# Rice Classification

Wenyue Zheng
Department of Cyberspace Security
Nanjing University of Posts and Telecommunications
Nanjing, China
1024041141@njupt.edu.cn

Abstract—In recent years, as rice recognition and selection have gained significant attention, the challenges associated with manually identifying different rice varieties have become increasingly pronounced, particularly concerning efficiency and accuracy. Consequently, image-based deep learning technology presents an effective and automated solution for rice classification. This project utilizes a dataset of 75,000 rice images sourced from Kaggle to classify five rice varieties by fine-tuning the VGG16 model, achieving a classification accuracy exceeding 96%. To further enhance the model's credibility, a straightforward post-calibration method was implemented to address the issue of confidence levels not accurately reflecting true classification accuracy. The findings demonstrate that this project can effectively identify samples with high uncertainty and clearly differentiate between the confidence levels of correct and incorrect predictions. This not only confirms the potential of deep learning in rice classification but also highlights the critical role of uncertainty assessment in enhancing model reliability, which is significantly relevant to the advancement of applications in rice genetic research, breeding, and pest and disease management.

Index Terms—Image Classification, Temperature Scaling, Uncertainty.

## I. Introduction

In recent years, the recognition of rice varieties has emerged as a prominent area of interest in agriculture and agronomy research. Due to the diverse rice varieties, the manual identification of different types is not only time-consuming and labor-intensive but also challenging in terms of accuracy. Image-based deep learning algorithms can leverage visual features to automatically classify rice varieties, enabling more efficient identification and monitoring of rice quality.

Accurate and reliable classification of rice is crucial in agriculture, particularly in genetic research, breeding, and pest and disease management. This classification provides scientists with a foundational understanding of the genetic characteristics of various rice varieties, facilitating the selection of desirable traits. Moreover, precise classification supports the development of high-yield, disease-resistant, and adaptable rice varieties, thereby contributing to food security.

The project titled "Rice Classification" aims to utilize the publicly available Kaggle dataset, which includes 75,000 rice images, to classify five types of rice through the construction of a deep neural network model. To enhance the model's accuracy and reliability, fine-tuning will be employed to train the VGG model, followed by the application of a straightforward and effective temperature scaling method for confidence calibration. To further investigate the effectiveness of fine-tuning and post-calibration techniques, I will present visualizations of model feature maps, reliability graphs, and uncertainty distribution maps of the model's prediction results.

## II. Related Work

Despite the limited research on rice classification, methodologies relevant to this field have been published in plant classification [1], recognition [2], and crop disease detection [3]. The authors employ Convolutional Neural Networks (CNNs) to process plant images. CNNs extract multi-level features from these images through convolutional layers, gradually capturing key information such as shape, texture, and color to accurately classify the images into their respective plant species.

To further enhance the accuracy and efficiency of classification, researchers have fine-tuned pre-trained deep convolutional neural network models [4] to better align them with the requirements of plant image classification tasks.

However, most related work does not adequately address the reliability of classification results, often relying solely on the confidence levels provided by the model as the basis for judgment. This reliance may lead to situations where misclassifications are not effectively identified or corrected. In fact, model post-calibration techniques [5] can adjust the model's outputs, aligning the confidence distribution more closely with actual accuracy. This process not only aids in identifying classifications with high uncertainty but also provides more reliable decision support, thereby enhancing the overall credibility and practicality of the classification system.

## III. Problem Statement

In this project, the dataset was obtained from Kaggle [6] and contains 75,000 images, with each rice variety (Arborio, Basmati, Ipsala, Jasmine, Karacadag) consisting of 15,000 images. Figure 1 illustrates the image data for the five rice varieties in the original dataset. For dataset partitioning, this project implemented a strategy where 20% of the images were allocated for validation, 10% for testing, and the remaining 70% for training, ensuring

an even distribution of images across each category, as depicted in Figure 2. This balanced class distribution prevents model bias that could arise from class imbalance, thereby ensuring that the model performs consistently across different categories.

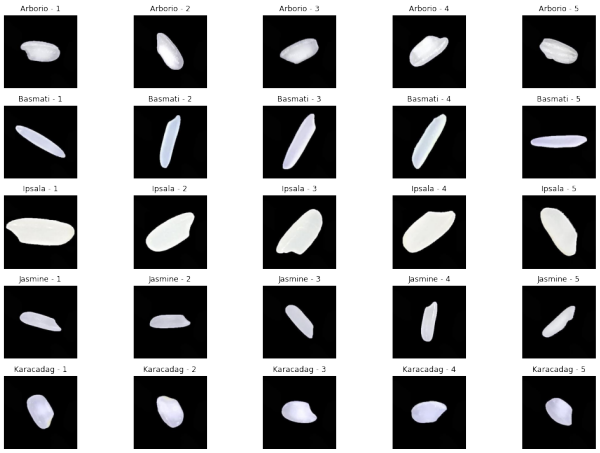


Fig. 1. Dataset Visualization

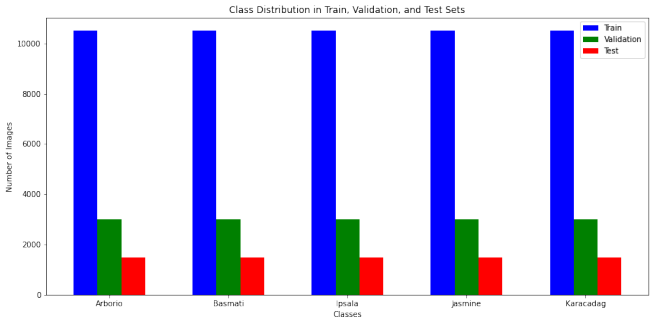


Fig. 2. Dataset Distribution

Assume that the original dataset of this project is denoted as $D$, where $D = \{(x_i, y_i)\}_{i=1}^{N}$. Here, $N$ represents the number of samples, $x_i \in \mathbb{R}^{w \times h}$ denotes the $i$-th image sample, and $y_i \in \mathbb{R}^1$ represents the label corresponding to the $i$-th image sample. This project aims to construct a mapping function $f$ that maps the input image $x_i$ to its corresponding class $\hat{y}_i$, where $\hat{y}_i \in \mathbb{R}^1$. Furthermore, it seeks to align the confidence distribution more closely with the actual accuracy:

$$P(\hat{y}_i = y_i \mid \hat{p}_{y_i i} = p) = p, \quad \forall p \in [0, 1] \tag{1}$$

where $\hat{p}_{y_i}$ represents the predicted probability component for the class $y_i$ of the $i$-th image sample.

## IV. Solutions

The foundational model of this project employs the VGG16 [7] architecture. VGG16 [7] constructs a deep convolutional network by stacking multiple convolutional layers, each equipped with 3×3 filters, followed by a max pooling layer after every group of convolutional layers(Figure 3). This architecture is composed of several modules, with the number of filters in each module increasing with the network depth. Specifically, the first few modules contain 64, 128, 256, and 512 filters, respectively, allowing the model to capture progressively more complex features.

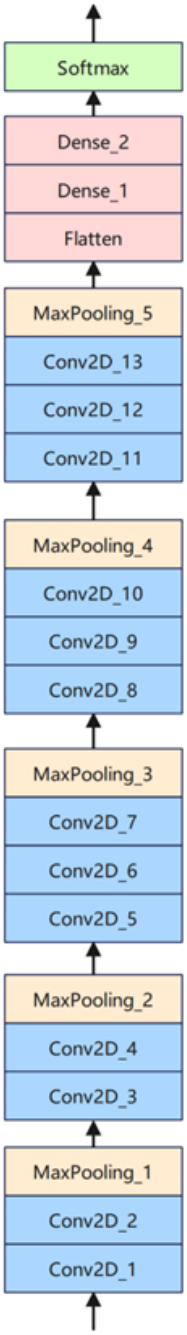


Fig. 3. VGG16 Architecture

Within this hierarchical structure, the lower convolutional layers primarily extract basic features such as edges and corners, while the middle convolutional layers

identify more complex shapes, textures, and local patterns. The higher convolutional layers capture global contextual information and high-level abstract features that are closer to the contours and structures of specific objects. To tailor the model for specific classification tasks, this project involved fine-tuning the final classification layer of VGG16, replacing the original output layer with one that outputs 10 classes, and employing the Adam optimizer during training to accelerate convergence.

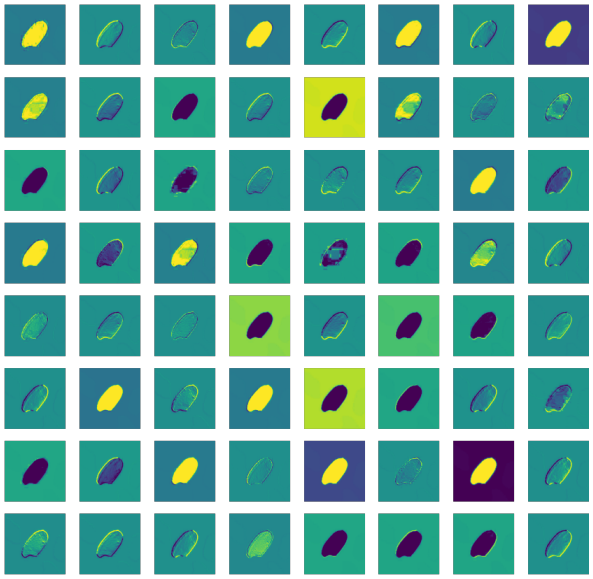

Fig. 4. Featuremap of Conv2D_1

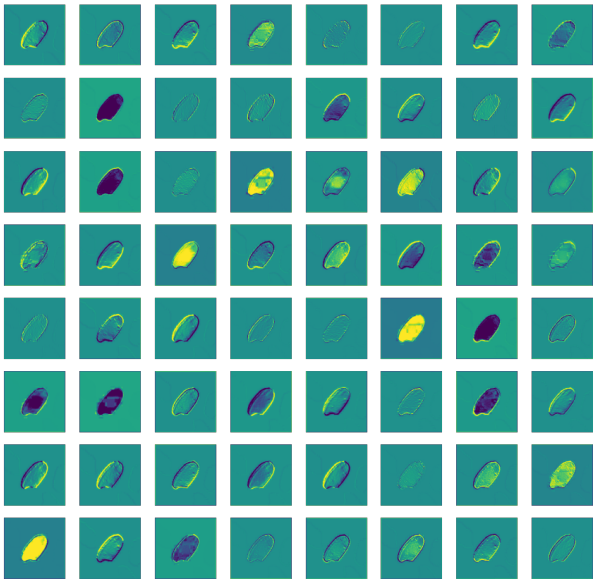

Fig. 5. Featuremap of Conv2D_2

To better illustrate the feature extraction process of the CNN model, we visualized the outputs of the first

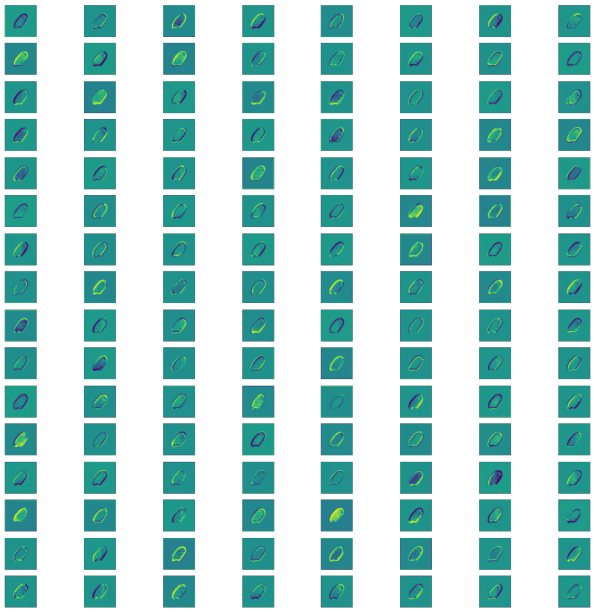

Fig. 6. Featuremap of Conv2D_3

three convolutional layers of the trained VGG16 model [8]. As depicted in Figure 4, Figure 5and Figure 6, each small square represents the feature map for a channel, with the combination of colors and shapes indicating the activation strength of each pixel. The feature maps of the first layer typically capture low-level features of the input image, such as edges and corners, representing fundamental patterns. As this is the first layer of the network, the features are relatively simple, and the variations in colors and shapes are rather uniform. As the network deepens, subsequent convolutional layers begin to capture higher-level features, including textures and patterns. Consequently, the feature maps of the second layer are more complex and diverse compared to those of the first layer. Following the first and second layers, the feature maps of the third layer begin to capture even more advanced abstract features.

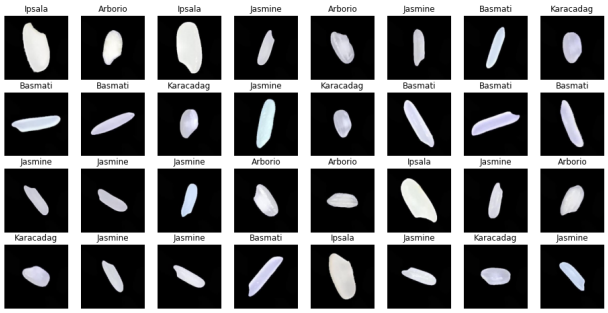


Fig. 7. Images of First Batch

Regarding image preprocessing before model training, the input images are first resized to $224 \times 224$ pixels to

ensure consistent input size for the model. Subsequently, normalization is applied to the images, which stabilizes the training process and enhances the model's overall training effectiveness. Additionally, to improve the efficiency of training, a data loader is constructed with a batch size of 32, enabling the model to process multiple images in each training iteration. Figure 7 illustrates the first batch of images from the training set. This training process is designed to leverage the hierarchical feature extraction capabilities of VGG16, allowing the model to achieve superior classification performance on specific tasks.

Since the task of this project is a multi-class classification problem, cross-entropy loss is utilized as the objective function:

$$L = -\sum_{i=1}^{N} y_i^T \log(\hat{p}_i) \tag{2}$$

where $y_i$ represents the one-hot encoded vector of the sample label, and $\hat{p}_i$ denotes the predicted probability vector for the $i$-th sample. The model optimized its parameters using stochastic gradient descent over 10 epochs, as illustrated in Figure 8.

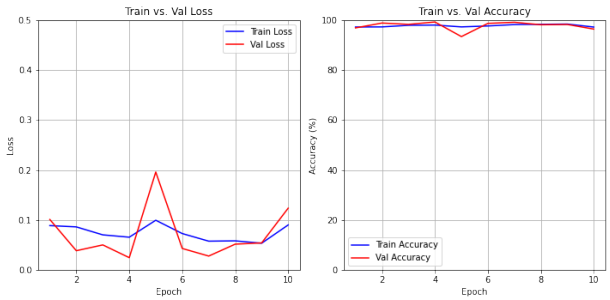


Fig. 8. Training Process

In the initial training rounds, the model quickly reached a high accuracy on the dataset, with the validation accuracy in the early stages also being notably impressive. After several epochs of training, the training accuracy approached 98%, while the validation accuracy reached approximately 99%. Although fluctuations were observed in validation loss and accuracy during certain epochs, overall, with appropriate adjustments and optimizations, the model's generalization capability was significantly enhanced. By fine-tuning the weights of specific feature layers in the VGG16 model, the model effectively captured the target features, leading to improved accuracy and robustness.

## V. Evaluation

### A. Original Model Evaluation

To evaluate the model's accuracy in classification tasks comprehensively, accuracy is used as the primary metric. However, modern deep neural networks' confidence outputs often do not accurately represent the model's true accuracy [5], making reliance on accuracy alone potentially misleading regarding model performance. Therefore, expected calibration error (ECE) is introduced as a supplementary metric for assessing model reliability. ECE quantifies the difference between the model's output confidence and the actual accuracy, reflecting the alignment between the model's predictions at various confidence levels and the true classification results.

The formula for calculating ECE is given by:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \times |\text{acc}(B_m) - \text{conf}(B_m)| \tag{3}$$

where $M$ denotes the number of confidence intervals, $B_m$ indicates the number of samples within the confidence interval, and $N$ is the total number of test samples. The introduction of ECE allows for a clearer understanding of the model's calibration across different confidence levels, thereby facilitating a more accurate assessment of the reliability of its predictions.

The left side of Figure 9 displays the distribution of predicted samples across various confidence levels, showing that the confidence in predicted labels is generally high and tends to approach the upper limit. However, as shown in Table I, the model's high confidence does not correlate with a similarly high accuracy, suggesting a tendency towards overconfidence. The right side of Figure 9 illustrates the model's reliability diagram, where the black dashed line signifies the ideal "perfect calibration line," indicating complete consistency between confidence and actual accuracy. The blue area represents the model's actual accuracy, while the red area highlights the calibration error—the gap between confidence and accuracy. Notably, the red area is prominently visible across most intervals, indicating that the model's confidence often exceeds the actual accuracy, particularly in the medium to high confidence ranges, demonstrating a significant inclination towards "overconfidence."

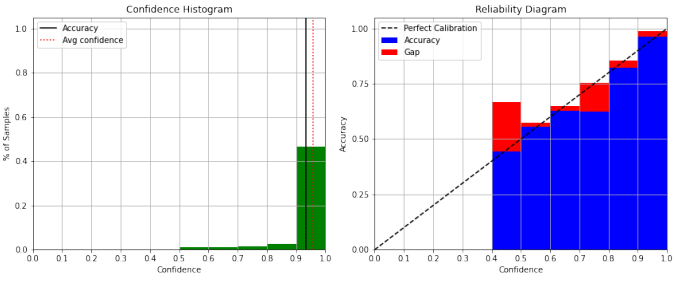


Fig. 9. Confidence Histogram and Reliability Diagram of the Original Model

### B. Model Calibration and Evaluation

To address the model's tendency toward overconfidence, we employ temperature scaling [5], an efficient post-processing calibration technique, to adjust the model's

TABLE I
Comparison of Model Performance Metrics Before and After Calibration

| Metric | Original Model | Calibrated Model |
|---|---|---|
| Accuracy | 96.44% | 96.44% |
| ECE | 1.46% | 0.86% |
| Avg ECE | 0.9589 | 0.9536 |
| Avg Accuracy | 0.9325 | 0.9410 |

output confidence. Temperature scaling adjusts the logit outputs of the model by a scaling factor, aligning predicted confidence levels more closely with actual accuracy without altering predicted labels. The calibration formula is as follows:

$$\hat{q}_i = \text{softmax}\left(\frac{z_i}{T}\right) \tag{4}$$

where $z_i$ represents the original logit vector, $T$ is the temperature parameter, and $\hat{q}_i$ denotes the calibrated probability distribution. In our experiments, we determine the optimal temperature $T$ on the validation set, with the best calibration achieved at $T = 1.205$. As shown in Figure 10, calibration significantly reduces the error across different confidence levels.

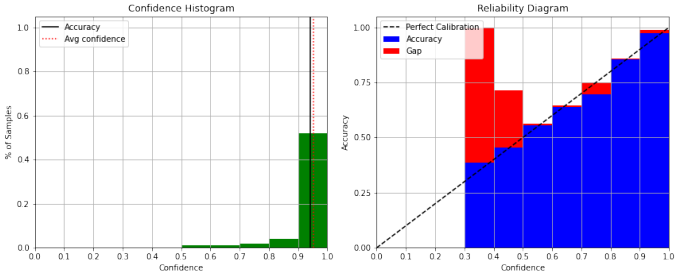


Fig. 10. Confidence Histogram and Reliability Diagram of Carlibrated Model

A comparison of reliability diagrams before and after calibration shows the following key improvements:

- Reduced Confidence-Accuracy Discrepancy: In Figure 10, the red area—indicating the gap between confidence and actual accuracy—decreases substantially post-calibration, especially in the medium to high confidence ranges. After calibration, the blue area (actual accuracy) aligns more closely with the black dashed line (the ideal calibration line), indicating that model confidence now more accurately reflects true performance.
- Lower Expected Calibration Error (ECE): As presented in Table I, the ECE decreases from 1.46% before calibration to 0.86% afterward. This reduction in ECE indicates a smaller discrepancy between predicted confidence and actual accuracy, signifying improved model calibration.
- Average Confidence Closer to True Accuracy: After calibration, the model's average confidence decreases slightly from 0.9589 to 0.9536, now closely matching the actual accuracy of 0.9410. This result shows that temperature scaling not only adjusts the numerical values of confidence but also effectively reduces overconfidence, improving the model's reliability.

## C. Further Assessment of Model Reliability

To deepen our understanding of model uncertainty, we performed additional evaluations across various confidence intervals. While the Expected Calibration Error (ECE) quantitatively measures the gap between model-predicted confidence and actual accuracy, ECE alone may not comprehensively capture model performance at differing confidence levels.

In this study, we refined our analysis by identifying an optimal uncertainty threshold on the validation set and measuring the Minimum Uncertainty of Error (MUE) on test samples. This approach aids in assessing the model's average uncertainty in mispredictions at a given threshold. The MUE is calculated as follows:

$$\text{MUE}(\delta) = 0.5 \cdot \frac{|U(D_c) > \delta|}{|D_c|} + 0.5 \cdot \frac{|U(D_i) \leq \delta|}{|D_i|} \tag{5}$$

where $\delta$ denotes the uncertainty threshold, defined as $\delta = 1 - p$ (where $p$ is the prediction confidence); $D_c$ represents correctly predicted samples, and $D_i$ represents incorrectly predicted samples.

TABLE II
Optimal Threshold and Minimum MUE

| Optimal Threshold | Minimum MUE |
|---|---|
| 0.039 | 0.174 |

TABLE III
Average Uncertainty

| Correct Predictions | Incorrect Predictions |
|---|---|
| 0.022 | 0.186 |

TABLE IV
Uncertainty Standard Deviation

| Correct Predictions | Incorrect Predictions |
|---|---|
| 0.196 | 0.164 |

As illustrated by the results in Table II, samples with an uncertainty below 0.04 are deemed to have predictions in which the model expresses adequate confidence. This low threshold indicates that the model maintains a high level of confidence across most samples, with a high likelihood of correct predictions for samples below the threshold. At this optimal threshold, the model's average misclassification rate is approximately 17.4%, reflecting well-controlled performance with limited misclassifications.

Additionally, as shown in Figure 11, we visualized the uncertainty distributions for both correct and incorrect predictions. The results demonstrate that the model

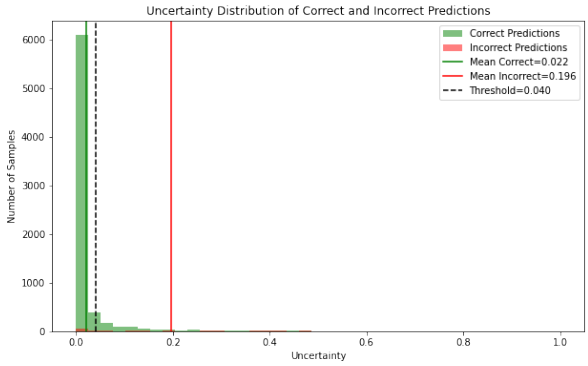


Fig. 11. Uncertainty Distribution Plot

generally displays high and stable confidence in correct predictions, while confidence diminishes significantly for incorrect predictions. This behavior aligns with expectations, as the model is more likely to err when processing uncertain samples. The marked contrast in average uncertainty between correctly and incorrectly predicted samples suggests that the model is proficient at identifying high-uncertainty cases, exhibiting increased uncertainty during errors. This trend underscores that the model's uncertainty estimation is both effective and reliable, enabling it to flag potential mispredictions in real-world applications, thereby enhancing overall dependability.

## VI. Conclusion

This project illustrates the effective application of deep learning techniques for the classification of rice varieties, highlighting the significant potential of automated image recognition in agricultural research. Utilizing a dataset of 75,000 rice images from Kaggle, we successfully developed a fine-tuned VGG16 model and calibrated its confidence levels using temperature scaling techniques. The experimental results indicate that the model achieved an accuracy exceeding 96% on both the validation and test sets, demonstrating the effectiveness of deep learning in rice classification tasks.

To enhance the model's reliability, we employed the Expected Calibration Error (ECE) as an evaluation metric and further computed the Minimum Uncertainty of Error (MUE). The results suggest that at lower uncertainty thresholds, the model exhibits high confidence for the majority of samples, effectively distinguishing between the confidence levels of correct and incorrect predictions. This not only enhances the model's capability to identify mispredictions but also provides robust support for decision-making in subsequent research.

In summary, the findings of this project present a viable method for the automated classification of rice varieties and establish a foundation for the application of deep learning in agriculture. Future work may focus on optimizing and expanding the model, addressing rice classification challenges in more complex environments, and advancing the automation and intelligence of agricultural processes. Furthermore, future visual applications could explore additional uncertainty assessment methods, such as ensemble techniques [9] [10] and Bayesian inference [11], to gain a more comprehensive understanding of the model's decision-making processes and improve its stability and reliability under diverse conditions. This approach will contribute to achieving more accurate and dependable rice classification in complex agricultural settings.

## References

[1] M. Dyrmann, H. Karstoft, and H. S. Midtiby, "Plant species classification using deep convolutional neural network," Biosystems engineering, vol. 151, pp. 72–80, 2016.

[2] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in 2015 IEEE international conference on image processing (ICIP). IEEE, 2015, pp. 452–456.

[3] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, p. 1419, 2016.

[4] A. K. Reyes, J. C. Caicedo, and J. E. Camargo, "Fine-tuning deep convolutional networks for plant recognition," CLEF (Working Notes), vol. 1391, pp. 467–475, 2015.

[5] C. Guo, G. Pleiss, Y. Sun et al., "On calibration of modern neural networks," in International Conference on Machine Learning. PMLR, 2017, pp. 1321–1330.

[6] "Rice image dataset," https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset, accessed: 2024-11-03.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[8] "Understanding your convolution network with visualizations," Understanding Your Convolution Network, accessed: 2024-11-03.

[9] T. G. Dietterich, "Ensemble methods in machine learning," in International Workshop on Multiple Classifier Systems. Springer Berlin Heidelberg, 2000, pp. 1–15.

[10] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," Advances in Neural Information Processing Systems, vol. 30, 2017.

[11] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" Advances in Neural Information Processing Systems, vol. 30, 2017.