1. Introduction

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

• Team ID: LTVIP2025TMID46520

• Team Members:

- Harshita – Application Developer

- Dinesh Kanna Saragadam – Model Trainer

- Veerendra – Backend

- Yaswanth – Frontend

2. Project Overview

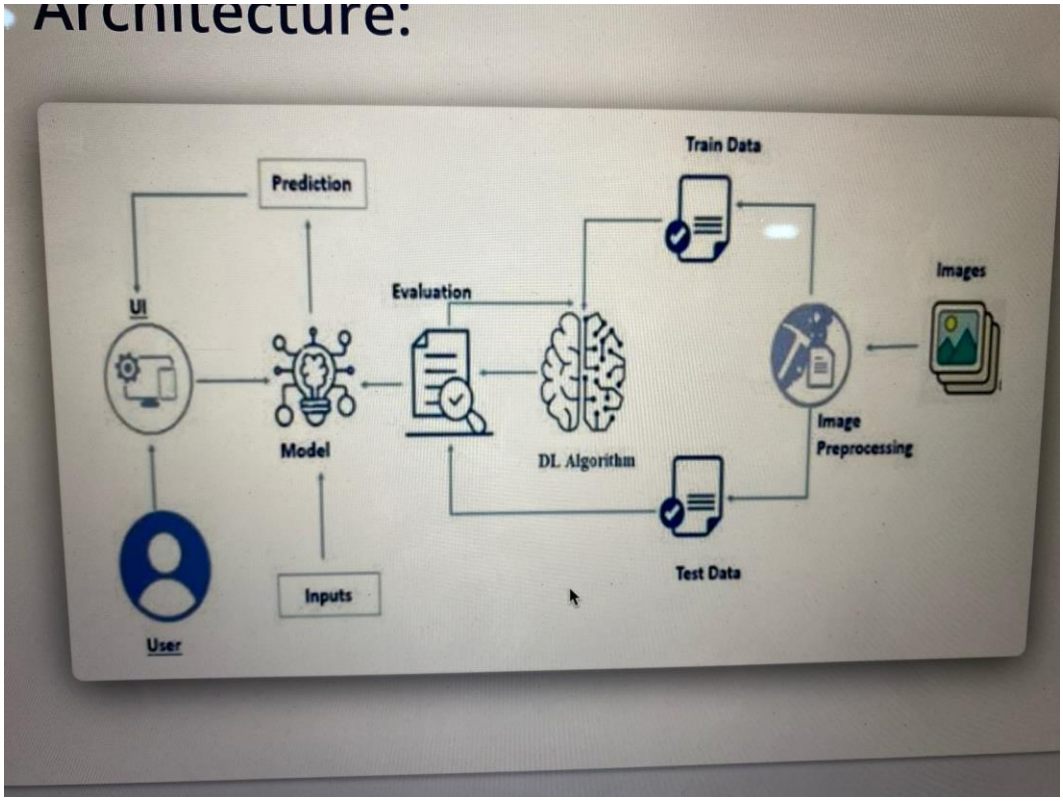
• Purpose:

This project aims to assist poultry farmers and veterinarians by automatically detecting common poultry diseases from chicken images using deep learning. It uses Transfer Learning and provides a Flask-based web interface.

• Features:

- Image upload functionality
- Instant disease prediction
- Support for 4 categories: Salmonella, Newcastle Disease, Coccidiosis, and Healthy
- HTML-based UI
- Flask-powered backend
- TensorFlow model integration

3. Architecture



System Architecture Diagram:

• Frontend: HTML5, CSS3 web page using Flask template engine (Jinja2).

• Backend: Python Flask application that loads a pretrained VGG16 model and handles prediction logic.

• Database: No external database is used. Model predictions are processed on-the-fly from uploaded images.

4. Setup Instructions

• Prerequisites:

- Python 3.x
- Anaconda Navigator
- Required packages: numpy, pandas, tensorflow, flask, etc.

• Installation:

- Clone/download the project
- Open Anaconda Prompt and navigate to the folder

- Install packages using pip
- Run `python app.py` to launch the Flask app

5. Folder Structure

• Project Directory:

```
poultry-project/
├── app.py
├── healthy_vs_rotten.h5
├── static/uploads/
└── templates/
    ├── index.html
    └── result.html
```

6. Running the Application

• Step 1: Open terminal and navigate to poultry-project directory

• Step 2: Run `python app.py`

• Step 3: Open browser and go to http://127.0.0.1:5000
• Step 4: Upload a chicken image and view prediction

7. API Documentation

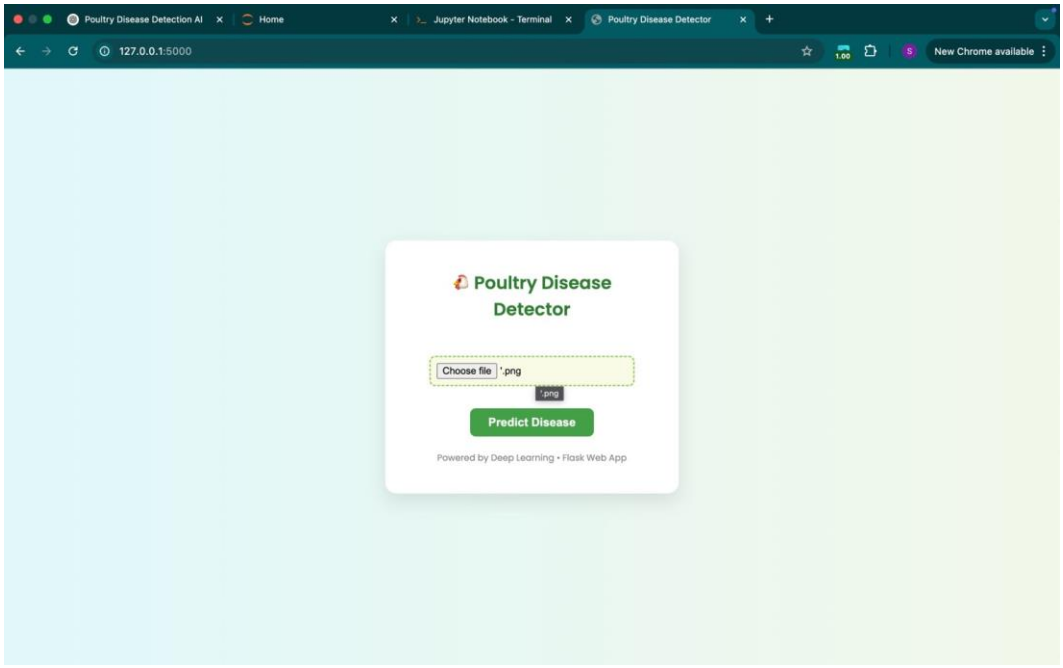
• POST `/predict` – Handles image file upload and returns predicted label

• GET `/` – Loads upload form (index.html)

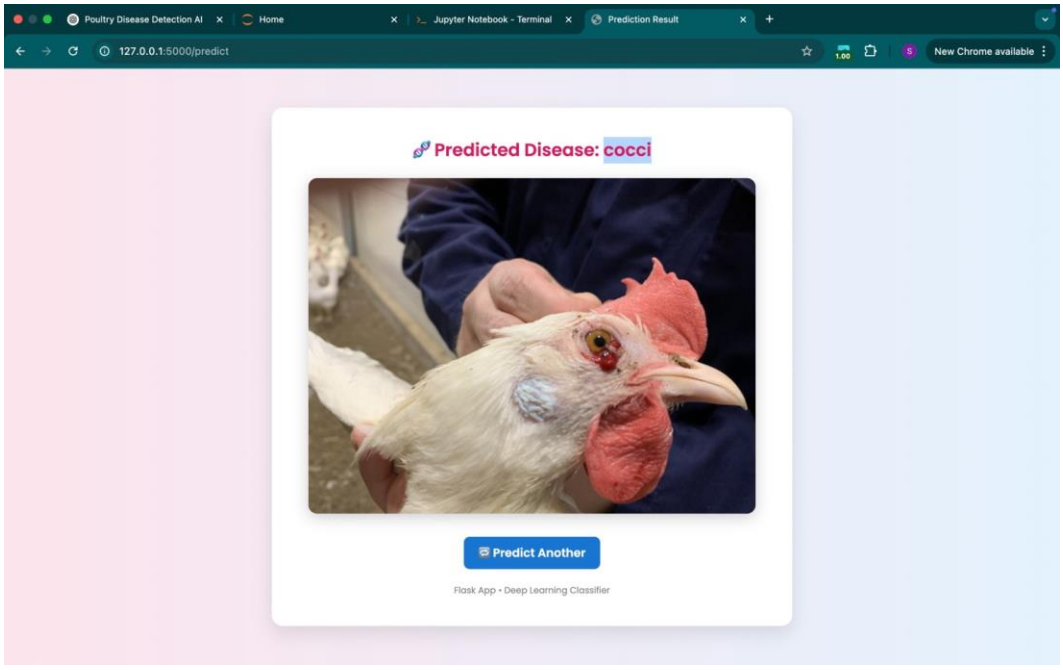
8. Authentication

• Not required. All users can access the prediction page without login.

9. User Interface



index.html – File upload interface



result.html – Displays prediction result

• index.html – File upload interface

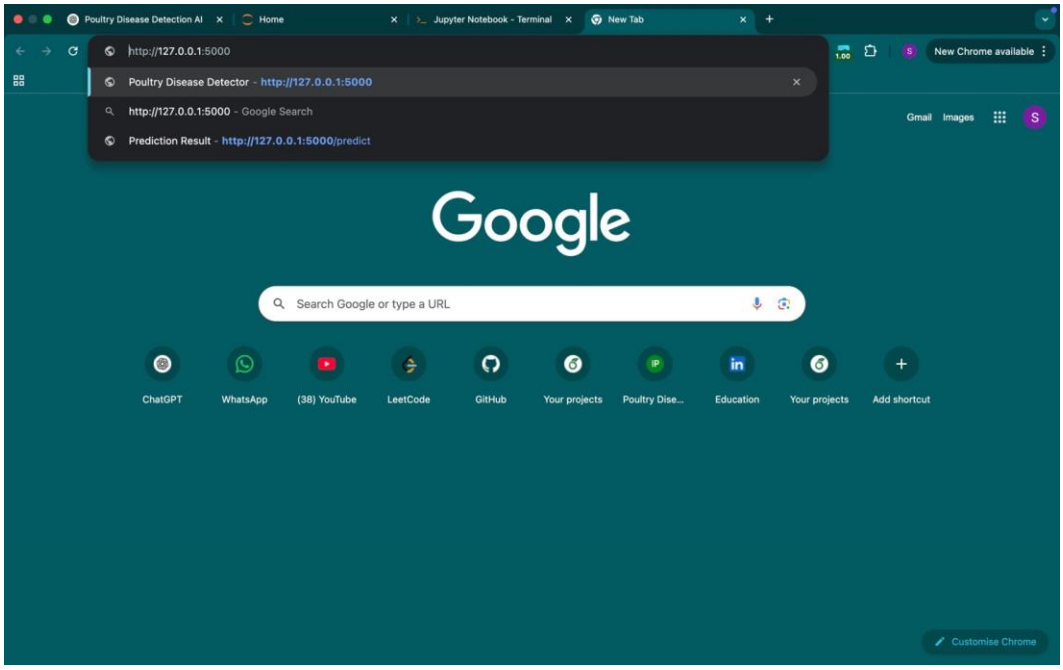
• result.html – Displays prediction result and image

10. Testing

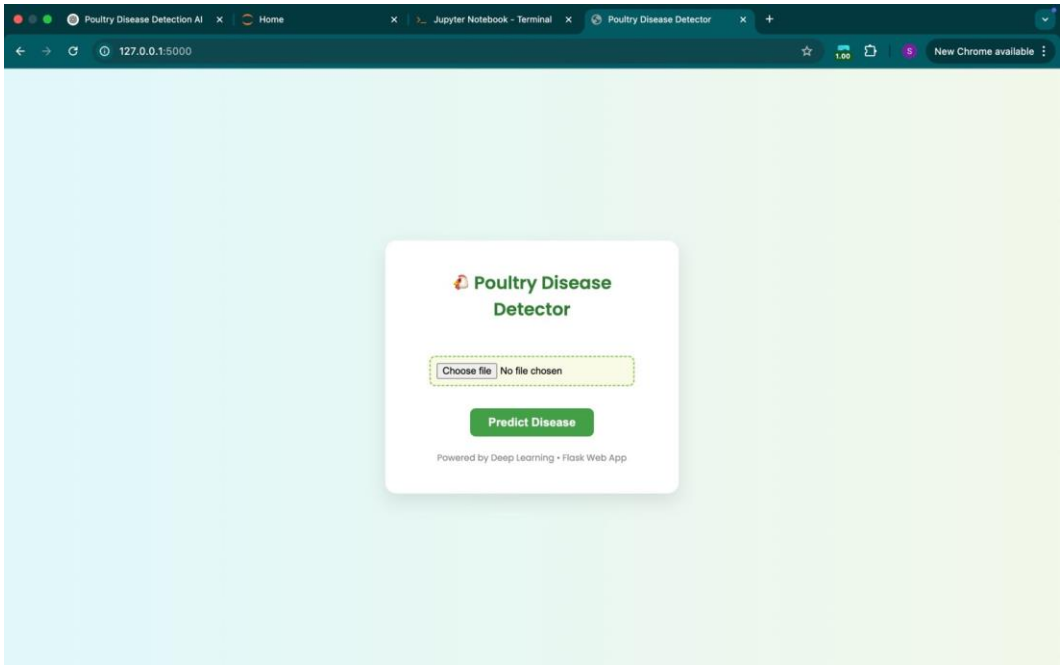
• Manual testing using poultry images from dataset

• Accuracy and confusion matrix evaluated using test data

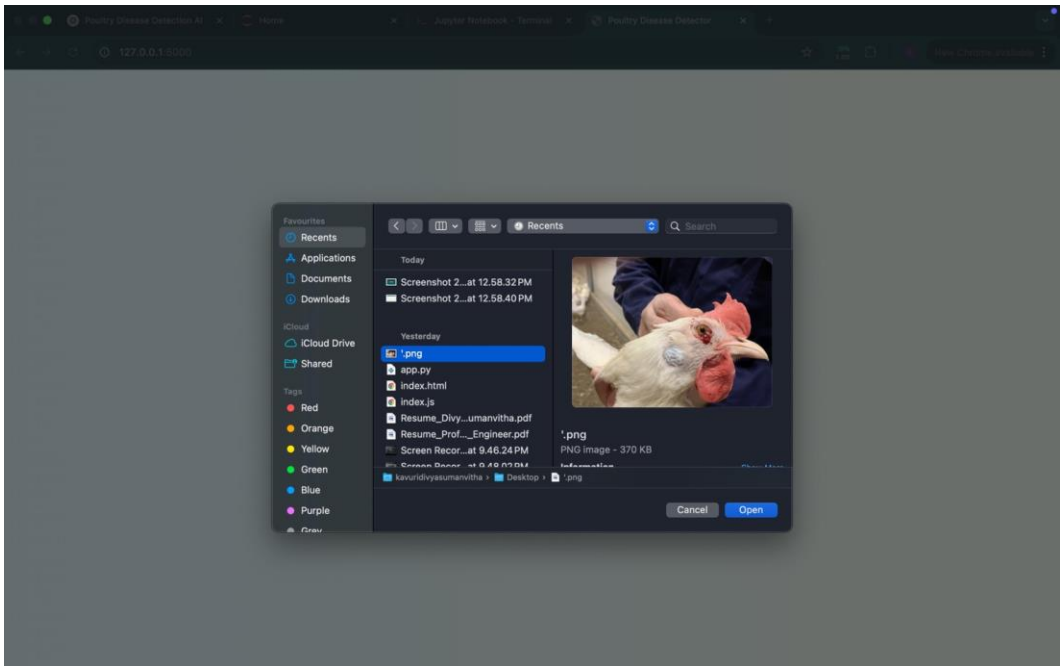
11. Screenshots or Demo



Launching the application in browser



User Interface – Upload Form



Selecting an image to upload

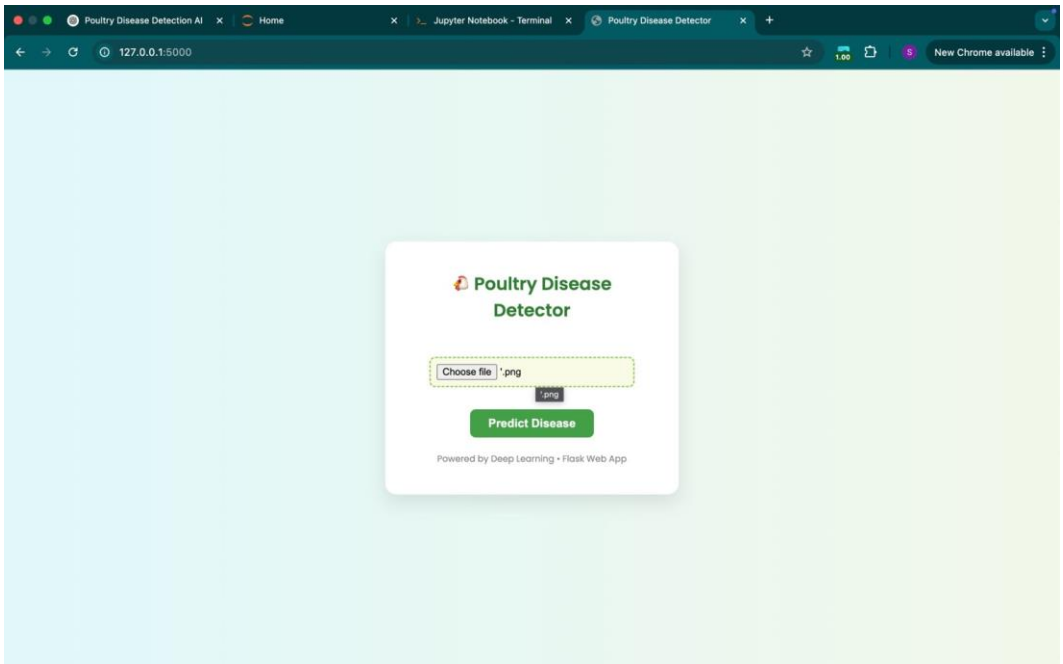
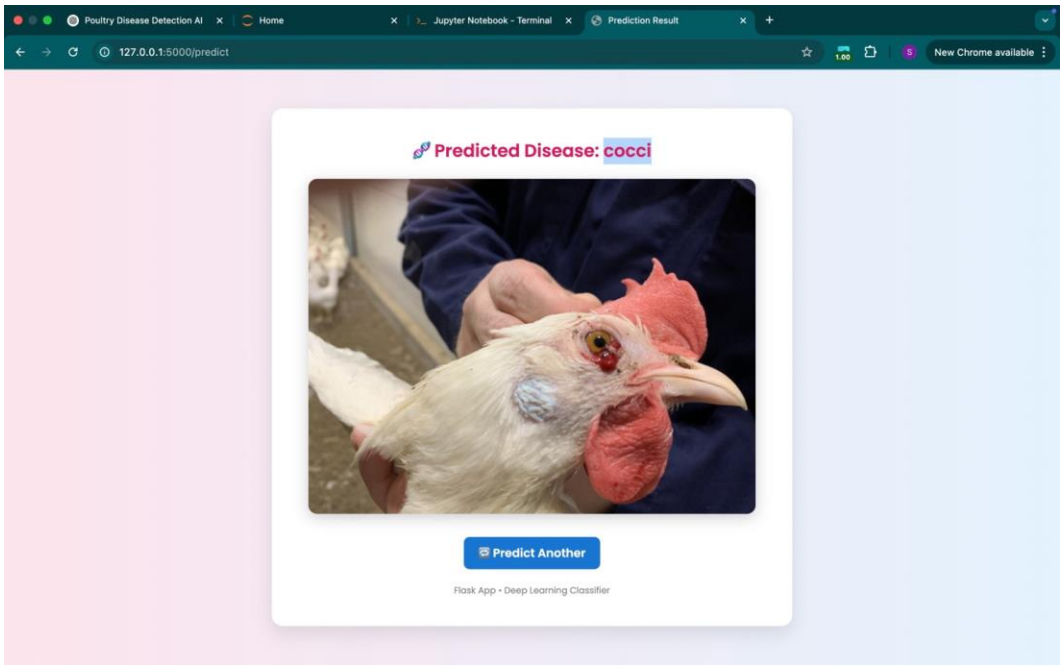


Image uploaded for prediction



Prediction result: Hen diagnosed with Coccidiosis

12. Known Issues

- No camera integration
- Dataset limited to 4 categories
- Predictions are dependent on image quality

13. Future Enhancements

- Add camera capture
- Expand dataset for more diseases
- Add confidence score
- Deploy to cloud
- Add multilingual/voice feedback