Rice Type Classifier

A web application that uses deep learning to classify rice grain images into five varieties: Arborio, Basmati, Ipsala, Jasmine, and Karacadag.

Features

- Upload a rice grain image and get instant prediction of its type.
- Displays prediction confidence.
- Modern, responsive user interface.

Project Structure

```
Rice_Image_Dataset/
├── app.py                # Flask web application
├── predict_rice_type.py    # (Optional) Standalone prediction script
├── rice_type_model.h5      # Trained Keras model
├── train_rice_model.py     # Model training script
├── templates/
│   └── index.html          # Main HTML template
├── static/
│   └── uploads/            # Uploaded images
├── Arborio/                # Sample images for Arborio rice
├── Basmati/                # Sample images for Basmati rice
├── Ipsala/                 # Sample images for Ipsala rice
├── Jasmine/                # Sample images for Jasmine rice
├── Karacadag/              # Sample images for Karacadag rice
```

Setup Instructions

1. Install Requirements

Make sure you have Python 3.7+ installed. Install dependencies:

```
bash
pip install flask tensorflow numpy pillow werkzeug
```

2. Model File

Ensure `rice_type_model.h5` (the trained model) is present in the project root. You can train your own model using `train_rice_model.py`.

3. Run the Application

```
bash
python app.py
```

The app will be available at `http://127.0.0.1:5000/`.

4. Usage

→ Open the web app in your browser.
→ Click "Choose Image" and select a rice grain image.
→ Click "Predict" to see the predicted rice type and confidence.

Model Training (Optional)

To retrain the model, use `train_rice_model.py`. Make sure your dataset folders (Arborio, Basmati, etc.) are structured with images inside.

File Descriptions

1. **app.py:** Main Flask app for web interface and prediction.
2. **templates/index.html:** User interface for uploading and viewing results.
3. **rice_type_model.h5:** Pre-trained Keras model for rice classification.
4. **train_rice_model.py:** Script to train a new model.
5. **static/uploads/:** Stores uploaded images for prediction.

Notes

- Only image files (`.jpg`, `.jpeg`, `.png`) are accepted.
- Uploaded images are stored in `static/uploads/`.
- The UI provides a live preview of the selected image before upload.

Acknowledgments

- TensorFlow/Keras for deep learning
- Flask for the web framework
- Dataset: [Provide source if public]

Author

Gajjela Bharath