### **Project Title**

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

### **System Requirements**

* Windows 8 machine
* Minimum two web browsers installed
* Internet bandwidth: 30 Mbps

**Project Description**

The **GrainPalette AI Model** assists farmers, agriculture researchers, and gardening enthusiasts by accurately classifying **five types of rice grains** using deep learning and transfer learning via **MobileNetV2**. Users upload rice grain images and receive type predictions to make informed decisions related to **crop planning, fertilization, irrigation, and education**.

### **Agile Terminology**

* **Sprint**: A fixed iteration where selected tasks are completed
* **Epic**: A large feature (Ex: Model Training)
* **Story**: A smaller task under an Epic
* **Story Point**: A number representing the task's effort, using the **Fibonacci Scale**

**Fibonacci Story Point Scale**

| **Points** | **Meaning** |
| --- | --- |
| 1 | Trivial |
| 2 | Simple (1-2 hours work) |
| 3 | Medium complexity |
| 5 | High complexity (research + implementation) |

## **Sprint 1**

### Epic 1: Dataset Preparation

| **Story ID** | **Task** | **Story Point** |
| --- | --- | --- |
| USN1 | Organize rice dataset folders | 1 |
| USN2 | Apply ImageDataGenerator with augmentation | 2 |
| USN3 | Split data into train and validation sets | 2 |
| USN4 | Visualize sample batches for verification | 1 |

### Epic 2: Model Architecture

| **Story ID** | **Task** | **Story Point** |
| --- | --- | --- |
| USN5 | Load MobileNetV2 base model (freeze layers) | 2 |
| USN6 | Add dense, dropout, and classification head | 2 |
| USN7 | Compile model with optimizer and loss | 1 |

**Total Points: 11**

## **Sprint 2**

### Epic 3: Model Training & Evaluation

| **Story ID** | **Task** | **Story Point** |
| --- | --- | --- |
| USN8 | Train model with training and validation sets | 3 |
| USN9 | Evaluate training and validation accuracy | 1 |
| USN10 | Plot accuracy and loss graphs | 2 |
| USN11 | Generate confusion matrix | 2 |
| USN12 | Generate classification report | 2 |
| USN13 | Calculate accuracy score | 1 |

### Epic 4: Deployment

| **Story ID** | **Task** | **Story Point** |
| --- | --- | --- |
| USN14 | Save model as .h5 file | 1 |
| USN15 | Create technical summary/documentation | 2 |

**Total Points: 14**

## **Velocity Calculation**

| **Sprint** | **Story Points** |
| --- | --- |
| Sprint 1 | 11 |
| Sprint 2 | 14 |

**Total Points Completed = 25**  
 **Number of Sprints = 2**  
 **Velocity = 25 / 2 = 12.5 story points/sprint**

## **Final Deliverables**

* Trained model saved as: rice\_type\_model.h5
* Final training accuracy: ~{train\_acc:.2%}
* Final validation accuracy: ~{val\_acc:.2%}
* Model evaluation: confusion matrix, classification report
* Accuracy & loss plots for reporting