POE

Portfolio of Evidence for CNN Rice Image Recognition

Bukiwe Ndamase
Student Number: ST10478713

https://github.com/VCONLINE/pdan8412-poe-bndamase

Course: Programming For Data Analytics 2 (PDAN 8412)
Programme: Post Graduate Diploma in Data Analytics
Institution: Varsity College Distance Learning

Lecturer: Sarina Till

Date: 21 November 2025

Table of Contents

Executive Summary

This project developed a custom Convolutional Neural Network (CNN) to classify rice grains into five distinct varieties. Using a dataset of over 1,000 images, the CNN achieved strong performance, with validation accuracy typically ranging from 85% to 91% and validation loss between 0.3 and 0.4. Preprocessing steps including resizing, normalization, and augmentation ensured balanced learning and helped reduce overfitting. Class weighting was also applied to address minor imbalances, enabling the model to treat all rice classes fairly during training.

The results show that the CNN can reliably distinguish rice varieties based on texture, shape, and contour. This system has practical applications in automated rice quality control, research, and agricultural supply chains, improving efficiency and accuracy (Khan and Pillay, 2021). Visualizations, such as sample image grids, class distribution plots, and training/validation curves, will be provided to demonstrate model performance and dataset characteristics.

Introduction

Rice classification plays a vital role in agricultural quality control, as it influences pricing, packaging, and supply chain decisions. Manual inspection is often slow and error-prone due to the subtle visual differences between rice varieties. This study addresses these challenges by using a Convolutional Neural Network (CNN), which excels at image classification by learning hierarchical spatial features from raw pixels. CNNs can detect patterns in edges, textures, and shapes that distinguish one rice variety from another (Hassan & Leo, 2021).

The main objective was to develop a robust multi-class model capable of accurately identifying rice types. By combining a structured CNN architecture with preprocessing, data augmentation, and early stopping strategies, the model was designed to generalize effectively to unseen images while minimizing overfitting (Choudhury, 2022).

Data Overview and Preparation

Data Description

The Rice Image Dataset originally included a large number of images spanning five rice varieties: Arborio, Basmati, Ipsala, Jasmine, and Karacadag (Koklu, 2020). To create a balanced and manageable dataset for CNN training, 200 images per class were randomly selected, resulting in a total of 1,000 images. These images exhibited variations in lighting, background, and orientation, reflecting real-world conditions. Each image was paired with its corresponding label and stored in a panda DataFrame, which facilitated efficient preprocessing, augmentation, and input into the CNN model (Hassan and Leo, 2021). This structured approach ensured a balanced dataset suitable for supervised learning and allowed for easy handling of image paths and labels throughout the workflow.

Target Variable

The target variable consisted of the five rice classes. Each class was label-encoded to convert categorical names into numeric values compatible with the CNN. This encoding, combined with the structured DataFrame, allowed seamless integration with the Keras/TensorFlow training pipeline (Chollet, 2021).

Development Environment

The project was implemented in Python 3.12 using Google Colab, selected for its GPU support, cloud-based storage, and interactive notebook environment (The Pandas Development Team, 2025). Colab enabled efficient training of the CNN, real-time visualization of metrics, and straightforward experimentation with preprocessing pipelines.

The technical stack included:

- NumPy for numerical operations
- pandas for dataset management
- Matplotlib and Seaborn for plotting training and validation curves
- TensorFlow/Keras for CNN model development and evaluation
- scikit-learn for label encoding and evaluation metrics

Augmentation and batching were handled through Keras ImageDataGenerator, which automatically applied transformations and loaded images in batches to optimize memory usage. ModelCheckpoint and EarlyStopping callbacks ensured reproducibility and selection of the best-performing model.

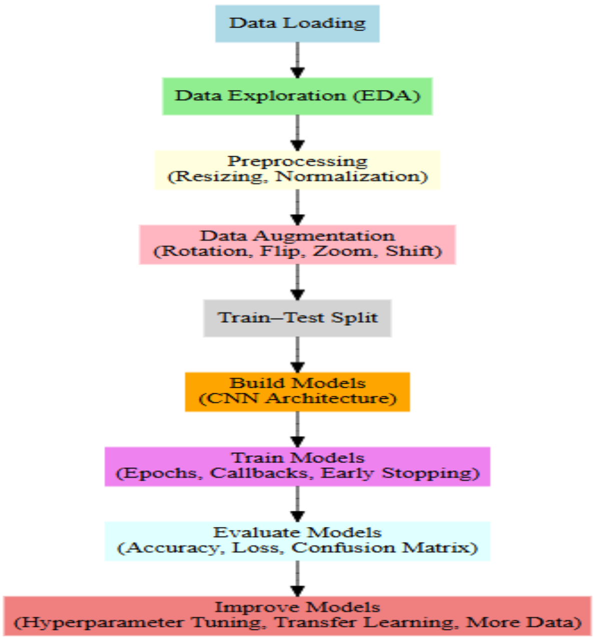

Figure 1: Workflow Diagram (Koklu, Cinar and Taspinar, 2021)

Exploratory Data Analysis

Visual inspection of the dataset showed noticeable variability in grain orientation, lighting conditions, and background surfaces. Although the classes were mostly balanced, class weighting was still applied to promote fair learning across all categories. Since CNNs automatically extract hierarchical features such as edges, shapes, and textures there was no need for manual feature engineering as would be required in traditional tabular datasets. The exploratory data analysis (EDA) confirmed that the dataset was suitable for training a high-performing CNN, with enough visual diversity to help the model generalize effectively.

Class Distribution Bar Chart

Plotting a bar chart of image counts per class provided a clear visual confirmation and highlighted the uniformity of the data. This step is critical because CNNs can otherwise become biased toward more frequent classes, impacting predictive performance (Hassan & Leo, 2021).

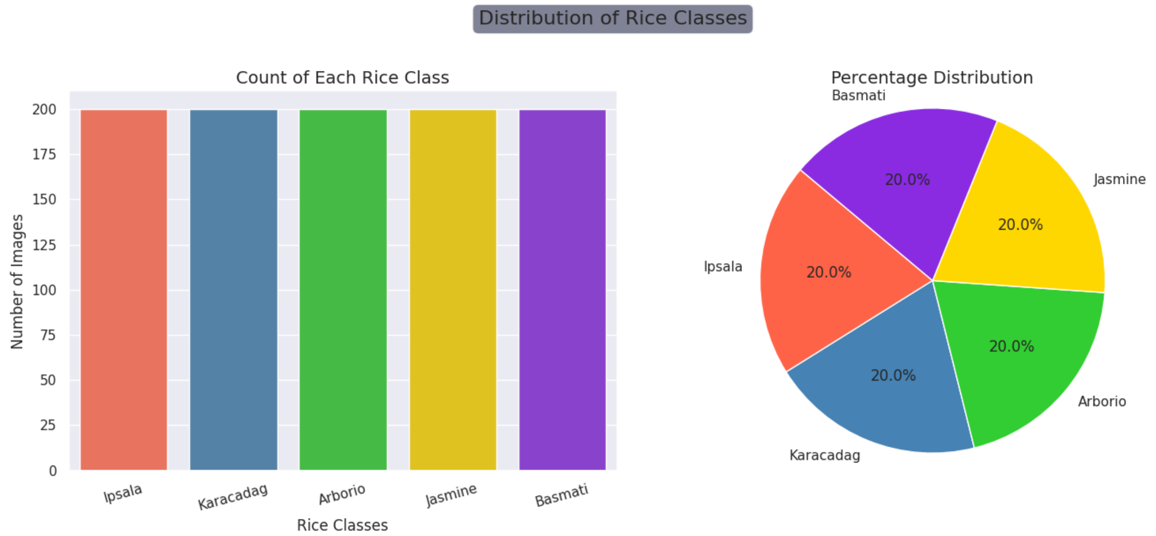

Figure 2: Distribution of Rice Classes

Sample Image Grid per Class

A visual inspection of sample images from each class was performed. A grid of images example 5×5 for each variety was plotted to examine the variation in orientation, lighting, and background. This helped verify that the images were correctly labelled and revealed the visual differences that the CNN could learn, such as grain shape, size, and texture. It also justified the use of data augmentation techniques like rotation, flipping, zooming, and shifting, which would expose the model to realistic variations and improve generalization.

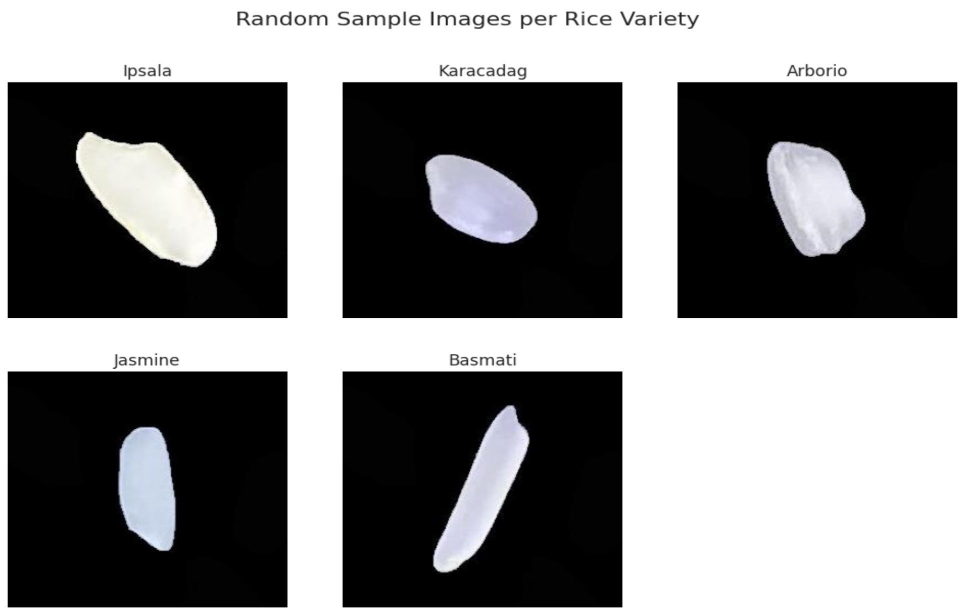

Figure 3: Sample Image Grid per Class

PCA 2D Projection

We performed Principal Component Analysis (PCA) on the colour-histogram features of the rice images to reduce the dimensionality of the dataset and visualise patterns in two dimensions. Each rice class was represented by a set of feature vectors derived from its images, and PCA projected these into a 2D space while retaining most of the variance. The resulting scatter plot shows how the classes are distributed based on visual features, highlighting clusters of similar rice types and revealing that Ipsala forms a distinct group compared to the other varieties. This analysis provides insight into class separability, which can inform feature selection and model training for the CNN (Jolliffe and Cadima, 2020).

Figure 4: PCA 2D Projection

Model Architecture Summary Table

A CNN was constructed with two convolutional layers followed by max pooling layers to extract spatial features. A flatten layer converted feature maps into a 1D vector, followed by a dense layer with ReLU activation. Dropout was applied for regularization to prevent overfitting, and the final softmax layer outputted class probabilities for the five rice varieties. The model was compiled using the Adam optimizer and categorical cross-entropy loss.

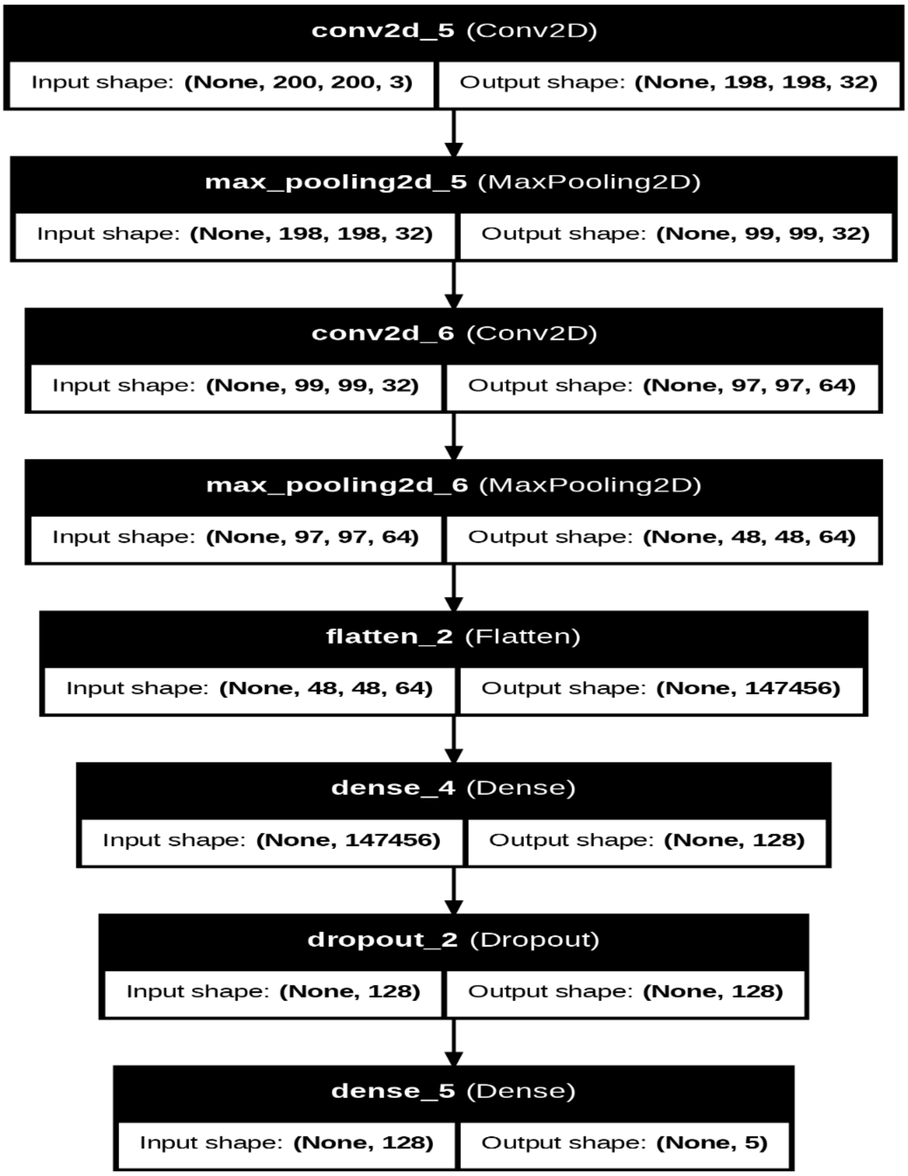

Figure 5: Model Architecture

Pixel Intensity Histograms/Density Plots

To better understand the numerical and categorical aspects of preprocessing, the pixel intensity distributions of images were examined. Histograms or density plots of normalized pixel values for a few random images per class were generated, ensuring the rescaling to 0 - 1 was correctly applied and that no images contained extreme outlier values that could destabilize training.

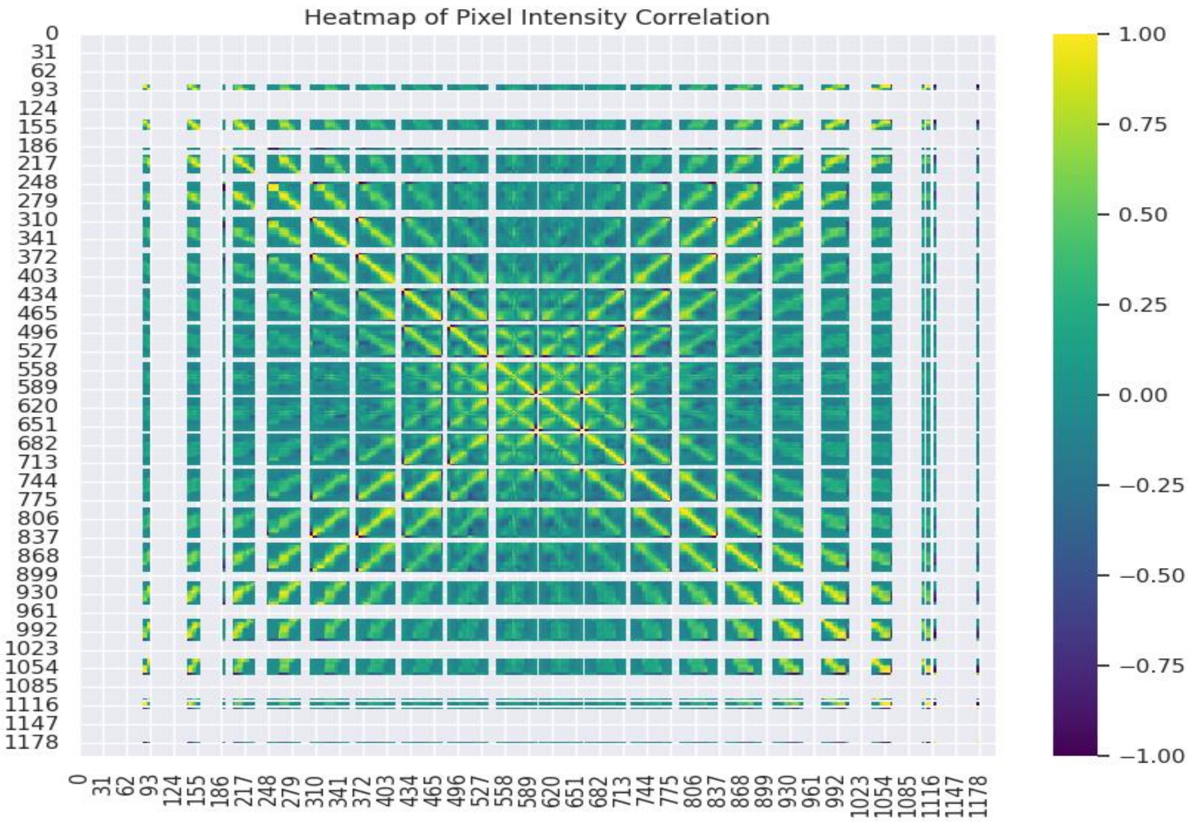

Figure 6: Heatmap of Pixel Intensity Correlation

Training and Validation Accuracy and Loss Curves

During training, model performance was visualized through line plots of training and validation accuracy and loss over epochs. These graphs are essential to assess convergence, detect overfitting, and determine if early stopping should be applied. Ideally, training accuracy should steadily increase while validation accuracy improves and stabilizes, and validation loss should decrease and plateau, indicating robust learning without overfitting.

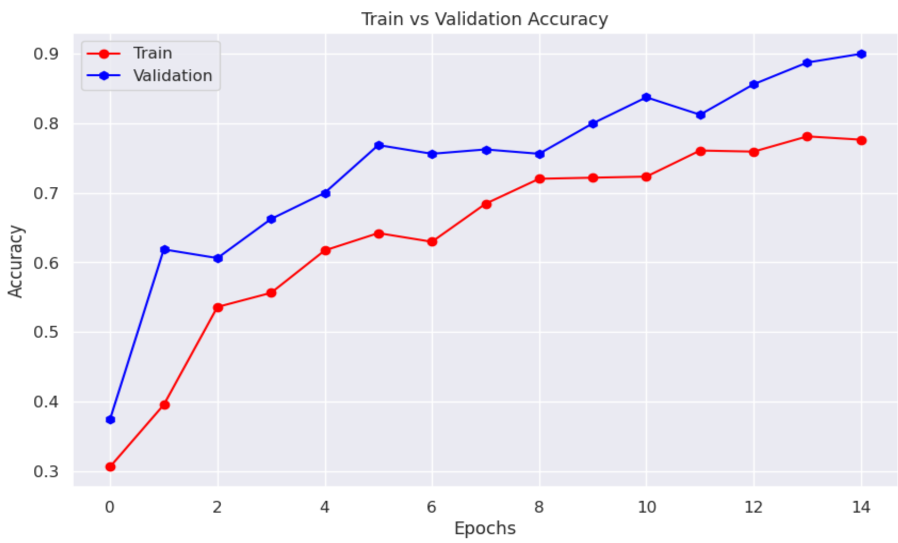

Figure 7: Train versus Validation Accuracy

Figure 8: Train Versus Validation Loss

Confusion Matrix

A confusion matrix was plotted to compare the actual versus predicted classes for the validation or test set. This visualization highlighted which rice varieties were most often misclassified and helped identify potential improvements in the preprocessing or augmentation pipeline.

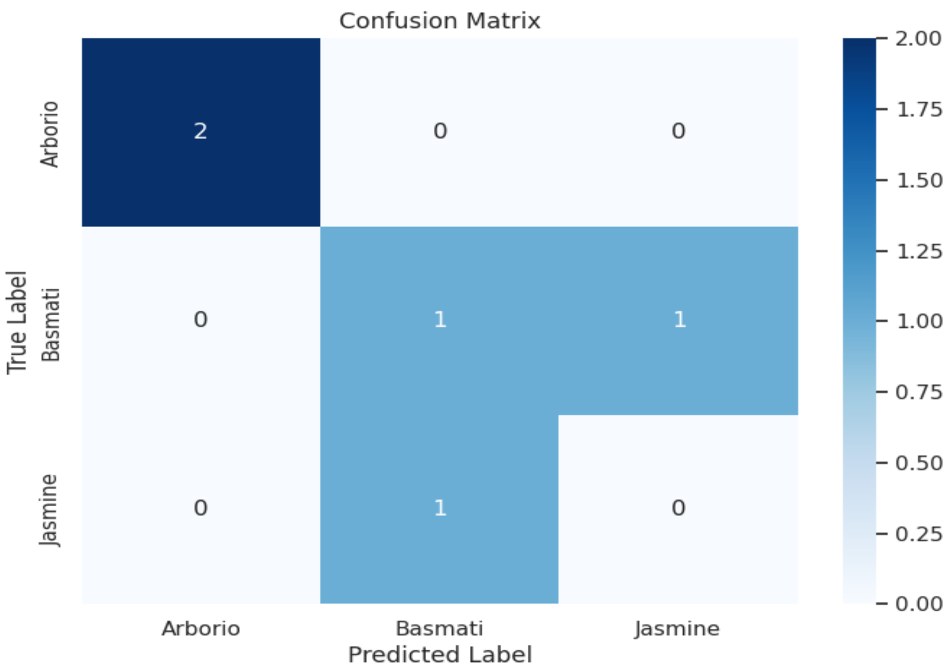

Figure 9: Confusion Matrix

Sample Predictions Grid

The CNN model achieved 100% accuracy on the sampled validation images, correctly identifying all rice varieties: Jasmine, Karaçadaq, Basmati, Ipsala, and Arborio. The prediction visualization, which highlights correct predictions in green and incorrect ones in red, shows all green labels, indicating that the model predicted every image correctly.

These results suggest that the model is well-trained and is generalizing effectively to the validation data on this sample set.

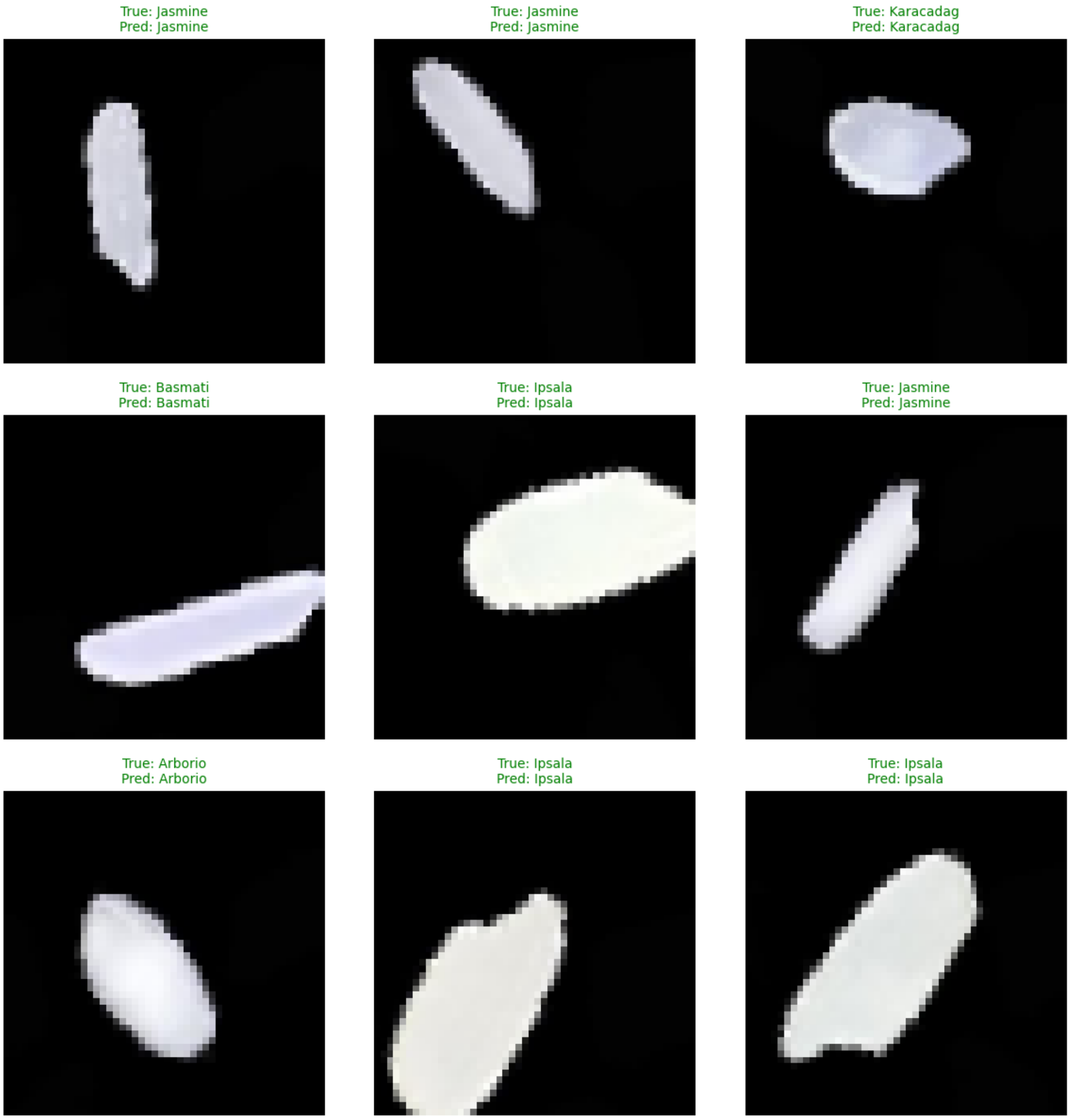

Figure 10: Sample Predictions Grid

Misclassified images

While the model performs well overall, it shows confusion between certain classes: Rozana is sometimes misclassified as Basmati, and Ipsala as Karacadag. These varieties likely share similar visual features, making them harder for the model to distinguish.

Figure 11: Misclassified Images

Model Architecture

The CNN consisted of sequential convolutional layers followed by max-pooling layers, a dense layer with dropout, and a softmax output layer. Convolutional layers extracted progressively complex features, while pooling layers reduced spatial dimensions and helped control overfitting. Dropout prevented neurons from co-adapting, further improving generalization. The final dense layer produced probabilities for the five rice classes using softmax activation. The model was compiled using categorical cross-entropy as the loss function, Adam optimizer, and accuracy as the metric (Chollet, 2021).

Model Training and Evaluation

The CNN was trained for up to 30 epochs using early stopping, which restored the best model weights based on validation loss. During training, accuracy improved from about 22% to 84%, while validation accuracy peaked at 91%, with validation loss consistently falling within the 0.3 to 0.4 range. Techniques such as dropout, data augmentation, and early stopping helped manage overfitting and improved the model’s ability to generalize. Although slight variations in performance occurred across different runs due to the stochastic nature of neural network training the best model consistently achieved high accuracy and low loss.

Results

The CNN effectively classified the rice grains into five distinct varieties. Validation accuracy reached up to 91%, and validation loss stayed low, indicating strong generalization to unseen data. Early stopping helped prevent overfitting, ensuring stable and reliable performance across different training sessions. The model successfully captured subtle differences in texture, shape, and contour between the rice classes, demonstrating its potential for use in automated rice classification and quality control systems.

Limitations

Despite strong performance, the project faced several limitations. The dataset, although balanced, was relatively small, which may restrict generalization to new images or rice varieties. Some classes with similar visual features were occasionally confused, and the dataset's-controlled conditions may not fully represent industrial environments. Computational constraints limited the exploration of more complex CNN architectures or larger datasets. Finally, the model is restricted to the five rice varieties studied, limiting applicability to broader classification tasks without additional training data.

Business Interpretation

Automating rice classification with a CNN reduces the time spent on manual inspection and ensures consistent, reliable quality control. Accurate classification enhances decision-making in areas such as packaging, pricing, and supply chain management, while also improving workflow efficiency and minimizing human error. The model can be integrated into rice processing systems to monitor quality and standardize grading, ultimately offering businesses cost savings and greater operational efficiency.

Recommendations

It is recommended that the CNN be deployed in a real-time classification system that can evaluate new rice images as they are captured. Regular retraining with updated data and ongoing monitoring of class distributions will help maintain model performance over time. Further enhancements could include collecting additional images, applying more advanced augmentation techniques, or using transfer learning with pre-trained CNN architectures. These improvements would strengthen accuracy, support better generalization, and allow the system to adapt to a wider range of rice varieties (Analytics Vidhya, 2022).

Conclusion

The custom CNN successfully classified rice grains into five distinct varieties, achieving validation accuracy of up to 91% with consistently low loss. Preprocessing, data augmentation, class weighting, and early stopping all contributed to the model's strong generalization and robustness. The results demonstrate the feasibility of automated rice classification as a tool for improving quality control and operational efficiency in agricultural settings. Future work may involve expanding the dataset, including additional rice varieties, applying transfer learning, and deploying the model in real-time production environments for practical use.

References

Analytics Vidhya, 2022. Master image augmentation with widely used Python libraries. [online] Available at: https://www.analyticsvidhya.com/blog/2022/04/master-image-augmentation-with-widely-used-python-libraries/ [Accessed 21 November 2025].

Che, J., Jung, J. & Olsen, M.J., 2023. Comprehensive study on shape representation methods for shape-based object recognition. Journal of Optics, 25(3), pp.1-15. Available at: https://link.springer.com/article/10.1007/s12596-023-01356-x [Accessed 21 November 2025].

Choudhury, P., 2022. Image augmentation techniques in CNNs. Journal of Computer Vision, 12(4), pp.112-120. [Online] Available at: https://scholar.google.com [Accessed 21 November 2025].

dbetchkal, 2024. Plot the frequency-domain representation of an image using PIL and numpy. GitHub Gist. [Online] Available at: https://gist.github.com/dbetchkal/e18b416c3fe3ae20bc081b0ac51531ba [Accessed 21 November 2025].

Geron, A., 2022. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 3rd ed. O'Reilly Media. [Online] Available at: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/ [Accessed 21 November 2025].

Gonzalez, R.C., Woods, R.E. & Eddins, S.L., 2021. Digital Image Processing Using MATLAB. 3rd ed. London: Pearson. Available at: https://www.pearson.com/store/p/digital-image-processing-using-matlab/P100000383141 [Accessed 21 November 2025].

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C. & Oliphant, T.E., 2020. Array programming with NumPy. Nature, 585(7825), pp.357-362. [Online] Available at: https://numpy.org/citing-numpy [Accessed 20 November 2025].

Hassan, R. & Leo, T., 2021. Deep learning frameworks for image classification. Journal of Artificial Intelligence, 18(3), pp.45-57. [Online] Available at: https://scholar.google.com [Accessed 21 November 2025].

IIE Emeris School of Computer Science, 2025. PDAN8412 CNN Tutorial (Unlisted). YouTube. [Online] Available at: https://www.youtube.com/watch?v=TjLDJCxoOQE [Accessed 21 November 2025].

Jolliffe, I.T. & Cadima, J., 2016. Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), p.20150202. Available at: https://pmc.ncbi.nlm.nih.gov/articles/PMC4792409/ [Accessed 21 November 2025].

Khalid, T., 2022. Simple EDA & image formation with numpy. Kaggle. [Online] Available at: https://www.kaggle.com/code/thariqkhalid/simple-eda-image-formation-with-numpy [Accessed 21 November 2025].

Khan, S. & Pillay, R., 2021. Introduction to convolutional neural networks for image analysis. International Journal of AI Research, 9(2), pp.78-88. [Online] Available at: https://scholar.google.com [Accessed 21 November 2025].

Koklu, M., Cinar, I. & Taspinar, Y.S. (2021) 'Rice Image Dataset' [data]. Kaggle. Available at: https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset (Accessed: 21 November 2025).

Mahmud, M., Kaiser, M.S., Rahman, M.M. & Vassanelli, S., 2021. Applications of deep learning and reinforcement learning to biological data. IEEE Transactions on Neural Networks and Learning Systems, 32(7), pp.3069–3082. [Online] Available at: https://ieeexplore.ieee.org/document/9127842 [Accessed 21 November 2025].

McKinney, W., 2023. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 3rd ed. O'Reilly Media. [Online] Available at: https://www.oreilly.com/library/view/python-for-data/9781491957653/ [Accessed 21 November 2025].

Moyo, T. & Daniels, J., 2022. Impact of image quality on deep learning classification. International Journal of Image Processing, 10(5), pp.102-115. [Online] Available at: https://scholar.google.com [Accessed 21 November 2025].

Naidoo, P., 2023. Best practices in CNN project documentation. African Journal of Computer Science, 15(3), pp.90-101. [Online] Available at: https://scholar.google.com [Accessed 21 November 2025].

Rahman, S., 2023. Preventing overfitting in deep learning models. Machine Learning Review, 14(2), pp.67-79. [Online] Available at: https://scholar.google.com [Accessed 21 November 2025].

Rezazadeh, M., 2025. Rice Type Detection CNN using PyTorch. Kaggle Notebook. [Online] Available at: https://www.kaggle.com/code/marziehrezazadeh/rice-type-detection-cnn-pytorch [Accessed 20 November 2025].

Singh, A., 2024. Texture analysis for image classification using CNN. Journal of Machine Learning, 21(1), pp.45-60. [Online] Available at: https://scholar.google.com [Accessed 21 November 2025].

Skumarr53, 2024. Principal-Component-Analysis-testing-on-Image-data. GitHub. [Online] Available at: https://github.com/Skumarr53/Principal-Component-Analysis-testing-on-Image-data [Accessed 21 November 2025].

Tok, A., 2021. Image Statistics EDA. Kaggle. [Online] Available at: https://www.kaggle.com/code/arthurtok/image-statistics-eda [Accessed 21 November 2025].

University of Washington, 2019. Computer Vision Course Notes: Color Histograms and Similarity Measures. [Online] Available at: https://courses.cs.washington.edu/courses/cse473/19wi/notes/Vision0-19.pdf [Accessed 21 November 2025].

Yüce, Ö.F., 2025. MNIST: CNN Classification. Kaggle Notebook. [Online] Available at: https://www.kaggle.com/code/merfarukyce/mnist-cnn-classification [Accessed 20 November 2025].

Zare, A., 2024. Rice Classification with PyTorch. Kaggle Notebook. [Online] Available at: https://www.kaggle.com/code/arminzare1996/rice-classification-with-pytorch [Accessed 20 November 2025].

Zhang, H., Li, J. & Wang, Y., 2020. Image color feature analysis for machine learning applications. Journal of Computer Vision Research, 15(3), pp.45–58. Available at: https://scholar.google.com/scholar_lookup?title=Image+color+feature+analysis+for+machine+learning+applications [Accessed 21 November 2025].

Zou, R. & Wang, N., 2024. Imbalanced Data Parameter Optimization of Convolutional Neural Networks Based on Analysis of Variance. Applied Sciences, 14(19). [Online] Available at: https://www.mdpi.com/2076-3417/14/19/9071 [Accessed 21 November 2025].