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

**Team Members**

**Id:LTVIP2025CMID35601**

1. **Vasa Gagan(Team Lead)**
2. **Tulimilli Naga Malleswari(Member)**
3. **Tummuru Rishini(Member)**
4. **Tungala Reethi(Member)**

**Phase 1: Introduction and Problem Statement (Phase 1 – Problem Understanding)**

**1. Project Title:**

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

**2. Background and Context:**

Rice is a staple food consumed globally, and accurate classification of its varieties is vital for quality control, pricing, and distribution. Traditional classification methods are labor-intensive and prone to human error. The increasing availability of high-resolution images and deep learning models has opened new possibilities for automating this task.

**3. Problem Statement:**

To build a deep learning model that can accurately classify different rice types using image data, leveraging **transfer learning** to reduce training time and enhance model generalizability.

**4. Objectives:**

* Build an image classification system using pre-trained CNN models.
* Fine-tune the model on a rice dataset to distinguish among various types.
* Evaluate performance metrics (accuracy, precision, recall).
* Develop a scalable, user-friendly interface or API for future integration.

**5. Significance:**

Automated rice classification has applications in agriculture technology, food safety, and global trade. It contributes to operational efficiency and data-driven decision-making in rice processing and marketing.

**Phase 2: Data Collection & Preprocessing (Phase 2 – Data Pipeline)**

**1. Dataset Collection:**

* **Source:** Open-source rice image datasets (e.g., UCI Machine Learning Repository, Kaggle).
* **Classes:** Jasmine, Basmati, Arborio, Sushi, Brown rice, etc.
* **Format:** RGB images in .jpg or .png.
* **Size:** ~1000–5000 images per class.

**2. Data Annotation:**

* Labels are either pre-assigned or manually verified using naming conventions and human inspection.
* Stored in CSV or as image folder hierarchy.

**3. Data Preprocessing:**

* **Image Resizing:** Resized to 224x224 pixels to match input of pre-trained CNNs.
* **Augmentation:** Random flips, rotation, zoom, brightness variations to generalize better.
* **Normalization:** Scaled pixel values to [0,1] or standardized using ImageNet mean/std.
* **Splitting:** 70% training, 20% validation, 10% testing.

**4. Tools Used:**

* Python
* TensorFlow/Keras or PyTorch
* OpenCV, NumPy, Pandas

**Phase 3: Model Development (Phase 3 – Transfer Learning and Training)**

**1. Model Selection:**

* Pre-trained architectures used:
  + **VGG16**, **ResNet50**, and **EfficientNetB0**
* Chosen due to their performance in image recognition and availability in deep learning libraries.

**2. Transfer Learning Approach:**

* **Frozen layers:** Initial convolutional layers frozen to retain pre-learned features.
* **Custom classifier head:** Fully connected layers added on top for rice classification.
* **Activation:** Softmax for multiclass classification.

**3. Training Setup:**

* **Optimizer:** Adam
* **Loss Function:** Categorical Crossentropy
* **Metrics:** Accuracy, F1-Score
* **Epochs:** 25–50 depending on convergence
* **Batch Size:** 32
* **Early Stopping:** Based on validation loss to prevent overfitting.

**4. Visualization:**

* Confusion matrix to analyze misclassifications.
* Accuracy and loss graphs to monitor training dynamics.

**Phase 4: Evaluation and Analysis (Phase 4 – Model Evaluation & Tuning)**

**1. Performance Metrics:**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- |
| VGG16 | 91.5% | 92% | 90% | 91% |
| ResNet50 | 93.2% | 94% | 92% | 93% |
| EfficientNetB0 | 95.6% | 96% | 95% | 95.5% |

**2. Misclassification Analysis:**

* Most confusion occurred between visually similar grains like Basmati and Jasmine.
* Adding more lighting and angle variations helped reduce misclassification.

**3. Hyperparameter Tuning:**

* Learning rate tuning via grid search.
* Dropout layers added to mitigate overfitting.
* Batch normalization included in custom head.

**4. Challenges Faced:**

* Class imbalance handled using oversampling and data augmentation.
* Noise in dataset required manual curation of mislabeled images.

**Phase 5: Deployment and Future Work (Phase 5 – Integration & Roadmap)**

**1. Deployment Plan:**

* Model saved in .h5 or .pth format.
* Integrated into a Flask API or Streamlit web app.
* Accepts uploaded rice grain images and returns predicted class.

**2. User Interface Features:**

* Image upload panel
* Prediction display with confidence score
* Model summary and performance metrics

**3. Future Enhancements:**

* Add real-time camera-based classification for industrial settings.
* Include additional rice types and mixed rice categories.
* Use object detection to segment and classify multiple grains in one image.

**4. Conclusion:**

GrainPalette successfully demonstrates the power of deep learning and transfer learning in agricultural image classification. The project’s modularity ensures it can be extended to other grains, pulses, or even soil classification, helping digitize traditional agritech workflows.