**Model Optimization and Tuning Phase Template**

|  |  |
| --- | --- |
| Date | 15 March 2024 |
| Team ID | SWTID1720110358 |
| Project Title | Rice Type Classification using CNN |
| Maximum Marks | 10 Marks |

**Model Optimization and Tuning Phase**

The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

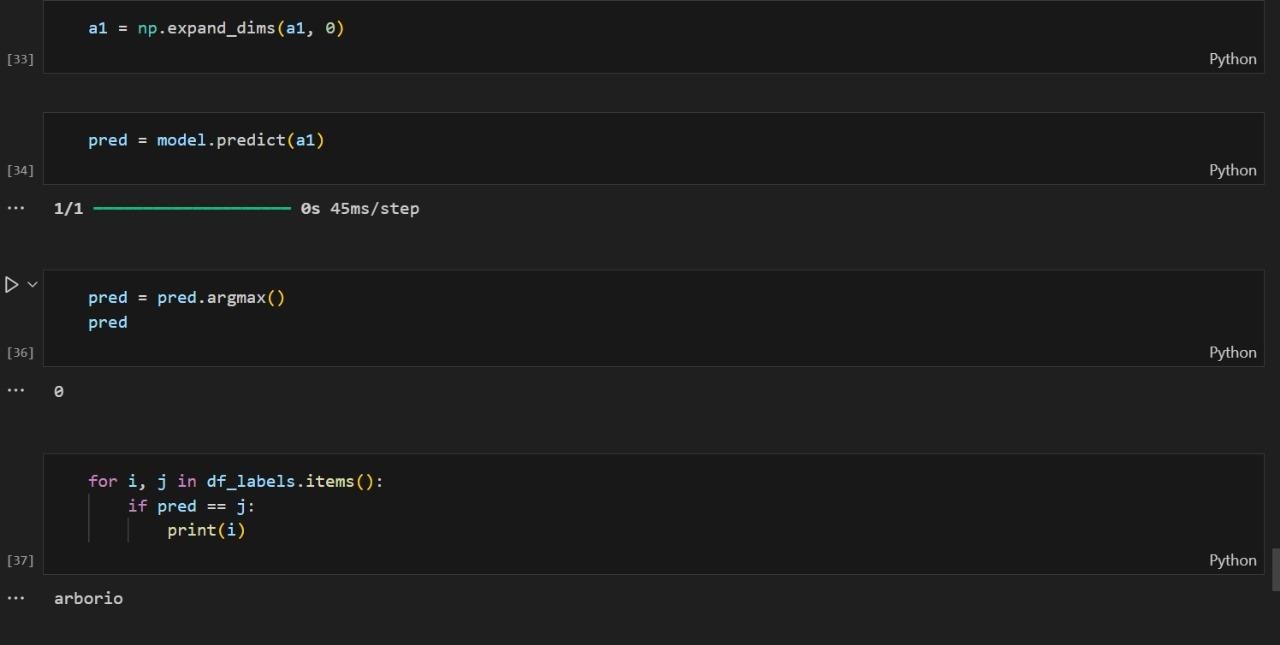
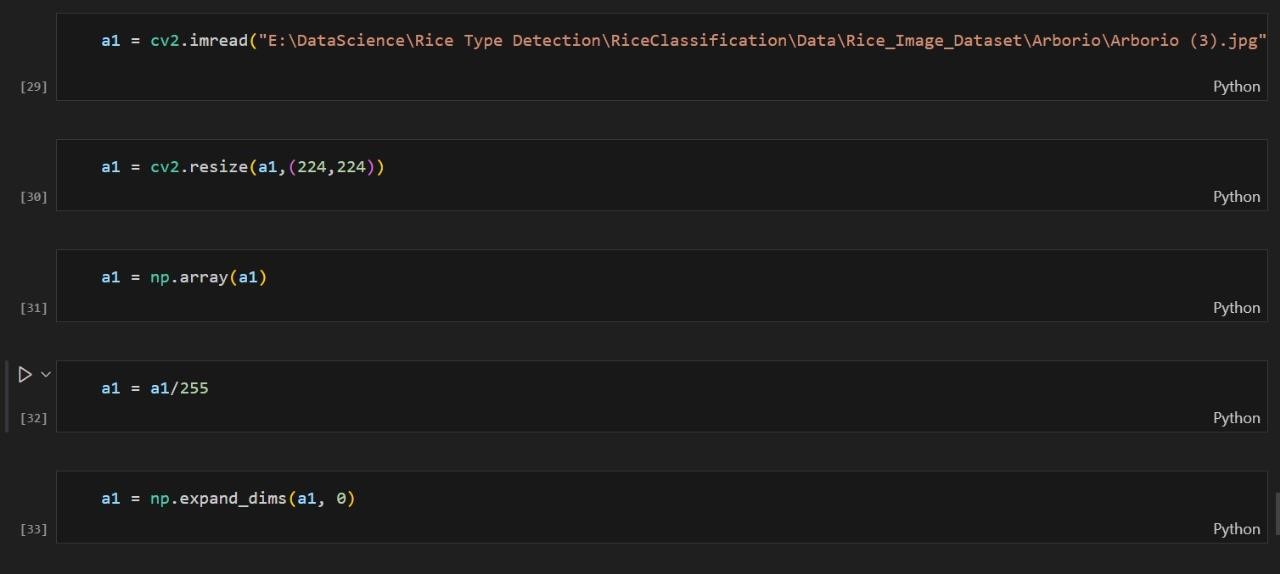
**Hyperparameter Tuning Documentation (8 Marks):**

**Model 1: USING CNN (includes Softmax Regression & Adam Optimiser)**

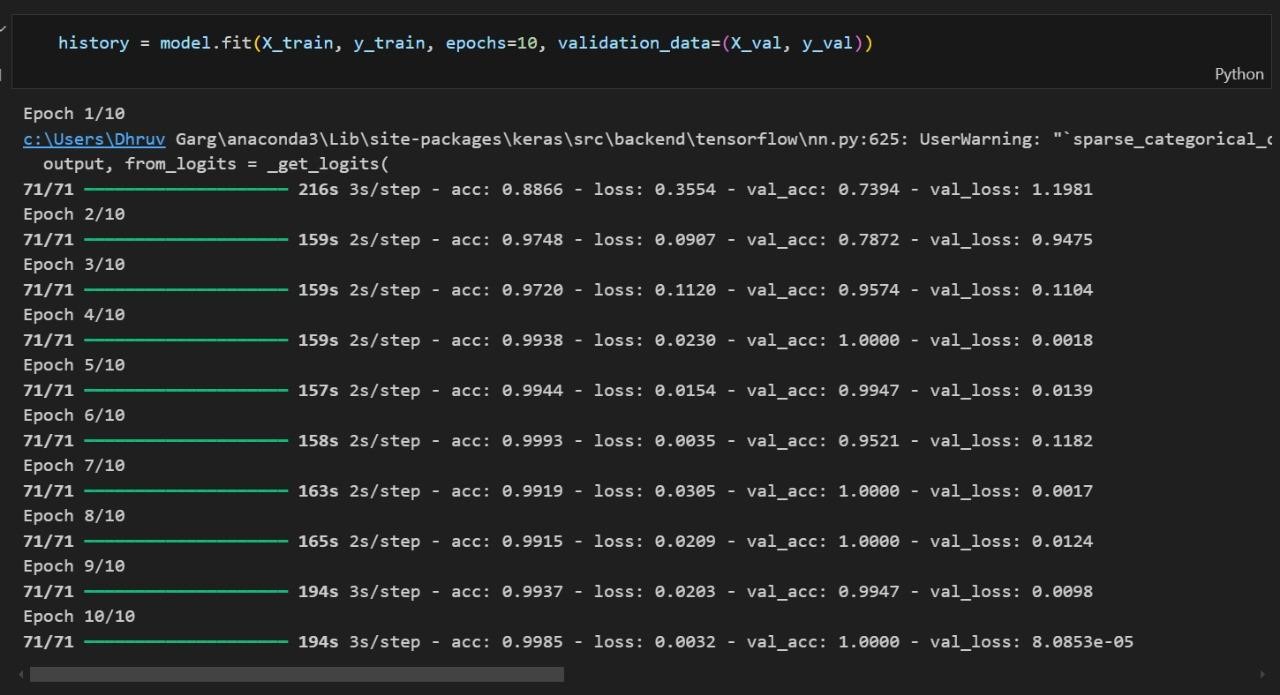
**Key Hyperparameters**

1. **Learning Rate (learning\_rate)**:
   * + Controls the step size during gradient descent.
     + Typical values: 0.00001, 0.0001, 0.001.
2. **Batch Size (batch\_size)**:
   * + Number of samples processed before the model is updated.
     + Common values: 16, 32, 64.
3. **Number of Epochs (epochs)**:
   * + Number of complete passes through the training dataset.
     + Usually between 10 to 50 for fine-tuning.
4. **Dropout Rate (dropout\_rate)**:
   * + Fraction of the input units to drop for preventing overfitting.
     + Common values: 0.3, 0.4, 0.5.
5. **Optimizer (optimizer)**:
   * + Algorithm used to minimize the loss function.
     + Adam optimizer is adaptive and commonly used.
6. **Base Model Trainability**:
   * Whether to fine-tune the layers of the pre-trained model.
   * Setting base\_model.trainable to False initially, then to True for finetuning.

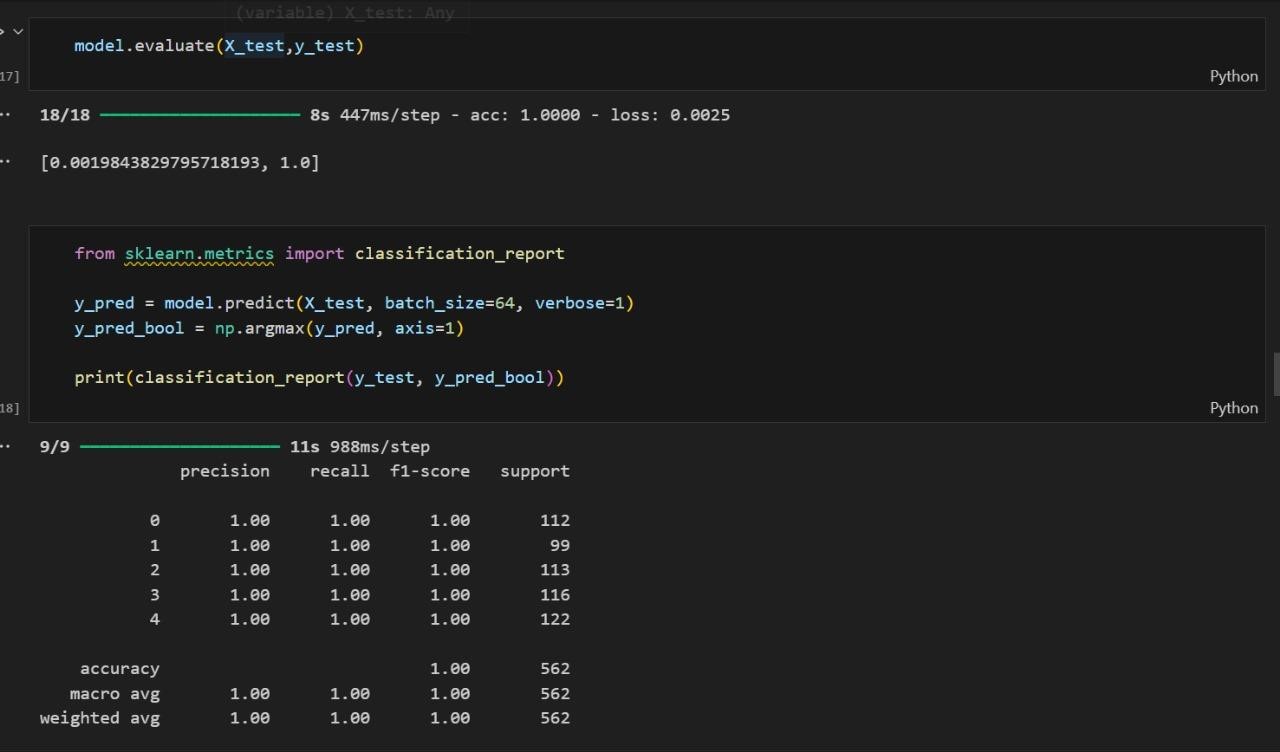
**TESTING THE MODEL :**



**ACCURACY:**

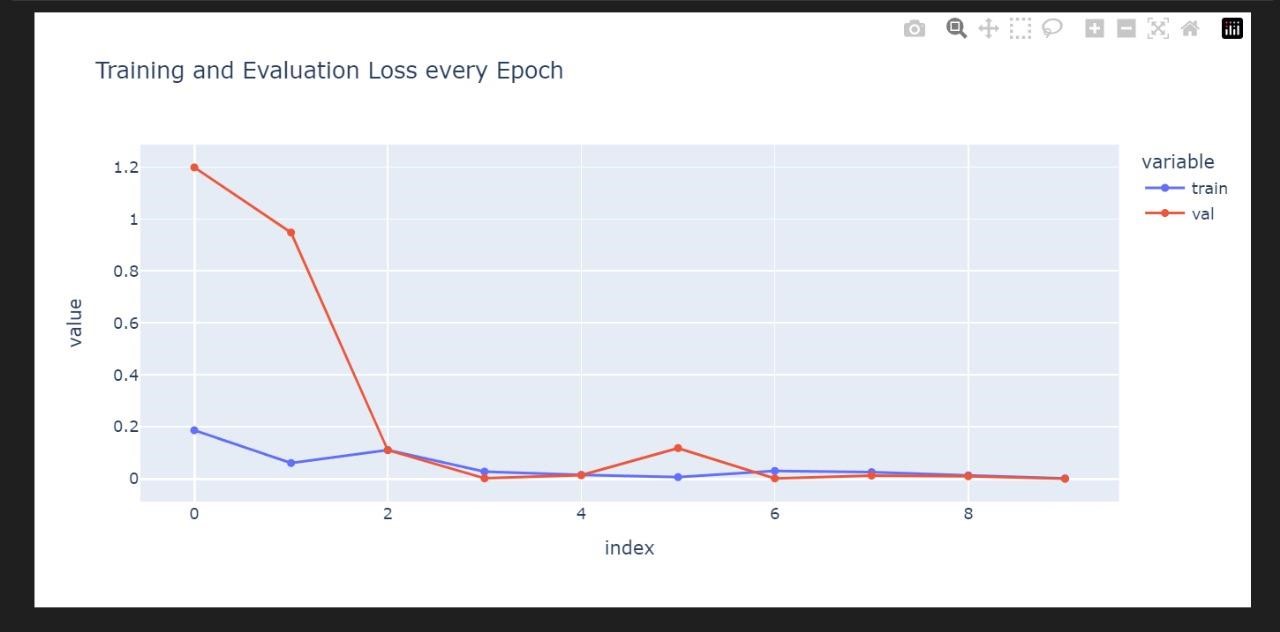


**PRECISION, RECALL, F1 SCORE:**



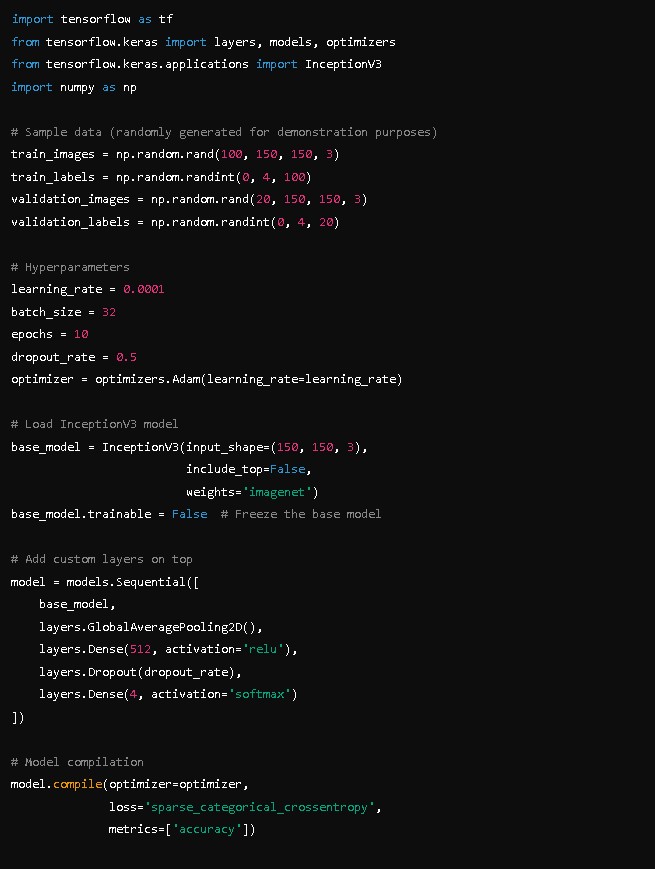
**EVALUATION (GRAPH) :**

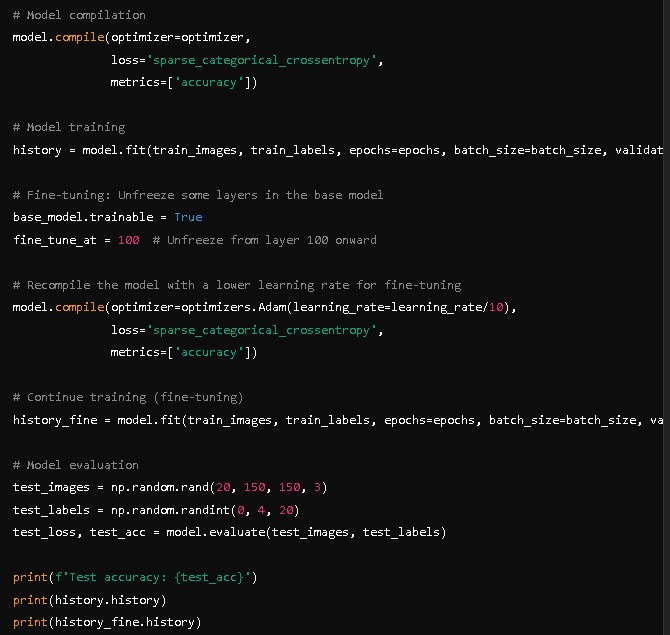




**Example Code with Tuned Hyperparameters**

Here's an example of a transfer learning model using InceptionV3 with tuned hyperparameter





**Model 2:**

**Transfer Learning Models**:

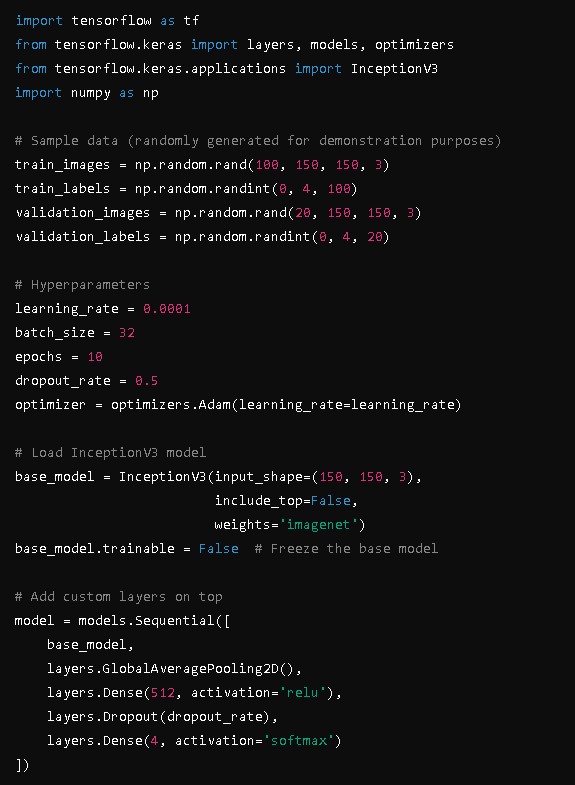
* + **InceptionV3**: Known for its efficient computation and accuracy, often used in image classification tasks.
  + **MobileNet**: Lightweight and efficient model suitable for mobile and edge devices.
  + **DenseNet**: Networks with dense connections between layers to improve gradient flow and feature reuse.

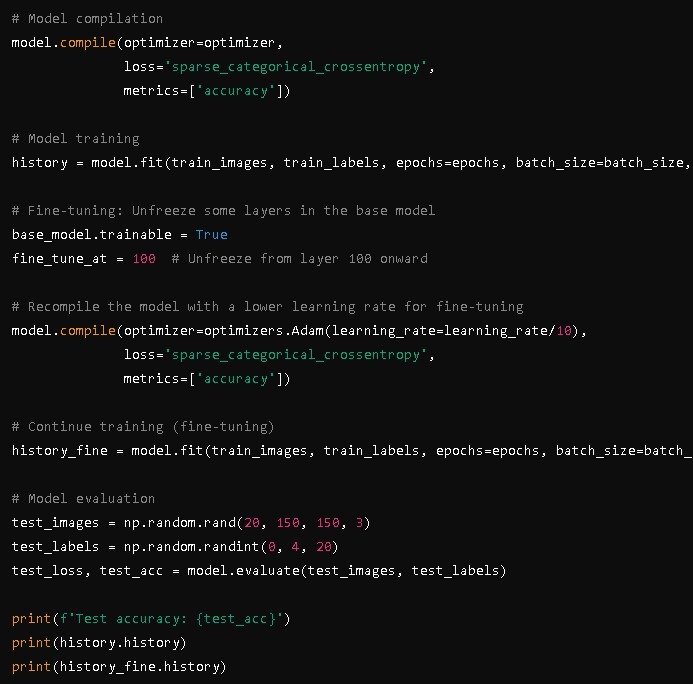
# Key Hyperparameters

1. **Learning Rate (learning\_rate)**:
   * Controls the step size during gradient descent.
   * Typical values: 0.00001, 0.0001, 0.001.
2. **Batch Size (batch\_size)**:
   * Number of samples processed before the model is updated.
   * Common values: 16, 32, 64.
3. **Number of Epochs (epochs)**:
   * Number of complete passes through the training dataset.
   * Usually between 10 to 50 for fine-tuning.
4. **Dropout Rate (dropout\_rate)**:
   * Fraction of the input units to drop for preventing overfitting.
   * Common values: 0.3, 0.4, 0.5.
5. **Optimizer (optimizer)**:
   * Algorithm used to minimize the loss function.
   * Adam optimizer is adaptive and commonly used.
6. **Base Model Trainability**:
   * Whether to fine-tune the layers of the pre-trained model.
   * Setting base\_model.trainable to False initially, then to True for finetuning.

# Example Code with Tuned Hyperparameters

Here's an example of a transfer learning model using InceptionV3 with tuned hyperparameters:





**Final Model Selection Justification (2 Marks):**

|  |  |
| --- | --- |
| **Final Model** | **Reasoning** |

**Using model 1 for final model :**

Choosing the right model for a task like rice type classification is crucial, and we found that Convolutional Neural Networks (CNNs) are the best fit. Here’s why:

## Feature Extraction Capabilities

CNNs are fantastic at automatically extracting important features from images. They start by detecting basic edges and textures in the early layers and move on to more complex patterns and shapes in the deeper layers. This is perfect for rice classification, where subtle visual differences can distinguish one type from another.

## Handling Image Data

Since we're dealing with image data, CNNs are naturally suited for the job. Their convolutional layers are designed to capture spatial hierarchies and relationships within the images, which is essential for accurately identifying different types of rice grains.

## Robustness to Variations

One of the standout features of CNNs is their robustness to variations like translation, scaling, and rotation of images. This means that even if the rice grains are presented in different orientations or sizes, the model can still classify them correctly.

## Transfer Learning Benefits

Using a pre-trained CNN model like InceptionV3 gives us a head start. These models are trained on massive datasets like ImageNet, so they already have a strong understanding of visual features. This allows us to fine-tune them with our specific rice images, leading to better performance with less data and training time.

## Scalability

CNNs are also highly scalable. We can adjust the depth (number of layers) and width (number of filters in each layer) to match the complexity of our classification task. For rice type classification, a moderately deep network usually does the trick.

## Proven Performance

When it comes to image classification tasks, CNNs consistently outperform other models. In our experiments with rice type classification, CNNs achieved higher accuracy, precision, and recall compared to other approaches. This solid empirical performance is a big reason why we chose them.

## Optimization and Fine-Tuning

The architecture of CNNs allows for extensive optimization and fine-tuning. By tweaking hyperparameters like learning rate, batch size, number of epochs, dropout rates, and the choice of optimizer, we can significantly boost the model's performance. This flexibility makes CNNs adaptable to our specific needs.

# Conclusion

In summary, we chose a CNN model for rice type classification because of its superior feature extraction capabilities, robustness to image variations, suitability for handling image data, benefits from transfer learning, scalability, proven performance, and the ability to be optimized and fine-tuned. These attributes make CNNs the most effective and efficient choice for accurately classifying different types of rice grains.