Poultry Disease Image Classification Web App

Overview

This project is a deep learning-based web application for classifying poultry diseases from images. It uses a MobileNetV2 model trained on poultry disease images and provides a user-friendly web interface for predictions.

Features

- Image classification for four poultry disease categories:
  - Coccidiosis
  - Healthy
  - New Castle Disease
  - Salmonella
- Flask web app for uploading images and viewing predictions
- REST API endpoint for programmatic predictions

Project Structure

```
Int_project/
│
├── Poultry Disease Classifier/
│   ├── train_image_model.py                # Model training script
│   ├── poultry_disease_image_model.h5      # Trained Keras model
│   ├── app.py                              # Flask backend (alternative location)
│   └── ...
│
├── app.py                                  # Flask web app (main)
├── templates/
│   └── index.html                          # HTML template for web interface
├── requirements.txt                        # Python dependencies
└── ...
```

How It Works

1. **Model Training**
   - The script `Poultry Disease Classifier/train_image_model.py` trains a MobileNetV2 model on poultry disease images organized in subfolders by class.
   - The trained model is saved as `poultry_disease_image_model.h5`.

2. **Web Application**
   - The Flask app (`app.py`) loads the trained model and provides a web interface for image upload and prediction.
   - Users can upload an image, and the app will display the predicted disease class.

3. **API Endpoint**
   - The `/predict` endpoint accepts POST requests with an image file and returns the predicted class as JSON.

Usage

1. Install Requirements

pip install -r requirements.txt

2. Train the Model (if not already trained)

python Poultry Disease Classifier/train_image_model.py

3. Run the Web App

python app.py

- Open your browser and go to `http://127.0.0.1:5000/`
- Upload an image to get a prediction.

4. Use the API Endpoint

Send a POST request to `/predict` with an image file:
bash
curl -X POST -F "file=@path_to_image.jpg" http://127.0.0.1:5000/predict

Data Preparation

- Organize your dataset as follows:

Poultry Disease Classifier/data/data/
train/
Coccidiosis/
Healthy/
New Castle Disease/
Salmonella/
test/
Coccidiosis/
Healthy/
New Castle Disease/
Salmonella/

- Each subfolder should contain images for that class.

Requirements

- Python 3.8+
- TensorFlow
- Flask
- Pillow
- numpy

Notes

- The model expects images of size 224x224 pixels.
- The class order in `CATEGORIES` must match the order used during training.
- For production, use a production-ready WSGI server instead of Flask's built- in server.

Author

Abhishek