Grain Palette – Project Report

Project Title: **Grain Palette –A Deep Learning Odyssey in Rice Type Classification Through Transfer Learning**

Team ID : LTVIP2025TMID45889
Team Size : 4
Team Leader : M Ravi Kiran
Team member : Kushitha Vaddi
Team member : Kunchala Hemanth Kumar
Team member : Kuthani Bhairava Satya Sanjeev1.

INTRODUCTION

1.1 Project Overview

Grain Palette is an intelligent grain classification system that uses deep learning to identify rice grain types from images. It supports image input through file upload or real-time camera capture and uses a trained Convolutional Neural Network (CNN) model to classify five major rice varieties: Arborio, Basmati, Ipsala, Jasmine, and Karacadag. The application is built for mobile, desktop, and web platforms, ensuring accessibility and scalability across diverse user bases.

1.2 Purpose

The main objectives of Grain Palette are:
- To automate rice classification for farmers, traders, and researchers.
- To provide quick, reliable predictions using AI, reducing human error.
- To offer farming tips and agricultural suggestions based on rice type.
- To enhance transparency and quality assurance in the rice supply chain.

2. IDEATION PHASE

2.1 Problem Statement

Manual identification of rice varieties is often prone to error, time-consuming, and dependent on expert knowledge. There is a need for an intelligent, efficient, and accessible solution to classify rice grains accurately and offer actionable insights.

2.2 Empathy Map Canvas

- Think & Feel: “I want to ensure the rice is of good quality.”
- Hear: “This type of rice is best, trust me.”
- See: Manual methods, unverified quality, and wrong labels.
- Say & Do: “I wish there was a fast and accurate way to identify rice.”

- Pain: Mislabeling, wastage, quality issues.
- Gain: Instant identification, confidence in product quality, recommendations.

2.3 Brainstorming

- Image classification using CNN
- Mobile and web-based accessibility
- Real-time camera input
- Agricultural tips based on rice type
- Offline model support using TFLite

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Stage	User Action	Experience	Platform
Discover	Finds app/website	Curiosity	Web/Mobile
Use	Uploads/captures image	Engaged	Web/Mobile
Result	Receives classification	Satisfied	Web/Mobile
Recommend	Shares insights	Trust built	Web/Mobile

3.2 Solution Requirements

Functional Requirements:
- Upload and process image input
- Real-time camera integration
- Predict rice variety using AI model
- Display confidence score and suggestions

Non-Functional Requirements:
- Fast response time
- Cross-platform compatibility
- User-friendly UI

3.3 Data Flow Diagram

Level 0 DFD:
User --> [Upload/Capture Image] --> [Rice Classifier] --> [Display Result + Suggestion]

Level 1 DFD:
[User] --> [Input Module] --> [Preprocessing] --> [CNN Model]
↓
[Rice Type + Confidence + Tips]

3.4 Technology Stack

- Frontend: HTML, CSS, Flutter
- Backend: Flask (Python)
- AI Model: TensorFlow, Keras, TFLite
- Image Processing: OpenCV
- Others: NumPy, Matplotlib

4. PROJECT DESIGN

4.1 Problem Solution Fit

The project fits well in addressing rice classification challenges by providing an automated, AI-powered, and platform-independent solution.

4.2 Proposed Solution

A deep learning model (CNN) trained on labeled rice images.
TFLite model for efficient inference on mobile and desktop.
Integrated web and app interface for input, output, and user interaction.

4.3 Solution Architecture

User Interface → Image Preprocessing → CNN Classifier → Rice Type + Confidence + Tips

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Week 1: Problem research & data collection
Week 2: Dataset preparation & cleaning
Week 3: Model building & training
Week 4: Model evaluation & conversion to TFLite
Week 5: UI/UX design for app and web
Week 6: Backend integration & testing
Week 7: Cross-platform deployment
Week 8: Final testing, documentation & submission

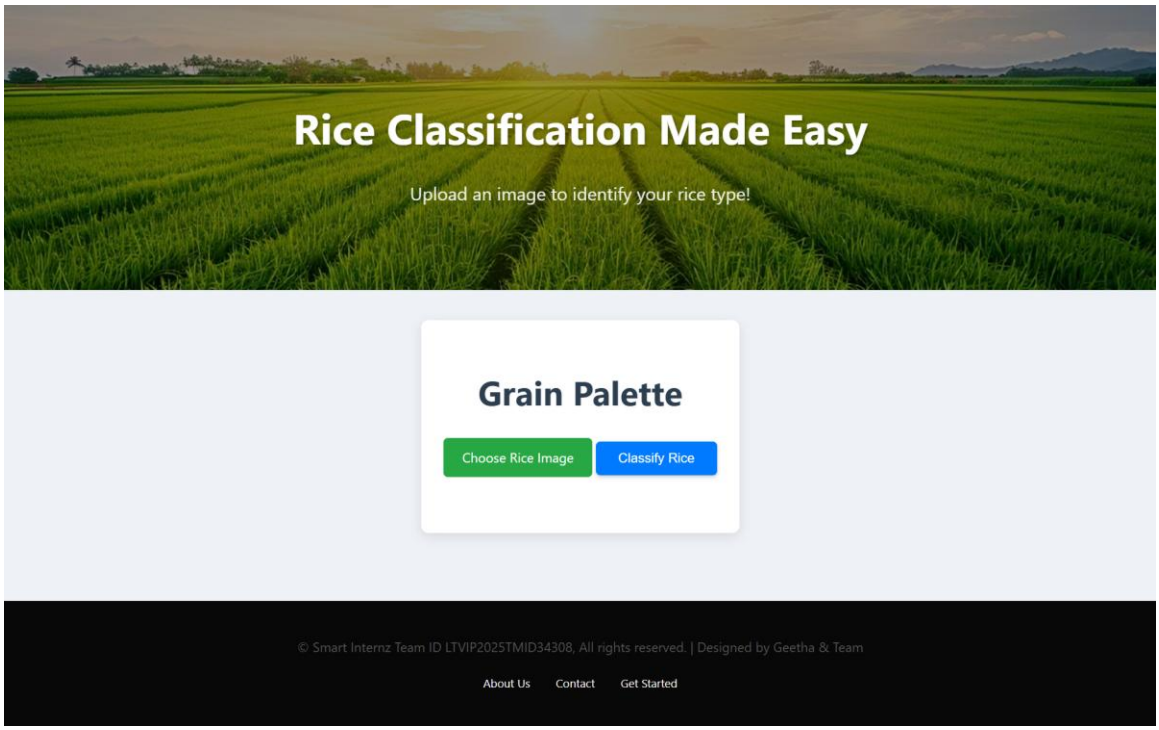
6. FUNCTIONAL AND PERFORMANCE TESTING

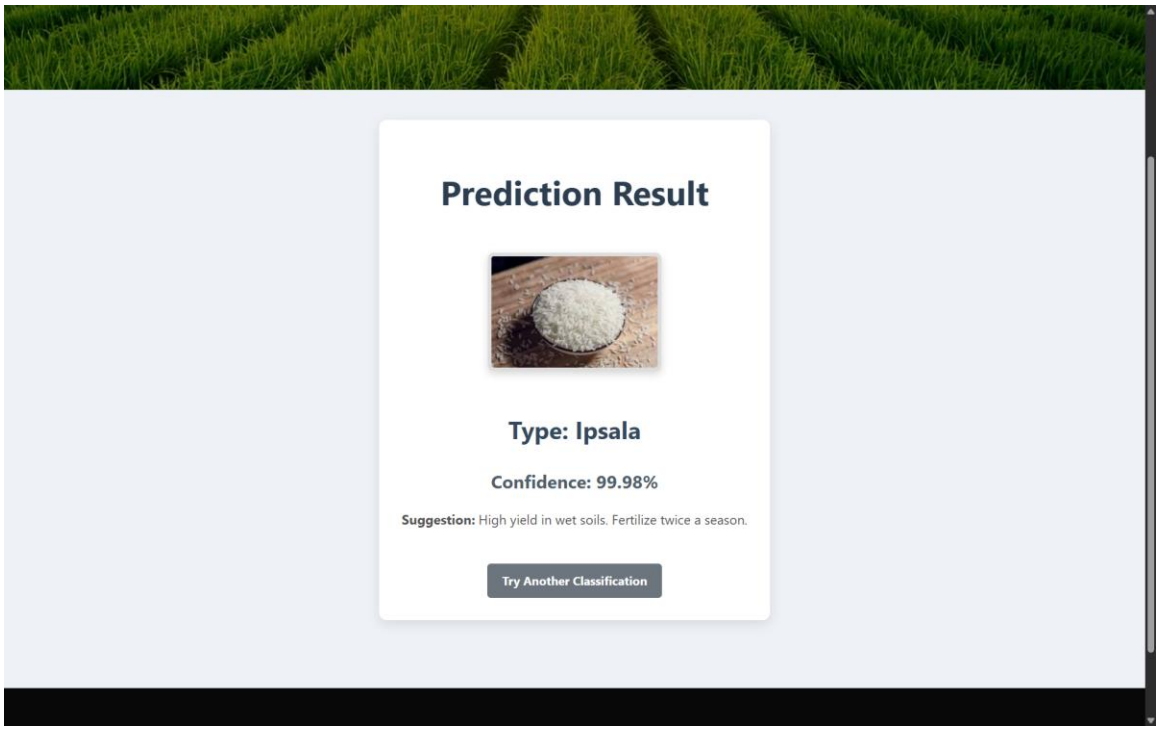
6.1 Performance Testing

- Accuracy: Achieved over 92% classification accuracy on test data
- Latency: Real-time predictions (<1 sec per image) using TFLite
- Cross-platform Testing: Successfully ran on Android, Windows, and Web

7. RESULTS

7.1 Output Screenshots





Example:
- Predicted: Basmati
- Confidence: 95.3%
- Tip: Needs moderate water, ideal for North Indian climates.

8. ADVANTAGES & DISADVANTAGES

Advantages:
- Fast and accurate rice classification
- Reduces dependency on experts
- Works offline (TFLite model)
- Real-time camera support
- Available on multiple platforms

Disadvantages:
- Limited to trained rice types
- Requires clear input images
- Model retraining needed for more types

9. CONCLUSION

Grain Palette demonstrates how AI can transform agricultural processes by providing instant grain classification. It enhances the accuracy of rice identification, supports better decision-making, and empowers users through a simple, intuitive platform. Its cross-platform nature ensures broader accessibility and usability.

10. FUTURE SCOPE

- Add more rice varieties and grains (wheat, pulses)
- Improve model accuracy using more diverse datasets
- Enable multi-language support in the app
- Introduce feedback-based learning for model updates
- Build APIs for integration with supply chains and agriculture platforms

11. APPENDIX

Python code:

https://colab.research.google.com/drive/1TTRWUdJ6eSLLrG27fI8XP2j3pCERGmtX?usp=sharing

Flask application:

https://drive.google.com/file/d/1kxfBjRjti-_CctZrqdKSQgpp8DSIdSQi/view?usp=sharing

Rice data set:

https://drive.google.com/file/d/1rxWHDElhxLF4HSZ44FFYojyv57zx9A8D/view?usp=sharing

Demo video:

https://drive.google.com/file/d/1I30Pxa65AuVoZ7yeks6HSJsaA9dLvWXi/view?usp=sharing

Git Hub:

https://github.com/kunchalahemanthkumar1432-sudo/Grain-palette.git