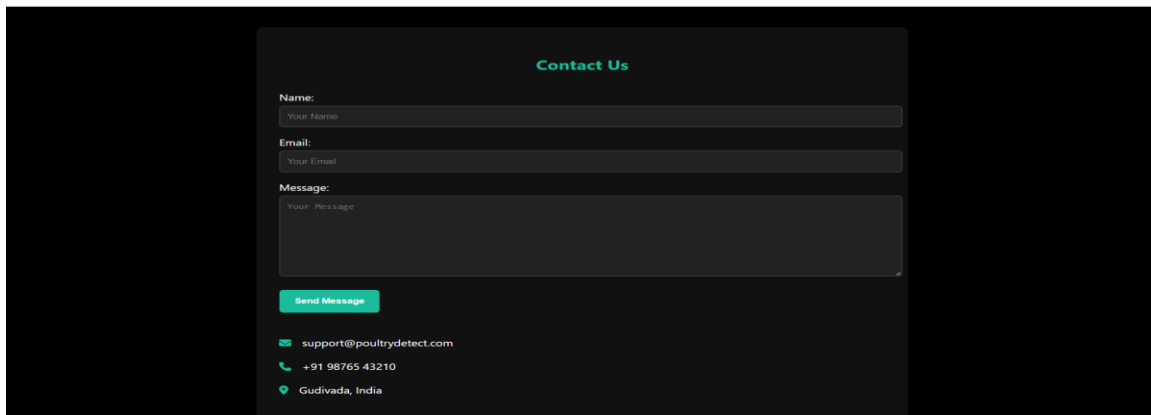
Home page



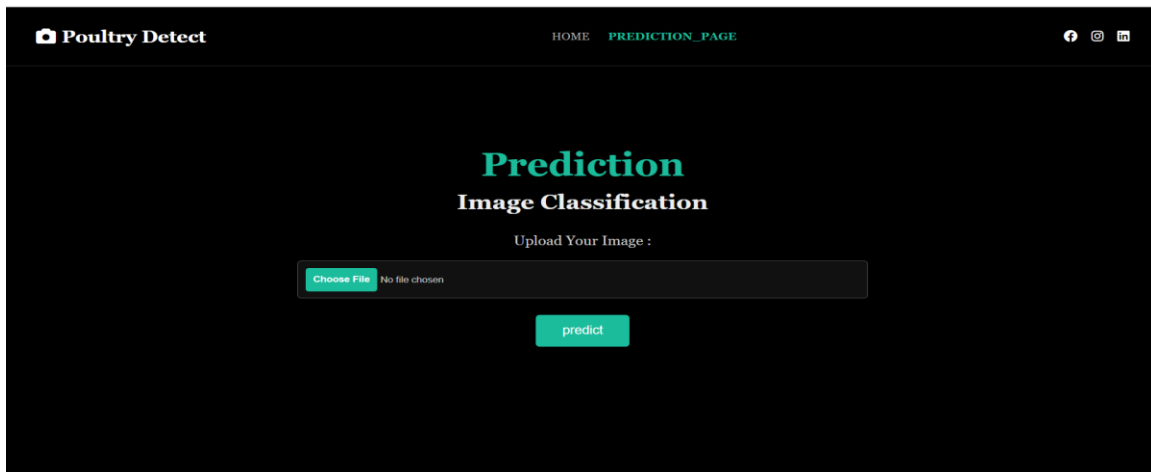
About Page

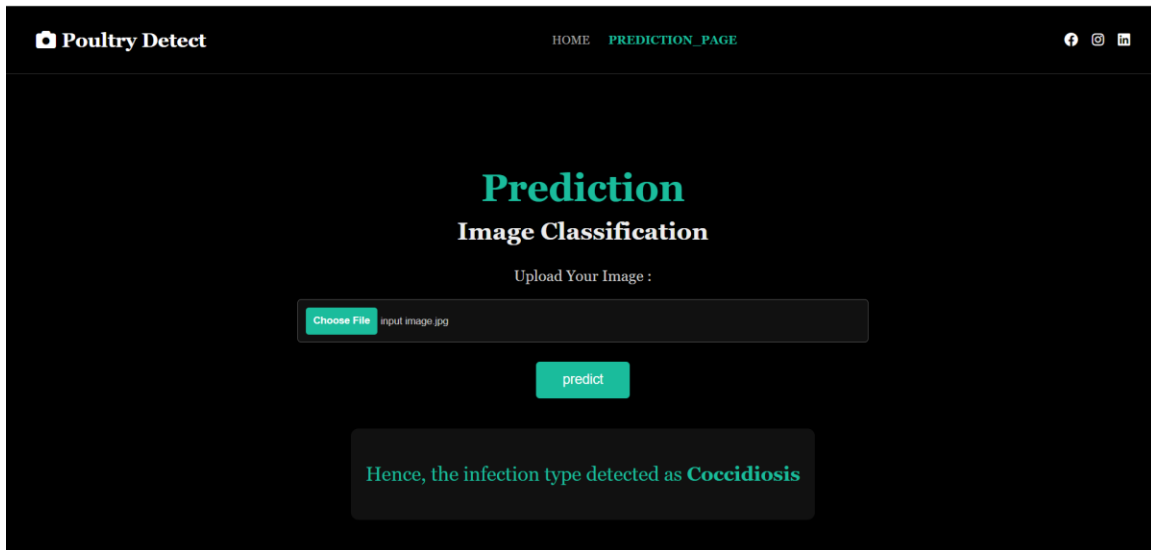


Contact Page



Prediction and Result Page





12. Known Issues

- The current model supports only four poultry disease classes.
- Image quality may impact prediction accuracy.
- No persistent storage or database functionality is available at present.

13. Future Enhancements

- Extend classification support to more disease types.
- Integrate cloud deployment (AWS, GCP) for global access.
- Add user authentication and feedback features.
- Build a mobile application version for farmers in remote areas.
- Incorporate voice-guided support and disease treatment suggestions.