PoultryVision: Smart Poultry Disease Detection Using Transfer Learning

Project Overview

PoultryVision is an AI-powered image classification system that uses transfer learning to detect and classify poultry diseases. The model identifies diseases like Coccidiosis, Newcastle Disease, Salmonella, and Healthy condition using MobileNetV2. The trained model is deployed using Flask, allowing users to upload poultry images and receive predictions.

Use Cases and Scenarios

Scenario 1: On-the-Field Poultry Diagnosis

Farmers can take photos of sick birds and receive predictions to begin treatment.

Scenario 2: Poultry Farm Management

Commercial farms can use the tool for daily health checks and isolate infections.

Scenario 3: Veterinary Education & Training

Students can identify diseases through images and improve diagnostic skills.

Project Objectives

- Understand image classification with CNNs
- Implement a MobileNetV2-based transfer learning model
- Deploy model using Flask
- Enable real-time poultry health prediction

Prerequisites

Software: Python 3.10+, VS Code, Browser

Libraries: flask, tensorflow, numpy, pillow

Knowledge: CNNs, Keras, Flask basics

Project Flow

1. User uploads an image via the web app

2. Flask routes image to backend

3. Image is preprocessed

4. Model predicts disease class

5. Result displayed on the page

Dataset

- Source: Roboflow
- Classes: Coccidiosis, Salmonella, Newcastle, Healthy
- ~7000 images, JPEG format, 224x224 resolution

Implementation Steps

Project Structure:

??? app.py

??? healthy_vs_rotten.h5

??? static/uploads/

??? templates/index.html, blog.html, etc.

Key Steps:

- Train MobileNetV2 model
- Save as .h5
- Build Flask UI
- Run locally and test predictions

Features Summary

- Real-time image classification
- Flask-based deployment
- MobileNetV2 for performance
- Upload preview and accurate results

Backend Logic (Flask)

- Load .h5 model

- Save uploaded image
- Resize and normalize
- Predict class
- Show result

Final Summary

PoultryVision shows how deep learning supports smarter farming. It helps farmers, educators, and vets identify diseases quickly using a simple, accessible AI tool.

Flask App Code (app.py)

```
from flask import Flask, render_template, request
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import os

app = Flask(__name__)
model = load_model("healthy_vs_rotten.h5")
class_names = ['Coccidiosis', 'Healthy', 'New Castle Disease', 'Salmonella']

@app.route('/')
def home():
    return render_template("index.html")

@app.route('/predict', methods=['POST'])
def predict():
    if 'file' not in request.files:
        return render_template("index.html", prediction="No file uploaded.")
    file = request.files['file']
    if file.filename == '':
        return render_template("index.html", prediction="No file selected.")
    file_path = os.path.join("static", file.filename)
    file.save(file_path)
    img = image.load_img(file_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0) / 255.0
    prediction = model.predict(img_array)
    predicted_class = class_names[np.argmax(prediction[0])]
    return render_template("index.html", prediction=predicted_class)

if __name__ == '__main__':
    app.run(debug=True)
```

PoultryVision: Smart Poultry Disease Detection Using Transfer Learning

HTML Template (index.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Poultry Disease Classifier</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body>
    <div class="container text-center mt-5">
        <h1>Poultry Disease Classifier</h1>
        <form action="/predict" method="post" enctype="multipart/form-data" class="mt-4">
            <input type="file" name="file" accept="image/*" class="form-control mb-3" required>
            <button type="submit" class="btn btn-success">Predict</button>
        </form>
        {% if prediction %}
            <h3 class="mt-4">Prediction: {{ prediction }}</h3>
        {% endif %}
    </div>
</body>
</html>
```