Planning Logic

Date	31 January 2025
Team ID	LTVIP2025TMID60812
Project Name	Grain Palette - A Deep Learning Odyssey In Rice Type Classification Through Transfer Learning
Maximum Marks	4 Marks

Planning Logic

Planning logic refers to the structured, step-by-step approach used to guide the development, training, evaluation, and deployment of the rice classification system. It ensures that the project flows logically from problem identification to a working solution.

□□ 1. Problem Understanding

- Define the objective: Automate rice grain variety classification using images.
- Identify stakeholders: Farmers, rice millers, traders, quality inspectors.
- Clarify constraints: Limited dataset, need for lightweight models, real-world deployment.

□□ 2. Data Strategy

- **Data Collection:**
  - Acquire images for multiple rice varieties under controlled and variable conditions.
  - Ensure class balance and diversity in the dataset.
- **Data Annotation:**
  - Label each image with the correct rice type.
- **Data Preprocessing Logic:**
  - Resize all images to 224x224 pixels (model input standard).
  - Normalize pixel values (0–1 range or using ImageNet mean/std).
  - Augment dataset (rotation, zoom, brightness) to improve generalization.

□ 3. Model Selection & Training Logic

- **Model Choice:**
  - Select models based on accuracy, size, and speed (e.g., VGG16, ResNet50, EfficientNetB0).
- **Transfer Learning Plan:**
  - Load pretrained weights (ImageNet).
  - Freeze base layers (initially) → fine-tune later layers → unfreeze for full training.
- **Classifier Head Design:**
  - Add GlobalAveragePooling → Dense → Dropout → Softmax layers for classification.
- **Training Loop Logic:**
  - Use early stopping, learning rate scheduler.
  - Evaluate on validation set after each epoch.
  - Save the best-performing model.

□ 4. Evaluation Logic

- Split data: 70% training, 15% validation, 15% test.
- Use metrics: Accuracy, Precision, Recall, F1-Score, Confusion Matrix.
- Perform:
  - Cross-validation to test robustness.
  - Error analysis on misclassified images.

□□ 5. Deployment Logic

- Export the trained model to ONNX or TensorFlow Lite if needed.
- Build a simple UI (desktop/web/mobile) for:
  - Image upload
  - Real-time prediction
  - Confidence score display
- Optimize inference time (Quantization, Pruning if needed).

□ 6. Reporting & Visualization Logic

- Export predictions and metrics.
- Use **Power BI** or **Matplotlib/Seaborn** to visualize:
  - Model comparisons
  - Class-wise accuracy
  - Data distribution and performance trends

✅7. Final Integration Logic

- Package the solution (model + UI + dataset link).
- Document dependencies, setup instructions, and test cases.
- Provide guidelines for retraining or updating the model with new rice varieties.