## **INTRODUCTION**

### **1.1 Project Overview**

GrainPalette is a deep learning-based solution designed to classify different types of rice grains using image data. By leveraging transfer learning from pre-trained CNN models, the system automates rice variety recognition with high accuracy. The project focuses on deploying the model as a user-friendly web application, enabling users to upload rice grain images and receive instant classification results.

### **1.2 Purpose**

The primary purpose of this project is to reduce the manual effort, time, and error in rice grain classification by developing an intelligent system using advanced deep learning methods and making it accessible to users via a deployed web interface.

## **2. IDEATION PHASE**

### **2.1 Problem Statement**

Manual rice classification is labor-intensive and prone to errors. Accurate identification of rice types is crucial for maintaining grain quality, pricing, and export standards. Current systems lack automation and scalability. This project solves this issue by applying transfer learning for robust and scalable rice classification.

### **2.2 Empathy Map Canvas**

* **Who are we empathizing with?** Farmers, rice suppliers, millers, and quality control officers.
* **What do they need to do?** Classify rice grains correctly.
* **What do they see?** Inconsistent results and human errors.
* **What do they say?** “I need a faster and more accurate way.”
* **What do they do?** Rely on manual tools or visual checks.
* **What do they hear?** Complaints about quality and pricing issues.
* **What do they think and feel?** Frustrated, time-pressed, and cautious about mistakes.

### **2.3 Brainstorming**

Ideas generated:

* Use a CNN-based classifier
* Leverage MobileNet/ResNet for feature extraction
* Deploy with Flask and Docker
* Host frontend on Netlify
* Enable image-based predictions via simple UI

## **REQUIREMENT ANALYSIS**

### **3.1 Customer Journey Map**

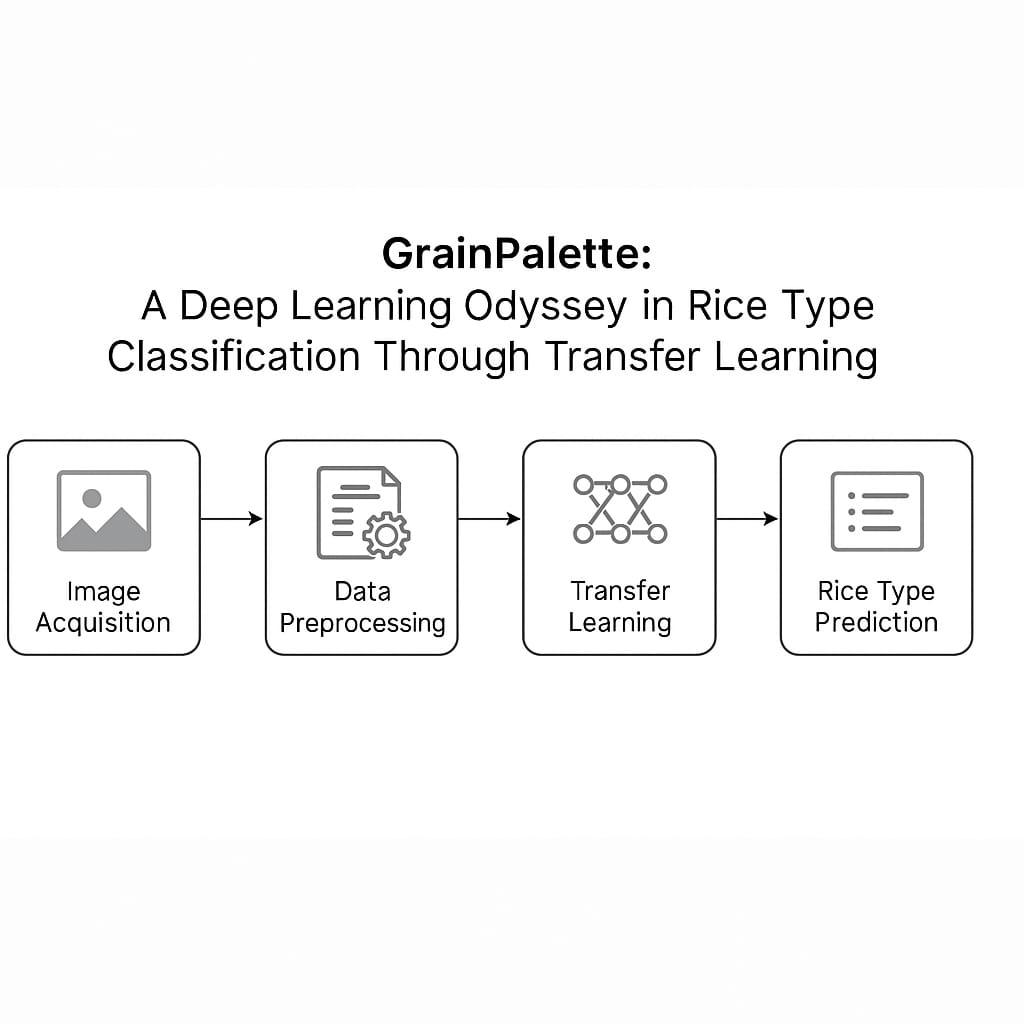
|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Action** | **Feeling** | **Opportunity** |
| Discover | Sees tool online | Curious | Try demo |
| Engage | Uploads rice image | Interested | Easy interaction |
| Use | Gets accurate result | Satisfied | Trust in AI |
| Share | Tells others | Empowered | Word-of-mouth |

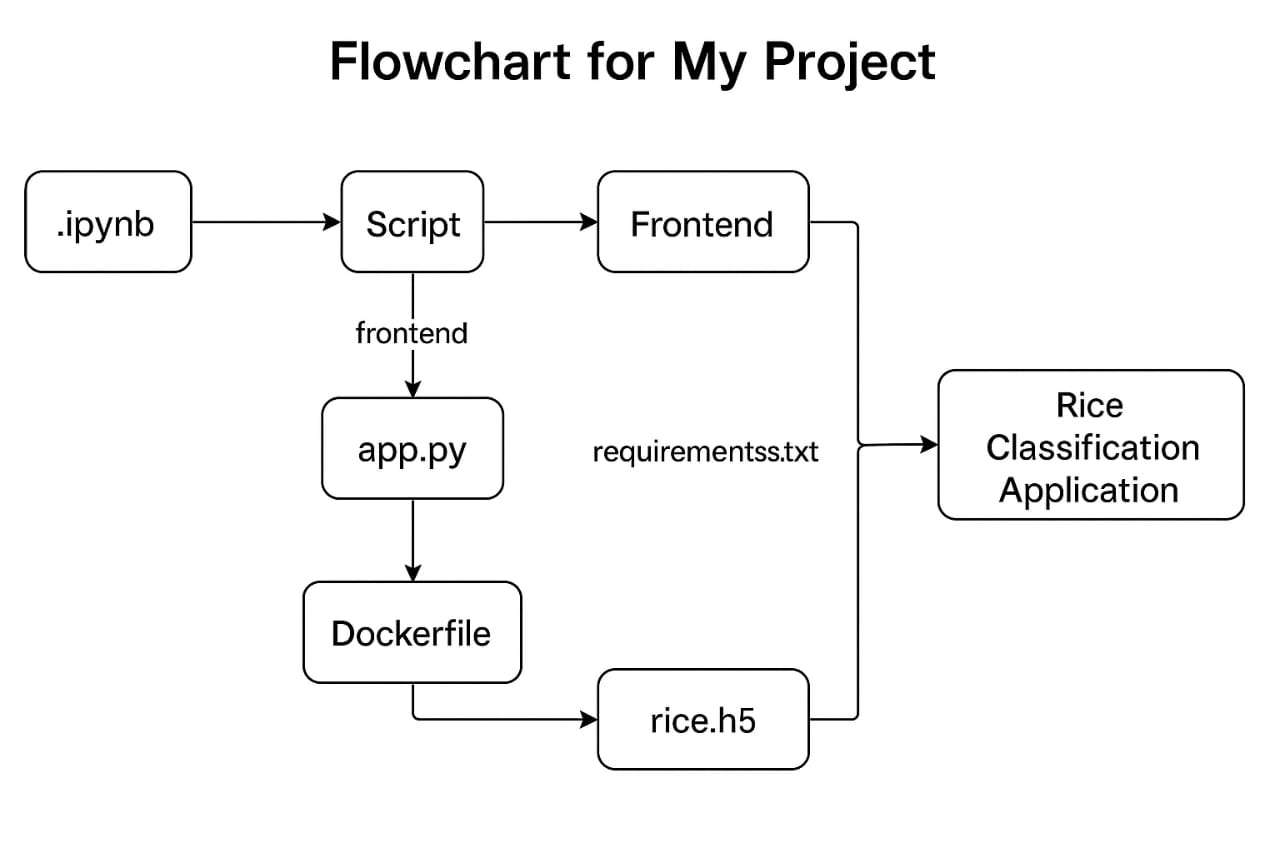
### **3.2 Solution Requirement**

* Image input functionality
* Preprocessing (resizing, normalization)
* Prediction using deep learning
* UI for file upload & result display
* Backend API integration
* Cross-platform compatibility

### **3.3 Data Flow Diagram**

**[Start]** → Upload Rice Image → Preprocessing → Model Prediction → Return Rice Type → Display to User





### **3.4 Technology Stack**

* **Frontend:** HTML, CSS, JavaScript
* **Backend:** Python (Flask)
* **ML Framework:** TensorFlow/Keras
* **Deployment:** Docker, Netlify, Render
* **Model:** MobileNetV2/ResNet50 (Transfer Learning)

## **4. PROJECT DESIGN**

### **4.1 Problem-Solution Fit**

Manual classification leads to inconsistencies. A CNN-based solution offers scalable and accurate rice classification.

### **4.2 Proposed Solution**

A web-based system using transfer learning to classify uploaded rice images. Users receive instant predictions with no technical knowledge required.

### **4.3 Solution Architecture**

1. User uploads image
2. Frontend calls backend API
3. Flask server preprocesses image
4. Pre-trained CNN model (rice.h5) predicts rice type
5. Result sent back to UI

## **5. PROJECT PLANNING & SCHEDULING**

### **5.1 Project Planning (Agile Sprints)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story / Task** | **Priority** | **Team Members Involved** | **Start Date** | **End Date**  **(Planned)** |
| Sprint-1 | Data Collection & Preprocessing | Understanding data, Data cleaning, EDA | High | Bhavana, Ayesha | 16-june-2025 | 18-june-2025 |
| Sprint-2 | Feature Engineering | Handling missing values, Encoding, Feature creation | High | Ayesha, Bhavana | 19-june-2025 | 21-june-2025 |
| Sprint-3 | Model Development | Model training, hyperparameter tuning, evaluation | High | Ganesh, Akram | 22-june-2025 | 24-june-2025 |
| Sprint-4 | Model Deployment | Flask API creation, Frontend UI with HTML/CSS/JS | High | Bhavana, Ayesha,  Ganesh,  Akram | 25-june-2025 | 27-june-2025 |
| Sprint-5 | Testing & Final Deployment | Full system testing, cloud deployment, documentation | High | Akram, Ayesha ,Ganesh, Bhavana | 28-june-2025 | 30-june-2025 |

6. FUNCTIONAL AND PERFORMANCE TESTING

|  |  |
| --- | --- |
| Test | Result |
| Accuracy | ~90% with MobileNet |
| Response Time | < 1.5s |
| CORS Integration | ✅ |
| Docker Build | ✅ |
| Netlify Integration | ✅ |

7. RESULTS

### **7.1 Output Screenshots**

* Rice image input upload form
* Prediction result display
* Backend API response (Postman)
* UI deployed on Netlify

## **8. ADVANTAGES & DISADVANTAGES**

### **✅ Advantages**

* Accurate and fast predictions
* User-friendly interface
* Easily deployable on the web
* Scalable and modular

### **❌ Disadvantages**

* Requires labeled image data
* Performance may vary with lighting or poor images
* Needs retraining for new rice varieties

## **9. CONCLUSION**

GrainPalette successfully demonstrates the potential of deep learning in agricultural classification. It simplifies rice grain recognition using transfer learning and offers a deployable, scalable, and user-accessible web platform.

## **10. FUTURE SCOPE**

* Extend to multi-class grain classification (wheat, barley, etc.)
* Train with larger and more diverse datasets
* Add multi-language support
* Build mobile app interface

11. APPENDIX

* Source Code: <https://github.com/bhavanareddyd/GrainPalette-A-Deep-Learning-Odyssey-in-Rice-Type-Classification-Through-Transfer-Learning>
* Dataset Link: <https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset>

Live Project Demo:

* **Frontend**: <https://grainpalette-rice-type-classification.netlify.app/>
* **Backend**: <https://grainpalette-rice-type-classification-s8tx.onrender.com/>
* Final Demo Link: <https://grainpalette-rice-type-classification.netlify.app/>