GrainPalette – A Deep Learning Odyssey in Rice Type Classification Through Transfer Learning

1. Introduction

Project Title:

GrainPalette – A Deep Learning Odyssey in Rice Type Classification Through Transfer Learning

Team Members:

Pothannaboyina Anil Kumar (Developer)

2. Project Overview

Purpose:
To utilize transfer learning techniques for building an intelligent system capable of classifying different types of rice grains using deep learning models, ensuring accuracy and efficiency in agricultural product identification.

Features:

- Image-based rice grain classification
- Deep learning model trained via transfer learning (e.g., ResNet, MobileNet)
- Web-based interface for uploading and analyzing images
- Fast and accurate predictions
- Visual result display with confidence scores

3. Architecture

Frontend:
Built using HTML, CSS, and JavaScript for a user-friendly interface.

Backend:
Python with FastAPI, responsible for model inference and serving predictions.

Machine Learning:
Pre-trained CNN models (ResNet/MobileNet) fine-tuned on rice grain

Database (Optional):

For storing image logs and classification results (MongoDB or SQLite).

4. Setup Instructions

Prerequisites:

- Python 3.10+
- Pre-trained model file (e.g., .h5 or .pt)
- FastAPI, Uvicorn, OpenCV, TensorFlow/Keras or PyTorch
- Basic knowledge of HTML/CSS/JS

Installation:

1. Clone the repository
2. Create a virtual environment
3. Install dependencies: pip install -r requirements.txt
4. Place the trained model file in the specified path
5. Run using: uvicorn main:app --reload

5. Folder Structure

Client:

- index.html
- style.css
- script.js

Server:

- main.py
- model_predict.py
- requirements.txt
- rice_model.h5 (or .pt depending on framework)

6. Running the Application

Frontend:

Open index.html in a web browser to upload rice grain images.

Backend:

Navigate to the server directory and run:

bash

CopyEdit

```
uvicorn main:app --reload
```

7. API Documentation

Endpoint: /predict
Method: POST
Description: Accepts image file input and returns the predicted rice type.

Example Request (multipart/form-data):

json

CopyEdit

```
{

 "image": "<uploaded_file>"

}
```

Example Response:

json

CopyEdit

```
{

 "prediction": "Basmati",

 "confidence": "95.3%"

}
```

8. Model and Inference

- Transfer learning used on rice dataset

- Models like ResNet50, MobileNetV2 fine-tuned for grain classification
- Image preprocessing includes resizing, normalization, augmentation

9. User Interface

- Simple UI with file upload feature
- Displays predicted rice type and confidence score
- Responsive design for mobile and desktop

(Screenshots or UI GIFs can be added here)

10. Testing

Strategy:

- Unit testing of model inference API
- Manual testing of image input-output
- Performance benchmarking using dummy and real datasets

11. Screenshots or Demo

- Include screenshots of terminal output, model results, and UI
- Demo link: GitHub.com/MKShiva/GrainPalette *(replace with actual)*

12. Known Issues

- Model may misclassify similar-looking grain types
- Requires high-quality image input
- Limited rice varieties (based on training data)

13. Future Enhancements

- Real-time image classification using mobile camera
- Support for additional grains/crops
- Integration with agricultural recommendation systems
- Multilingual support and voice interaction