GRAINPALETTE: A DEEP LEARNING ODYSSEY IN RICE TYPE CLASSIFICATION THROUGH TRANSFER LEARNING

1. INTRODUCTION

GrainPalette is a deep learning project focused on classifying different rice types using transfer learning techniques. With the rise in food quality and safety concerns, accurate rice classification plays a critical role in agriculture and food industries. Leveraging convolutional neural networks (CNNs) and pre-trained models, this project delivers a scalable and efficient solution for grain classification.

2. PREREQUISITES

- Python Programming
- Basics of Machine Learning
- Understanding of Convolutional Neural Networks
- Knowledge of Transfer Learning
- Familiarity with tools like TensorFlow, Keras, OpenCV

3. PRIOR KNOWLEDGE

- Supervised Learning principles
- Image Classification fundamentals
- Experience with Jupyter Notebooks or similar IDEs
- Handling datasets and preprocessing using OpenCV or PIL

4. PROJECT OBJECTIVES

- To classify rice grains into distinct categories using images
- To apply transfer learning using a pre-trained CNN model
- To create a user-friendly interface for end-users to classify rice types
- To deploy the model in a lightweight application using Streamlit or Flask

5. PROJECT FLOW

The project follows a systematic flow:

1. Data Collection
2. Image Preprocessing
3. Model Building
4. Save the Model
5. Application Building

6. PROJECT STRUCTURE

The project is organized into the following key components:

6.1 DATA COLLECTION

Data was collected from online datasets and open-access resources. Categories include multiple rice types such as Basmati, Jasmine, and Arborio. Images were gathered and meticulously labeled according to their respective classes.

6.2 IMAGE PREPROCESSING

To ensure consistency and improve model performance, images were preprocessed. This involved resizing all images to a uniform dimension (e.g., 224x224 pixels). Data augmentation techniques, including rotation, zooming, and flipping, were applied to artificially expand the dataset. Pixel values were normalized to scale them between 0 and 1.

6.3 MODEL BUILDING

Transfer learning was employed, utilizing pre-trained CNN models like VGG16, ResNet50, or MobileNetV2. The top layers of these models were removed, and custom dense layers were added for the specific classification task. The model was compiled using the categorical crossentropy loss function and the Adam optimizer. Training was performed on a split of the dataset into training and validation sets.

6.4 SAVE THE MODEL

The trained model was saved in TensorFlow/Keras's standard .h5 format. This saved model is portable, allowing it to be loaded and utilized in the deployment environment without the need for retraining.

6.5 APPLICATION BUILDING

A user-friendly interface was developed using Streamlit, enabling users to upload rice images for classification. The backend of the application loads the saved model and predicts the rice type, displaying the results along with their probabilities directly on the screen.

7. TECHNOLOGIES USED

- **Python:** Core programming language
- **TensorFlow/Keras:** Deep learning framework
- **OpenCV:** Image preprocessing
- **Streamlit:** Web application deployment
- **Jupyter Notebook:** Model development and testing

8. CHALLENGES FACED

- Class imbalance in the dataset
- Selecting the most appropriate pre-trained model for the task
- Mitigating overfitting during the training process
- Addressing latency in real-time prediction

9. RESULTS

- Achieved over 90% accuracy on the validation data.
- Successfully deployed the model with an intuitive user interface.
- Optimized the pipeline for enhanced performance and accuracy.

10. CONCLUSION

GrainPalette successfully demonstrates the power of transfer learning in agricultural applications. With a high-performing model and a user-friendly deployment, this project can significantly aid in quality control processes

across rice production chains. Future work may involve expanding support to include additional grain types and integrating mobile app functionalities.

TEAM MEMBERS

- DUDDU HARSHA VARDHAN
- BORA CHAITANYA
- BOTSA JAGASATYANARAYANA
- CHETAN KATTA

REFERENCES

- TensorFlow Documentation
- Keras Tutorials
- OpenCV Python Guide
- Kaggle Dataset Resources
- Streamlit Official Docs