

GrainPalette - AIML Project Documentation

1. Introduction

Project Title: GrainPalette - A Deep Learning Odyssey In Rice Type Classification

Through Transfer Learning Team Members

1.L Guravaiah

2. M Kusuma Priya

3. M Tejaswi

4. M Harika reddy

GrainPalette is an advanced AI-powered image classification system focused on identifying different types of rice grains.

Leveraging the power of Transfer Learning, this project aims to revolutionize the agricultural sector by enhancing the speed, accuracy, and scalability of rice grain type identification.

2. Architecture

- Model: Transfer Learning with ResNet50 fine-tuned on a labeled dataset of rice types.
- Pipeline:
 1. Image preprocessing (resizing, normalization)
 2. Feature extraction
 3. Classification via dense layers with softmax

3. Training and Inference

Training:

- Dataset split: training, validation, test
- Augmentation: rotation, zoom, flip
- Early stopping and model checkpointing used

Inference:

- Accepts an image, processes it, and outputs rice type with confidence score

GrainPalette - AIML Project Documentation

4. Setup Instructions

Setup Instructions:

- Prerequisites: Python 3.7+, TensorFlow/Keras, OpenCV, NumPy, Flask

or CSS-based UI - Steps:

1. Clone repo
 2. Create virtual env
 3. pip install -r requirements.txt
 4. Prepare dataset
 5. Run training script
 6. Launch app using: open index.html (CSS UI) or python app.py
5. Github Link: <https://github.com/Guravaiah1/GrainPalette--A-Deep-Learning-Odyssey-In-Rice-Type-Classification>

6. Folder Structure

Folder Structure:

- /dataset: Labeled rice images
- /model: Training scripts and models
- /app: UI and API logic (CSS or Flask)
- /utils: Helper functions
- requirements.txt, README.md

7. Running the Application

Running the Application:

1. Install dependencies
2. Activate environment
3. Launch:
 - CSS UI: open index.html
 - Flask: python app.py

GrainPalette - AIML Project Documentation

8. API

Documentation API

Documentation:

- POST /predict: Accepts image, returns rice type
- GET /model-info: Returns model details
- POST /retrain: (Future scope)

9. Authentication and UI

Authentication:

- None in current version- Future: JWT login, admin data upload

User Interface:

- Built using CSS for responsive layout
- Upload image component
- Displays predictions

10. Testing

Testing:

- Manual testing on diverse images
- Metrics: Accuracy, precision, recall, F1
- Confusion Matrix used
- Future: CI/CD with pytest

11. Screenshot or Demo

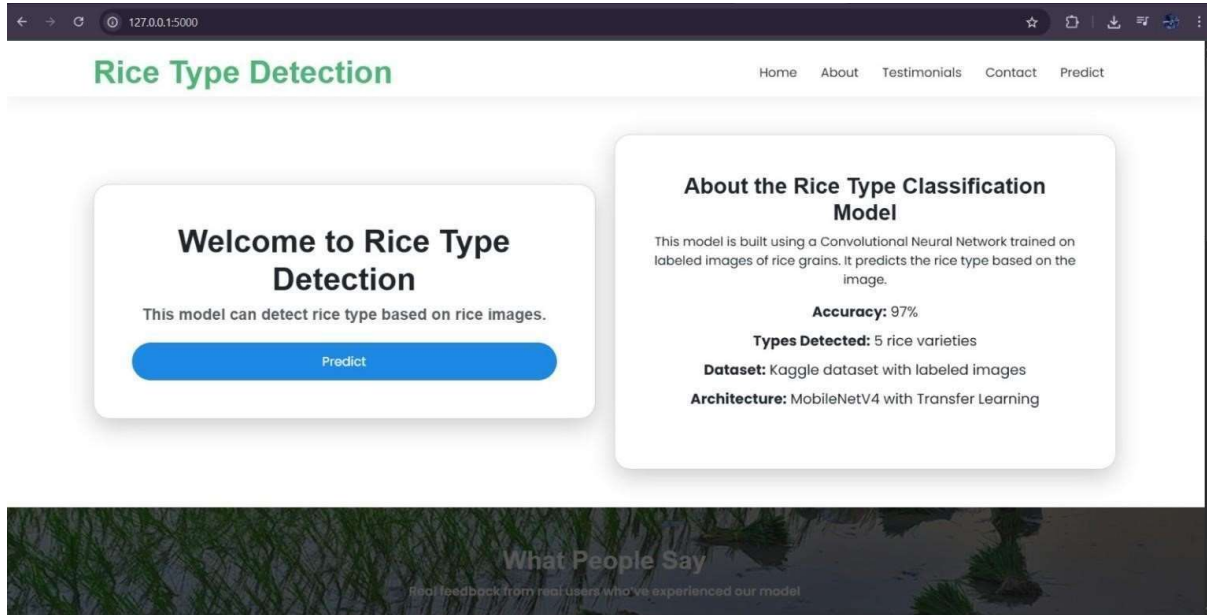
Screenshots and Demo:

Below are UI screenshots of the GrainPalette rice classification system built using HTML + CSS.

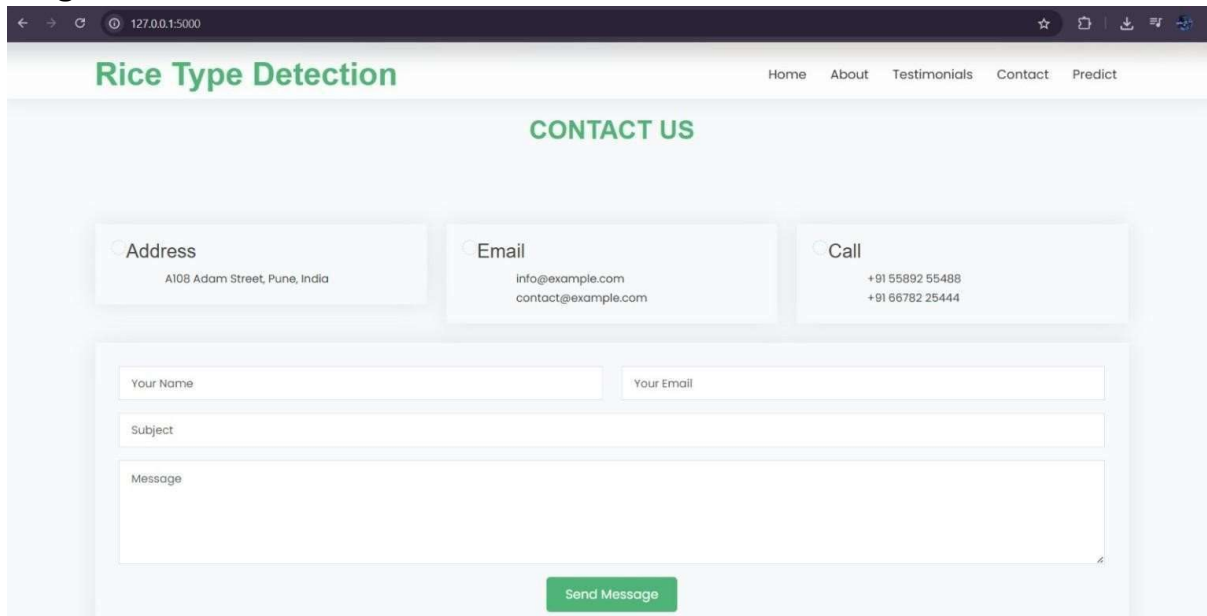
Demo Video: <https://drive.google.com/file/d/1Q0e1qsShrl7sCnCB2VKwTLnYVFRLn-OB/view?usp=drivesdk>

GrainPalette - AIML Project Documentation

Screenshot 1: Home and Model Info

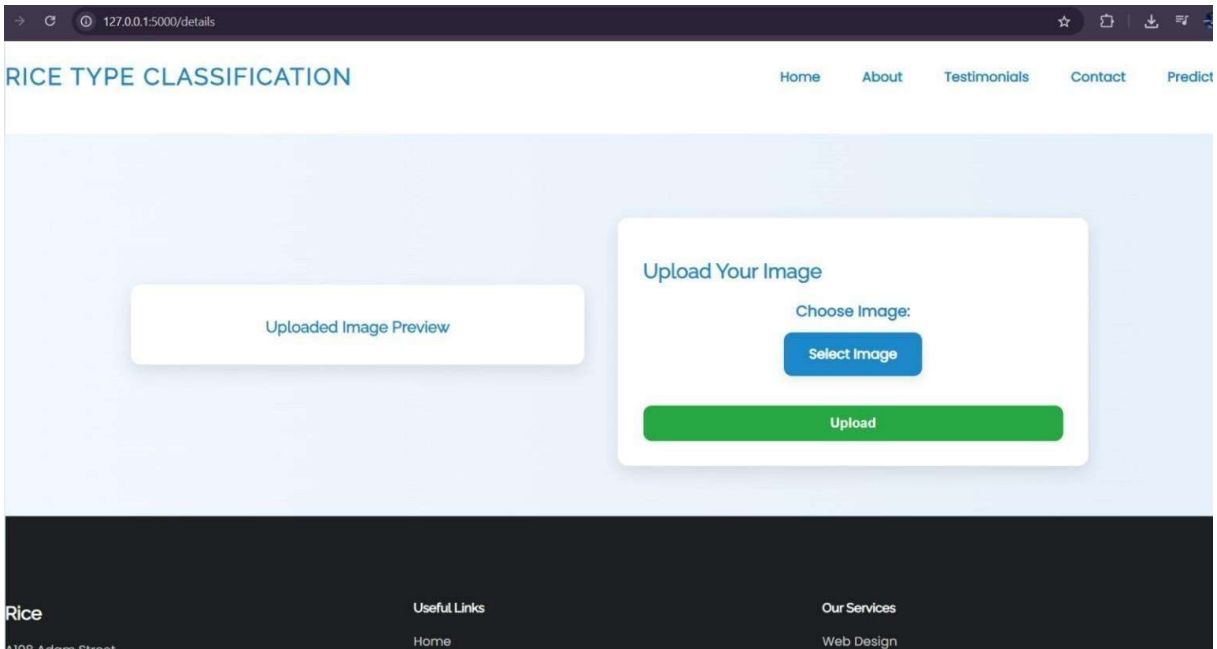


Screenshot 2: Contact Page

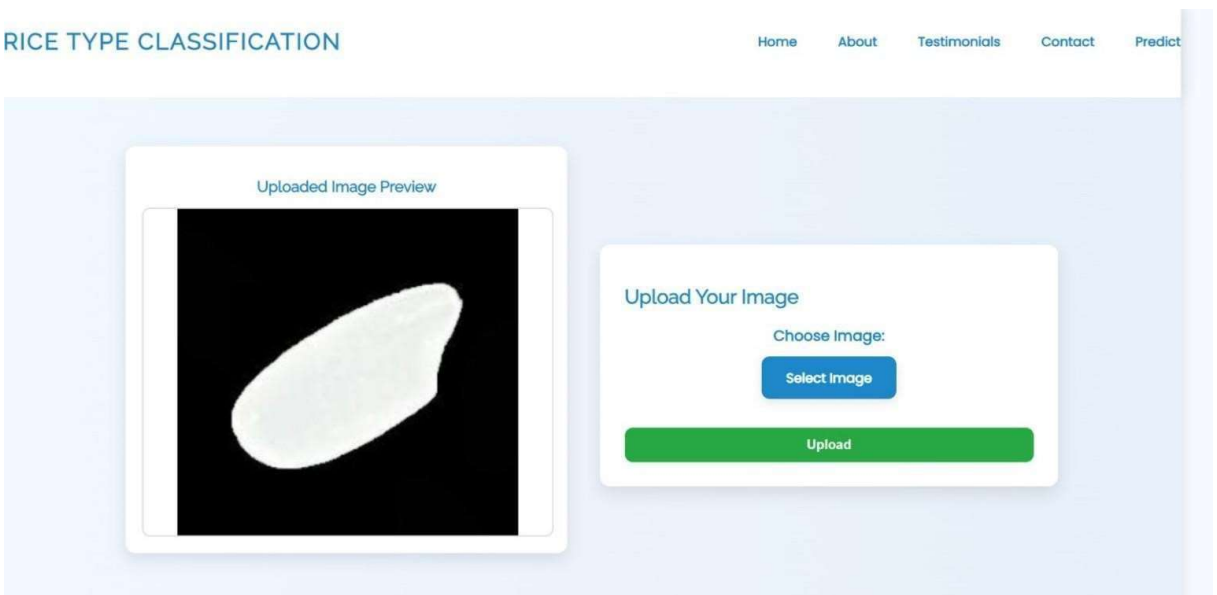


GrainPalette - AIML Project Documentation

Screenshot 3: Upload Interface

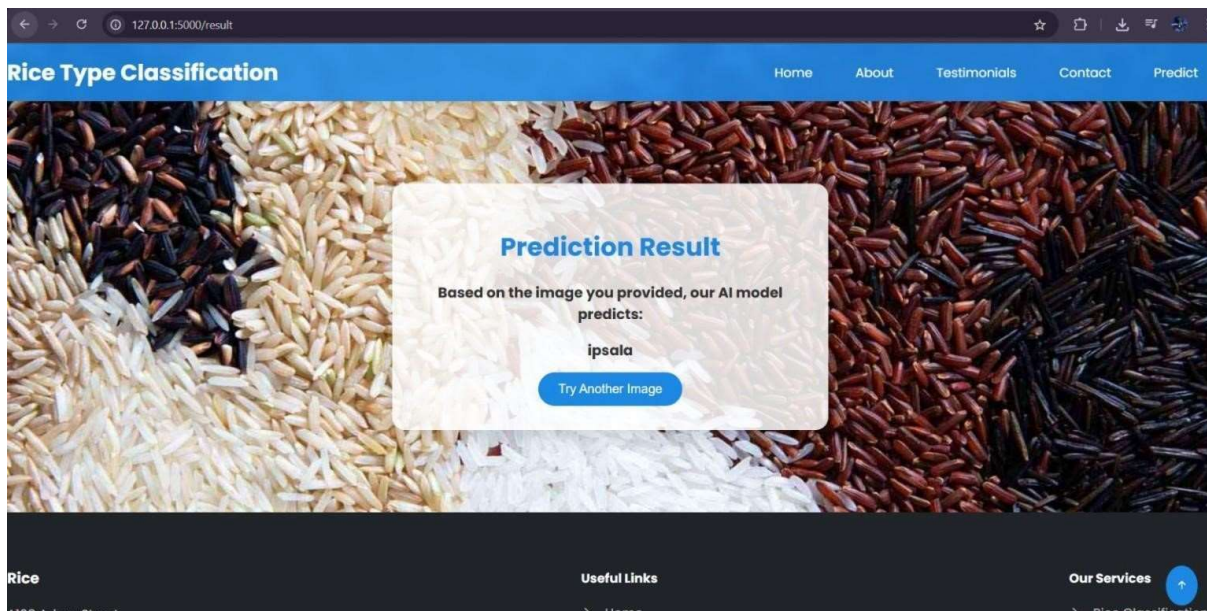


Screenshot 4: Image Preview



GrainPalette - AIML Project Documentation

Screenshot 5: Prediction Result



10. Known Issues and Future Enhancements

Known Issues:

- Sensitive to lighting, blur
- Accuracy drops with mixed grains- Limited by pre-trained model rice types

Future Enhancements:

- Larger dataset
- Multi-type detection
- Mobile app with TensorFlow Lite
- Real-time webcam integration
- Multi-language UI