**📘 Final Project Report**

**GrainPalette - A Deep Learning Odyssey in Rice Type Classification**

**1. INTRODUCTION**

**1.1 Project Overview**

GrainPalette is a deep learning-powered web application designed to classify rice grains into five distinct varieties—Basmati, Jasmine, Arborio, Ipsala, and Karacadag—using image recognition. This project applies transfer learning via MobileNetV2 and is deployed using a user-friendly Flask interface.

**1.2 Purpose**

The project aims to assist farmers, agricultural professionals, and researchers in identifying rice types accurately and cost-effectively through AI-based classification.

**2. IDEATION PHASE**

**2.1 Problem Statement**

Manual rice classification is time-consuming, expensive, and subjective. Our goal is to automate this process using AI, eliminating the need for domain experts.

**2.2 Empathy Map**

Understanding farmer needs:

* **Think & Feel:** Need for affordable, accessible tools
* **Hear:** High costs for expert classification
* **Say & Do:** Prefer simple, mobile-accessible solutions
* **Pain:** No access to machine learning tools
* **Gain:** Easy rice type identification

**2.3 Brainstorming**

* Chose MobileNetV2 for efficiency
* Selected Flask for fast backend deployment
* Used custom dataset of rice images
* Planned a clean UI with upload and prediction feedback

**3. REQUIREMENT ANALYSIS**

**3.1 Functional Requirements**

* Upload rice grain image
* Classify image into 5 categories
* Display prediction result and image preview

**3.2 Non-Functional Requirements**

* Lightweight model for fast inference
* Responsive UI for better usability

**3.3 Technology Stack**

| **Component** | **Technology** |
| --- | --- |
| Frontend | HTML, CSS |
| Backend | Flask (Python) |
| Model | MobileNetV2 (TensorFlow/Keras) |
| Tools | Jupyter, Anaconda |
| Hosting | Localhost |

**4. SYSTEM DESIGN**

**4.1 Architecture Diagram**

1. User uploads image via HTML form
2. Flask handles the POST request
3. Model loads and classifies image
4. Output displayed on the same page

**4.2 Data Flow Diagram**

[User] -> [HTML Form] -> [Flask App] -> [Model Prediction] -> [Output Result]

**5. IMPLEMENTATION**

**5.1 Dataset**

* Custom rice image dataset with 5 folders (one per class)
* Preprocessed to 224x224 for MobileNetV2 compatibility

**5.2 Model**

* Base: MobileNetV2
* Layers: GlobalAveragePooling + Dense
* Optimizer: Adam
* Accuracy Achieved: ~94%

**5.3 Frontend**

<form action="/" method="POST" enctype="multipart/form-data">

<input type="file" name="file" required>

<input type="submit" value="Upload and Classify">

</form>

**5.4 Backend (Flask)**

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**6. TESTING & EVALUATION**

**6.1 Functional Testing**

* Tested each rice class upload
* Validated predictions and output display

**6.2 Performance Testing**

* Model inference time: ~2s per image
* File upload limit: ~5MB per image

**7. OUTPUT SCREENS**

* Image Upload Form
* Prediction Result Display
* Screenshot stored as static/demo\_screenshot.png

**8. ADVANTAGES & LIMITATIONS**

**✅ Advantages:**

* Fast and lightweight
* Easy to use via browser
* Accurate classification

**❌ Limitations:**

* Works only on static images
* Dependent on dataset quality
* No mobile app yet

**9. FUTURE ENHANCEMENTS**

* Convert to a mobile app (Flutter or React Native)
* Add more rice types and samples
* Deploy on cloud using Heroku or AWS
* Include bulk image classification support

**10. TEAM DETAILS**

| **Team ID** | **Members** |
| --- | --- |
| LTVIP2025TMID41040 | Garadappagari .hemanthkumar   K Joshitha   K Sirisha   Kovi Mohan |
|  |  |

**11. APPENDIX**

* 📂 GitHub Repository: [GrainPalette GitHub Repo](https://github.com/mohankovi/Grain-Palette-A-Deep-Learning-Odyssey-In-Rice-Type--Classification--Through-Transfer-Learning)
* 📸 Demo Screenshot: static/demo\_screenshot.png
* 🖥 Run App:

python app.py

Then visit: http://127.0.0.1:5000