Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 9 (2027) 3432

Journal of the Nigerian Society of Physical Sciences

Rice classification via convolutional neural network and transfer learning

Salamatu Osanga Ibrahim*, Muhammad Dahiru Liman, Habila Alaku Akolo, Jeffrey Ovre Agushaka

Department of Computer Science, Federal University of Lafia, Nasarawa, Nigeria

Abstract

This study classifies 20 rice varieties using three custom-built CNN architectures and four transfer-learning models: MobileNetV2, VGG16, ResNet50, and DenseNet121. The Bangladesh rice dataset, comprising original and augmented sets, was used to address the challenges of limited training data and poor model generalization. The performance of the proposed custom architectures and transfer-learning models was compared using accuracy, precision, recall, and F1-score. Custom Architecture 1 achieved the highest classification accuracy of 98.51%. Among the proposed architectures, Custom Architecture 1 achieved the highest classification accuracy on the combined dataset (original and augmented training sets), compared with 87.94% on the original dataset. The results show an increase in classification accuracy for the combined dataset, although the magnitude of the change varied across architectures. ResNet50 outperformed all transfer-learning models on the original dataset, whereas VGG16 achieved the highest accuracy of 97.87% on the combined dataset. In addition, image-resolution analysis shows that increasing the dataset size together with high-resolution inputs produced the best performance when compared with other resolutions.

DOI: 10.46481/jnsps.2027.3432

Keywords: Rice, Classification, Data augmentation, Transfer learning, Convolutional neural networks

Article History:
Received: 15 April 2026
Received in revised form: 14 September 2026
Accepted for publication: 24 September 2026
Available online: 03 November 2026

© 2027 The Author(s). Published by the Nigerian Society of Physical Sciences under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

Communicated by: B. J. Falaye

1. Introduction

Rice (Oryza sativa) is a favorite and one of the world's most sought-after cereal crops, serving as a staple and source of food to more than 60% of the world's population thereby contributing significantly to curtailing malnutrition, a solution for food security, and as a source for economic development, specifically in developing countries [1].

It is a grain, rich in carbohydrate and used by an estimated number of about three billion people worldwide, with Asia topping the list as the major consumer and producer of Rice, thereby making China the leading contributor to the resultant global output of (28.9%). [2, 3]

Rice is important for the economy because it is grown, eaten, imported, and exported in many countries. It is among the main crops that support and boost agricultural economies. [4]

Rice exists in thousands of varieties worldwide, each distinguished by specific genetic, physical, and commercial characteristics such as color, shape, texture, aroma, taste, and cooking properties. [5]

The above-mentioned traits determine the grade, price, and choice of rice available in the market [6]. Despite these known traits, so many types of rice still look very similar, most especially with the rice grown in the same location. This makes the rice separation physically difficult and often almost impossible

*Corresponding Author Tel. No.: +234-706-655-9140.
Email address: siosanga@gmail.com (Salamatu Osanga Ibrahim)

[7]. For these stated reasons, farmers, dealers, and exporters can therefore lose valuable time and money if they do not correctly identify what their consumers need as this can lead to production of low-quality food and safety concerns [8].

In the past, people who were experts at looking at grains for colour, texture, and shape were the main way to classify rice varieties. These manual methods take a lot of time, money, and work, and they are also more likely to make mistakes because of tiredness, personal bias, not enough light, and human vision problem [9].

Other methods include physical and chemical methods of rice identification [10], but chemical identification using near-infrared spectroscopy is rapid and non-destructive [11].

The issue of rice being intentionally or unintentionally adulterated also creates a big trust issue for importers, exporters, and consumers as this goes to portray how important it is to not only have an automated way to identify rice also on how reliable the method is [12].

The sharp rise in global rice production and demand [8], has made it even more important to accurately and effectively classify rice varieties so as to meet up issues such as; quality control, seed certification, crop management, storage, and transportation decisions. Additionally, advances in computer vision, machine learning, and deep learning have also played a vital role in the automatic, fast, and objective classification of rice varieties, which seems to be a promising alternative to traditional methods [13].

One of these methods is convolutional neural networks (CNNs), which has been very successful at classifying images because they can automatically find unique features without needing to be preprocessed by hand [14].

Despite the excellent performance of deep learning models there are still issues like lack of model generalization and overfitting due to lack of large and diverse dataset, limited use of custom-built architecture, and transfer learning techniques. To address these issues three custom CNN architectures were built and trained on Bangladesh rice dataset and also, four transfer learning models were trained on the same dataset to classify rice varieties into 20 different categories. Additionally in order to evaluate the enhancement of classification accuracy, generalization, and reliability of these models, their performances were systematically compared. In essence, this study seeks to enhance quality assurance, reduce fraud [15] and adulteration [12], and facilitate informed decision-making for farmers, traders, regulators, and consumers within the rice value chain [16] through the automation of rice variety identification utilizing deep learning.

2. Literature review

2.1. Manual approach for rice classification

Classifying rice varieties by hand was one of the early employed method in history. This known method relies basically on people who know a lot about rice to tell the different kinds available, based on its color, texture, and shape. A major setback with this method is that it still takes a lot of time, needs a human expert, and it is easy to make a lot of mistakes [9].

2.2. Traditional machine learning methods for classifying rice

The setback earlier stated with manual approach was experienced before the advent of a normal machine learning approach. During the early days of research on rice variety categorization, hand-crafted variables like colour, texture, form, and morphological descriptors were often employed with standard machine learning techniques. These included Support Vector Machines (SVM), Artificial Neural Networks (ANN), Multi-Layer Perceptron's (MLP), Random Forest (RF), k-Nearest Neighbour (KNN), and Neuro-Fuzzy systems.

For instance, Sun *et al.* [17] made used of SVM on an approximated dataset of 1,700 rice grains to measure chalky rice kernels of Indica rice and Japonica rice and got an estimated accuracy of 98.5% and 97.6% respectively. Liu *et al.* [18] used only 200 samples (100 filled and 100 unfilled grains) to measure the percentage of rice filled grain. SVM has lowest missing rate and false detection rate compared to the other techniques. Pazoki *et al.* [19] also showed that classical models work well when they are used on well-designed features. They did this by classifying five types of rice with MLP and Neuro-Fuzzy techniques and getting accuracies of over 99%.

Some relevant researchers have also shown that the traditional method is quite effective too, some notable examples include the use of ANN and image processing as demonstrated by Abbaspour-Gilandeh *et al.* [20]. The work carried out looked at 13 types of Iranian rice and found non-linear links ranging from colour, morphology, to texture. Also, Cinar and Koklu [21] tested a lot of machine learning classifiers on a large dataset of about 75,000 rice samples and reported that MLP had the highest accuracy at 99.91%. Additionally, Saxena *et al.* [22] also discovered and reported that Random Forest was the best classifier out of the several classifiers he used. The researcher reported accuracies as high as 99.85% on large datasets. In all, these models work well, but they can be affected by changes such as; lighting, grain orientation, and dataset quality. These high accuracies were achieved mostly due to the use of manually extracted features.

Traditional Machine learning models required feature extraction but the proposed custom architectures and Transfer Learning Models used in this study automatically extract features.

2.3. Emergence of deep learning and CNN-based techniques

Deep learning has made convolutional neural networks (CNNs) the best at classifying different types of rice. CNNs can automatically find distinguishing features in images. Lin *et al.* [23] used a deep CNN to sort three types of rice with a 95.5% accuracy rate on 7,399 samples. Ahmed *et al.* [24] also used CNNs for rice grain disease classification, but they only used a smaller dataset of 200 samples and got an accuracy rate of 88.07%. These results show that the size of the dataset has a big effect on how well CNN works. This is one of the reasons this study used large dataset.

. CNNs are believed to be superior to conventional machine learning techniques, as stated in a number of studies, thereby making CNN-based methods a better choice as against traditional models in terms of classification accuracy and efficiency,

these studies are as demonstrated by Li *et al.* [25] and Deng [26]. Additionally, in a comprehensive study, Koklu *et al.* [14] also tested ANN, DNN, and CNN models using an estimated 75,000 rice photos from five different kinds of sources. Even though all the models did well, CNN had the best success rate, showing that it can handle image-based rice classification. In comparison, this study also used CNN but with large and diverse dataset.

2.4. Hybrid CNN architectures and attention mechanisms

In order to improve feature learning even more, attention mechanisms have recently been added to CNNs with a view to categorizing 14 different types of rice, Meng *et al.* [27] created a CNN with a self-attention mechanism that performed better than Random Forest and KNN models. In a similar vein, Meng *et al.* [28] also reported an overall accuracy of 98.93% with the use of a hybrid CNN and a trainable attention module. These studies demonstrated that focusing on the most informative regions of rice grain images significantly enhances CNN performance as wellHybrid Deep Learning Models have also gained the attention of some researchers, notably among whom are Verma *et al*. [29] who presented a Hybrid Convolutional Neural Network models (AlexNet+ResNet50+EfficientNet-b1) that achieved about 99.92% accuracy in rice variety classification. Also, Mamun *et al.*'s Hybrid Deep Learning Architecture outperformed VGG16 and MobileNetV2, with an accuracy of 99% demonstrating the value of mixing several deep learning techniques [30]. Even though the models used in this study are CNN, they are not hybrid.

2.5. Transfer learning for rice-variety classification

Transfer Learning Models have performed very well even with small amount of data due to their ability to transfer knowledge from one task to another. Researchers such as Hoai *et al.* 13] used the VNRICE dataset to test a number of pretrained deep learning models and found that DenseNet121 was the best, with an accuracy of 99.04%. When compared to pretrained models like InceptionV3 and InceptionResNetV2, Din *et al.* [12] also showed that RiceNet, portrayed the greatest accuracy of 94%.

Such outsatnding performances have also been reported with ResNet-based models in several investigations, while Razavi *et al.* [8] mixed ResNet designs with transfer learning to classify six rice varieties to achieve an accuracy of 99.85%. Shrivastava *et al.* [31] used ResNet20 to classify rice grains into five groups. Further evidence that refined ResNet models outperform other transfer learning architectures with accuracies exceeding 98% was shown by Prasetyo [32] and Dhaneesh *et al.* [16]. All the studies involving transfer learning models used small number of classes. This study used 20 rice varieties.

2.6. Alternative and spectral-based methods

Some studies investigated spectrum and hyperspectral methods in addition to RGB image-based approaches. Shi *et al.* [7] successfully classify four rice types using Near-Infrared (NIR) spectroscopy in conjunction with PCA and SIMCA. Kang *et al.* [33] also noted and reported an achieved accuracy of 95.3% in dimensionality reduction with the use of fluorescent hyperspectral technology when used along with PCA and t-SNE. To further categorize ten different types of rice, Weng *et al.* [34] introduced PCANet, which was derived from PCA and outperformed KNN and Random Forest with an accuracy of 98.66%.

2.7. Rice quality, fraud detection, and disease categorization

Fraud detection and quality evaluation are directly related to the classification of rice varieties. Various studies have shown that while Chen *et al.* [35] utilized SVM to classify faulty and broken rice grains with an accuracy of 99.3%, Pradana-Lopez *et al.* [36] concentrated on identifying low-quality rice types. Further highlighting the adaptability of deep learning in rice-related applications, disease-related studies by Shrivastava *et al.* [37], Ghosal and Sarkar [38] also portrayed the viability of CNN-based and transfer learning models in rice disease identification methods.

2.8. Challenges

A number of issues with regards to rice variety classification still exist despite rapid advancements. An issue such as model generalization is limited by the fact that a lot of studies have reported the use of a small or a less diverse dataset [39, 40]. The outcome is that performance is impacted by visual similarities across rice varieties, uneven manual labelling, class imbalance, overfitting, and excessive computational complexity. To address these compounding issues, Prova [39] suggested an attention-based CNN for categorizing 20 different types of rice, this method achieved an accuracy of 91.92%. However, more gains were noted when features were merged with KNN. Overall, the literature demonstrates a clear transition from traditional machine learning methods to deep learning and transfer learning approaches. While the use of pretrained models with CNNs have achieved remarkable performance, the need for large, diverse datasets and robust architecture also remains critical and sacrosanct in achieving reliable, and laudable real-world rice variety classification.

2.9. Research gap

There are still a lot of important holes in the huge amount of research on how to classify different types of rice. First, most of the research done today only uses a small number of rice varieties, usually between three and sixteen classes [23, 24, 27]. Some relevant and recent studies [40, 42] have also looked at larger class sizes, about 20 or more kinds, but these studies needed a lot of computer power, this in turn made them less useful in real life.

Second, dataset limitations are still a big problem. A significant proportion of earlier research utilised small to moderately sized datasets, generally comprising several hundred to a few thousand samples [7, 13, 18]. When used in real-world situations, small datasets make models less resilient and more likely to overfit and not generalise well, especially for deep learning models. As demonstrated by Koklu *et al.*, [14] and Cinar &

Table 1: Summary of related work.

Authors	Methods	No. of Rice Varieties	Dataset	Key Features Approach	Best Accuracy
Pazoki *et al.* [19]	MLP, Neuro-Fuzzy	5	450	Shape, color, morphology	99.73%
Lin *et al.* [23]	Deep CNN	3	7,399	Automatic feature extraction	95.5%
Ahmed *et al.* [24]	CNN	3	200	RGB images	88.07%
Weng *et al.* [34]	PCANet	10	4320	Morphology, texture, spectroscopy	98.66%
Abbaspour-Gilandeh *et al.* [20]	ANN	13	180	Color, morphology, texture	89.2%
Hoai *et al.* [13]	DenseNet121 (TL)	6	10508	Transfer learning	99.04%
Koklu *et al.* [14]	ANN, DNN, CNN	5	75,000	Image-based features	CNN best
Shrivastava *et al.* [31]	ResNet20	5	1000	Transfer learning	98%
Meng *et al.* [28]	Hybrid CNN + Attention	14	2100	Trainable attention module	98.93%
Cinar & Koklu [21]	MLP, RF, SVM, KNN	5	75,000	Color, shape, morphology	99.91% (MLP)
Shi *et al.* [7]	NIR + SIMCA	4	200	PCA + spectroscopy	Reliable
Din *et al.* [12]	RiceNet (CNN)	5	4748	Custom CNN	94%
Meng *et al.* [27]	CNN + Self-Attention	14	2100	Attention-based learning	CNN > RF, KNN
Razavi *et al.* [8]	ResNet34/50 (TL)	6	15000	Transfer learning	99.85%
Mamun *et al.* [30]	Hybrid DL	5	20,000	CNN-based hybrid	99%
Rajalakshmi *et al.* [40]	RiceSeedNet	13	13,000	Custom CNN	97–99%
Sharma *et al.* [41]	ResNet50 (TL)	22	600,000	Large-scale dataset	80.5%
Prova [39]	Attention-CNN	20	27,000	Attention + CNN	91.92%
Dhaneesh P *et al.* [16]	ResNet50 (TL)	5	4900	Fine-tuned ResNet	98.12%

Koklu, [21] they used large a datasets, but the setback realized was that they mostly focused on a small number of rice type.

Thirdly, a lot of conventional machine learning techniques rely mostly on manually created feature extraction, which is susceptible to illumination, grain orientation, and arbitrary preprocessing decisions [19, 22]. Although deep learning models mitigate this constraint, data imbalance, computational complexity, and inadequate utilization of model comparisons across architectures continue to limit their performance.

Fourthly, there is little comparison analysis between numerous transfer learning models and custom-built CNN architectures. The majority of research either suggest a single bespoke design [12, 40] or only concentrate on pretrained networks [8, 13] without conducting a comprehensive and cohesive performance comparison using the same dataset and experimental settings.

Lastly, there is still a lack of attention to the problem of large-scale multi-class categorization and dataset diversity. There is a vacuum in scalable and generalizable rice classification systems since attention-based and hybrid models have demonstrated encouraging results [28, 39], but their application has frequently been limited to fewer varieties or limited dataset diversity.

2.10. Novelty of the present study

First, this study classifies rice varieties into 20 classes,

whereas most previous studies used fewer classes; the larger-class studies cited here reported lower performance [42,40].

Second, the study uses the the large and diverse Bangladesh rice datasetset to address overfitting and model-generalization limitations reported in studies using smaller datasets.

Third, the study compares custom-built architectures and transfer learning models under the same experimental settings, including the dataset, number of epochs, and batch size, to provide insight into model performance and behaviour.

Finally, the models are evaluated for generalization and reliability to support potential real-world use in rice-adulteration detection, quality control, and rice inspection.

2.11. Contribution to knowledge

This work contributes to knowledge by providing reliable models for rice classification. The outcome effectively tackles challenges related to dataset size, class diversity, architectural design, and comparative assessment. To cap it all, the results displayed support precision, agriculture and food quality assurance by connecting experimental research with real-world use.

3. Materials and methods

3.1. Materials

The following materials were used in this research, rice dataset and Google Colab. Software versions: TensorFlow (2.20.0), Python (3.13.15), and NumPy (2.1.3).

3.1.1. Rice dataset

The dataset used in the experiment was obtained from Mendeley data [42].The dataset contains two folders, the original folder which contains the original images without augmentation and the augmented folder which contains augmented data. The dataset consists of 20 different varieties of Bangladesh rice from different regions. These varieties are 1_Subol_Lota, 2_Bashmoti, 3_Ganjiya, 4_Shampakatari, 5_Katarivog, 6_BR28, 7_BR29, 8_Paijam, 9_Bashful, 10_Lal_Aush, 11_Jirashail, 12_Gutisharna, 13_Red_Cargo,14_Najirshail, 15_Katari_Polao, 16_Lal_Biroi, 17_Chinigura_Polao, 18_Amon, 19_Shorna5, 20_Lal_Binni. To ensure diversity, the data was gathered from marketplaces and rural areas. iPhone 11 camera was used to capture these images with a 5x zoom feature.

Table 2 show the class distribution of Bangladesh rice dataset.

Original dataset

The class range in original dataset is minimum of 230 images (Bashmoti, BR28) and maximum of 256 images (Gutisharna). The difference is 26 images while the ratio of the largest to the smallest is approximately 1.11. This means the largest class has only 11.3% more samples than the smallest. For this reason, the dataset is approximately balance.

Combined Dataset

For the combined dataset, the minimum is 1380 images and the maximum is 1536 images. The difference is 156 images. The ratio of the largest class to the smallest class is 11.3% same with the original dataset.

Table 3 shows the statistical summary of the original and combined datasets.

The two datasets are approximately balanced across the 20 rice varieties. The original dataset contains images between 230 and 256 per class while the combined dataset ranges from 1380 to 1536 images per class. The small difference is the reason why no additional class-balancing technique was required during training.

Original image. This folder contains all the original images of 20 different rice varieties in JPG format. A total of 4730 images were captured in uniform size of 853×853 pixels.

Augmented image. This folder contains all the augmented images. These images were also saved in JPG format with a uniform size of 512×512 pixels. A total of 23,650 images were generated.

In this research, the original images were used to train, evaluate, and test the model architectures and also the original image training set was combined with the augmented images to form another dataset and used to train model architectures. The new training set now contains 27,450 images all together.

3.1.2. Google Colaboratory

The entire experiments were conducted using Google Colab because of the free GPU. All the models were trained using GPU due to its speed and because it does not need installing dependencies. The GPU used in this research is T4 GPU with Python 3 runtime.

3.2. Methods

Here the workflow used in this research is presented. This includes data collection, data preprocessing, model design, model training, and model evaluation.

3.2.1. Data collection

The data used in this research is from a secondary source as stated in section 3.1.1.

3.2.2. Data preprocessing

The RGB pixels were normalized from [0,255] to [0,1] for the different experiments (224×224, 160×160, and 128×128) and analyses involving custom architectures. For the transfer learning models, the study used the exact preprocess input function of Keras for each model. MobileNetV2 used `tf.keras.applications.mobilenet_v2.preprocess_input`, VGG16 used `tf.keras.applications.vgg16.preprocess_input`, ResNet50 used `tf.keras.applications.resnet50.preprocess_input`, and DenseNet121 used `tf.keras.applications.densenet.preprocess_input`. These functions transform the input images into the format that was used when the pretrained models were originally trained on ImageNet. ResNet50 and VGG16 convert RGB (Red, Green, Blue) images into BGR (Blue, Green, Red) format and

Table 2: Class distribution.

S/N	Class Name	Original data	Augmented data	Combined data
1	Subol_Lota	232	1160	1392
2	Bashmoti	230	1150	1380
3	Ganjiya	233	1165	1398
4	Shampakatari	234	1170	1404
5	Katarivog	233	1165	1398
6	BR28	230	1150	1380
7	BR29	233	1165	1398
8	Paijam	234	1170	1404
9	Bashful	252	1260	1512
10	Lal_Aush	231	1155	1386
11	Jirashail	231	1155	1386
12	Gutisharna	256	1280	1536
13	Red_Cargo	240	1200	1440
14	Najirshail	231	1155	1386
15	Katari_Polao	242	1210	1452
16	Lal_Biroi	231	1155	1386
17	Chinigura_Polao	237	1185	1422
18	Amon	250	1250	1500
19	Shorna5	238	1190	1428
20	Lal_Binni	232	1160	1392
Total		4730	23650	28,380

Table 3: Statistical summary of the Bangladesh rice dataset.

Dataset	Minimum	Maximum	Mean	Largest/Smallest Ratio
Original	230	256	236.5	1.11
Combined	1380	1536	1419	1.11

subtract the ImageNet channel means. The ImageNet means for the blue, green and red channels are 103.939, 116.779, and 123.680 respectively.

MobileNetV2 scales pixel values to the range [-1, 1], whereas DenseNet121 scales them to [0, 1], and subsequently standardize them using the ImageNet mean and standard deviation. Data augmentation was performed using transformations such as Random horizontal flipping, random rotation with a factor of 0.1, and random zoom with a factor of 0.1. The augmented folder also contained images generated using its own augmentation transformations [42].

The dataset was divided into training, validation, and test sets. For the original dataset, 4730 original images were first divided into training, validation, and test partitions using an 80:10:10 stratified split. The training, validation, and test set consists of 3800, 460, and 470 respectively. Data augmentation was subsequently applied only to the training partition. Consequently, all augmented derivatives of a given original image remained in the same partition as their source image, and no augmented derivatives of validation or test images were used for training. The split also ensured that all the 20 rice varieties were represented in the training, validation, and test partitions while approximately preserving their class proportions to avoid the risk of class imbalance.

This technique ensures that the training set is different from the validation set and the test set is entirely unseen during training and hyperparameter tuning.

For the combined dataset, the 23650 augmented images were only added to the training partition of the original data, and the new training set becomes 27,450. The validation and test set from the original images were used without any addition.

3.2.3. Data visualization

The data used in this research belongs to 20 different categories of rice. Below is a sample of the dataset.

3.2.4. Model design

To address the research gap, three custom architecturess were built and compared with the following transfer learning models namely; MobileNetV2, VGG16, ResNet50, and DenseNet121.

Custom architecture 1. This model consists of three convolutional layers and two dense layers with padding set to valid. The first convolutional layer has 32 neurons and 3×3 filter with ReLU activation function followed by max pooling layer. The



Figure 1: A sample of the Bangladesh rice dataset.

second convolutional layer had 64 neurons and 3×3 filter and a ReLU activation function followed by a max pooling layer. The third convolutional layer had 128 neurons with 3×3 filters and a ReLU activation function followed by a max pooling layer which was flattened and fed to a dense layer. The dense layer had 512 neurons with a ReLU activation function followed by a dropout with a 0.5 (50%) probability and a final layer, dense for classification with a softmax activation function. The architecture is depicted below.

Custom architecture 2. This model consists of four convolutional layers and two dense layerss with padding set to valid. The first convolutional layer had 32 neurons and 3×3 filter with ReLU activation function followed by max pooling layer. The second convolutional layer had 64 neurons and 3×3 filter and a ReLU activation function followed by a max pooling layer. The third convolutional layer had 128 neurons with 3×3 filters and a ReLU activation function followed by a max pooling layer. The fourth convolutional layer had 256 neurons with 3×3 filters and a ReLU activation function followed by a max pooling layer which was flattened and sent to a dense layer. The dense layer had 512 neurons with a ReLU activation function followed by a dropout with a 0.5 (50%) probability and a final layer, dense for classification with a softmax activation function. The architecture is depicted below.

Custom architecture 3. This model consists of five convolutional layers and two dense layerss with padding set to valid. The first convolutional layer had 32 neurons and 3×3 filter with ReLU activation function followed by max pooling layer. The second convolutional layer had 64 neurons and 3×3 filter and a ReLU activation function followed by a max pooling layer. The third convolutional layer had 128 neurons with 3×3 filters and a ReLU activation function followed by a max pooling layer. The fourth convolutional layer has 256 neurons with 3×3 filters and a ReLU activation function followed by a max pooling layer. The fifth convolutional layer has 512 neurons with 3×3 filters with a ReLU activation function followed by a max pooling layer which was flattened and sent to a dense layer. The dense layer had 512 neurons with a ReLU activation function followed by a dropout with a 0.5 (50%) probability and a final layer, dense for classification with a softmax activation function. The architecture is depicted below.

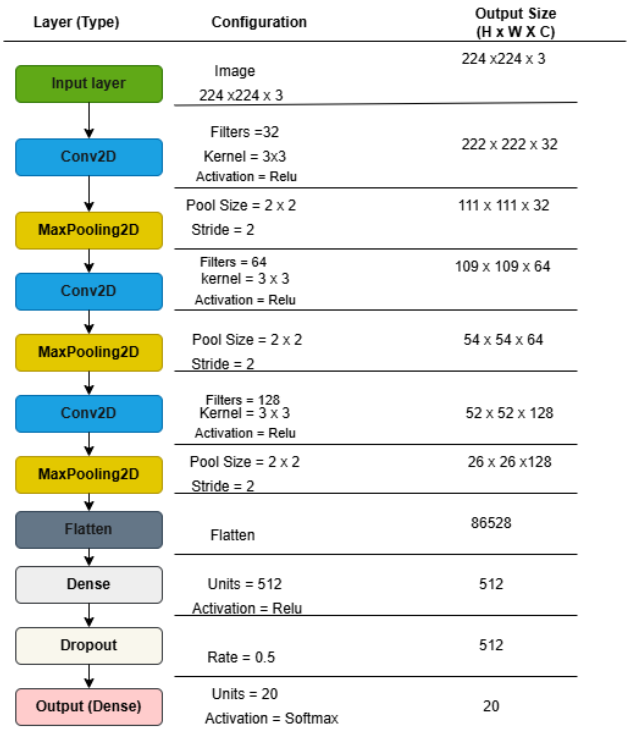


Figure 2: Custom CNN architecture 1.

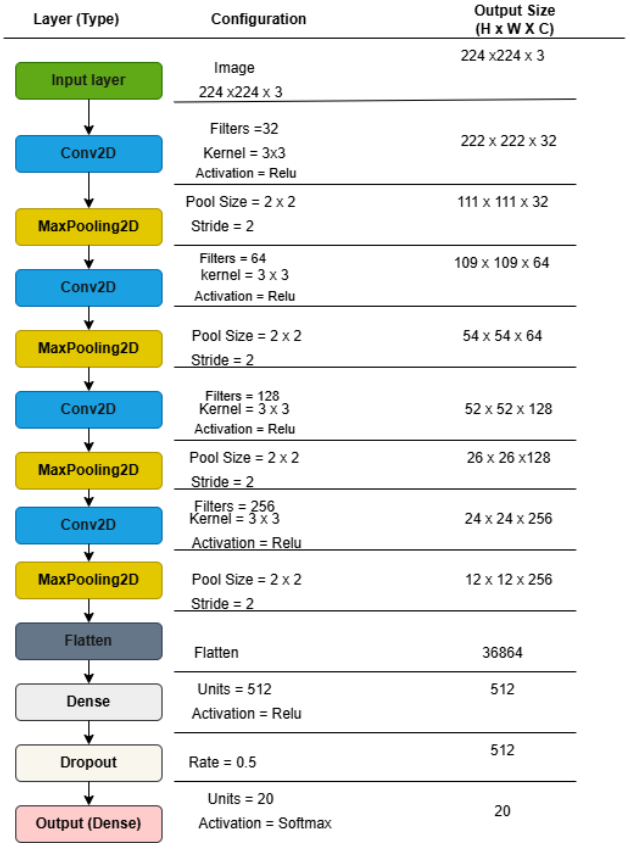


Figure 3: Custom CNN architecture 2.

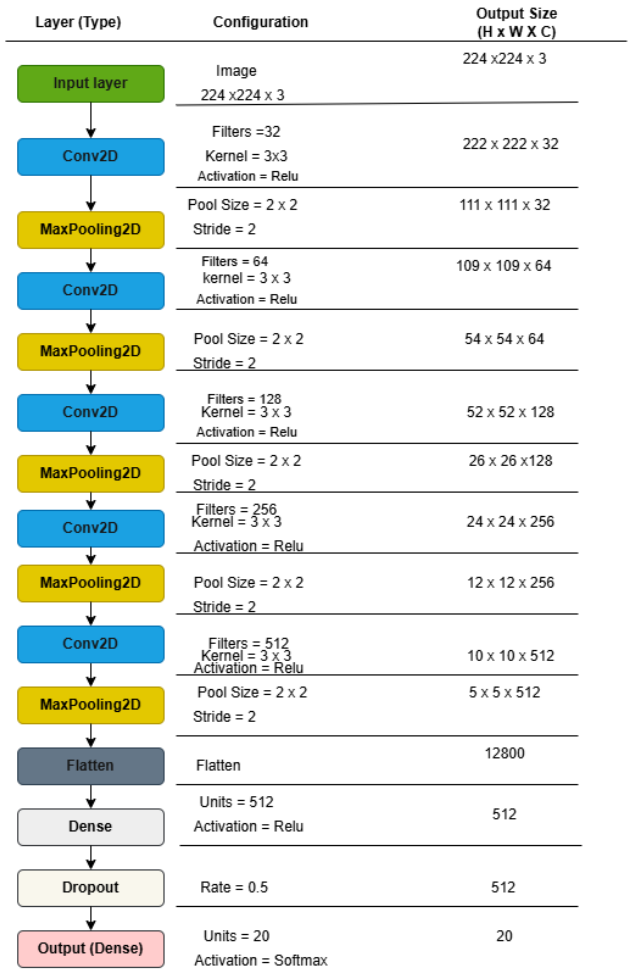


Figure 4: Custom CNN architecture 3.

MobileNetV2. Howard *et al.* [43] proposed MobileNet in 2017 for mobile applications and embedded devices. To achieve their target they used depth-wise convolution, pointwise convolution, and width multiplier to reduce the model size.

VGG16. In 2014 ImageNet Challenge, Simonyan and Zisserman [44] proposed VGG architecture based on AlexNet and it achieved an excellent accuracy of 92.3%, among the top 5. The VGG variant used in this research is VGG16.

ResNet50. He *et al.* [45] proposed ResNet in 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and won the competition. The corresponding paper was subsequently published at CVPR 2016. Their residual learning framework addressed the vanishing gradient problem associated with increasing network depth through the residual connections. The submitted ensemble achieved a 3.57% top-5 error rate on the ImageNet test set. The ResNet variant used in this research is ResNet50.

DenseNet121. Huang *et al.* [46] proposed DenseNet. Their work won the Best Paper Award at the 2017 Conference on Computer Vision and Pattern Recognition. Their model was

Table 4: Custom-architecture parameter counts.

Model	Total Parameters	Trainable Parameters
Custom Architecture 1	44,406,356	44,406,356
Custom Architecture 2	19,273,556	19,273,556
Custom Architecture 3	8,132,948	8,132,948

inspired by the skip connection in ResNet. They introduced the dense block.

Table 4 shows parameter counts of the three custom architectures. Each architecture used padding set to valid.

3.2.5. Model training

The models were trained on the 80% allocated for training and 10% for validation. The models were trained on both the original and combined training datasets. A maximum of 100 epochs was used, with a batch size of 32. Random seeds for model initialization and data ordering were set to one of the three values: 42, 43, or 44. The selected seed was applied to os, random, NumPy, and TensorFlow, and was used for dataset creation (shuffling) and model initialization. The pretrained models were initialized with weights set to 'imagenet' while the newly added classification heads used the default Keras weight initialization. A fixed learning rate was used for each experiment. For the custom Architectures, a learning rate of 1e-3 was used on the original training dataset, while 1e-4 was used on the combined training dataset. For the pretrained models on original training dataset, a learning rate of 1e-3 was used during the frozen-base training phase and 1e-4 was used during the fine-tuning phase.

For the combined training dataset, a learning rate of 1e-4 was used during the initial training phase and 1e-5 during fine-tuning. The Adam optimizer was used throughout all experiments, and no learning rate scheduler was used. Early stopping and checkpoint restoration were implemented using tf.Keras.callbacks.EarlyStopping, which monitored validation loss, used a patience of 3 epochs, and restored the best weights. In addition, tf.Keras.callbacks.ModelCheckpoint was used to save the best performing weights otherwise, the previous best weights are retained or restored.

For the transfer learning models, early layers were frozen and later unfrozen. Classifier heads were trained for several epochs, top layers were unfrozen and fine-tuned for several epochs. Relu activation function was used throughout to introduce non-linearity and avoid vanishing gradient problem while softmax was used for the final classification.

3.2.6. Transfer learning model fine-tuning strategy

The fine-tuning strategy has two phases. In phase 1, which is feature extraction, it involves freezing the base model (i.e. the entire pre-trained model) by setting trainable to False. This means during the training process, the weights within these layers will not be updated. Followed by a new classification head on top of the frozen layers and the model is compiled and trained for a few epochs. Phase 2 involves fine-tuning the entire or partial unfrozen model. At this stage, the entire base model is set to trainable equal to True, this allows all the layers within the entire model potentially trainable. The model is then recompiled trained again for additional epochs. The details for all the models used.

MobileNetV2. Phase 1: Freeze the entire base model by setting trainable to False. Follow by a classification head consisting of a GlobalAveragePooling2D layer, a dense layer of 256 neurons with ReLU activation function, followed by a dropout layer with rate of 0.5(50%) probability, and a dense layer with softmax activation function to classify 20 rice varieties. The model is compiled with an Adam optimizer, learning rate of (1e-3 on Original training set, and 1e-4 on combined training data) and a categorical cross-entropy loss function with accuracy as the metric. A maximum of 100 epochs was set but the model trained for several epochs due to early stopping.

Phase 2: This phase starts by unfreezing the entire base model. Then followed by freezing all layers except the last 30 layers. For original training dataset, the model is then recompiled and trained for several epochs, using Adam optimizer with a lower learning rate of 1e-4 on original training data and 1e-5 on combined training data to prevent large weight updates and categorical cross-entropy. While the model only trained for several epochs while using the combined training dataset.

VGG16. Phase 1: Freeze the entire base model by setting trainable to False. Followed by a classification head consisting of a flatten layer, a dense layer with 512 neurons with ReLU activation function, followed by a dropout layer with 0.5 (50%) probability, and a dense layer with softmax activation function to classify 20 rice varieties. The model is compiled with an Adam optimizer, learning rate of 1e-3 on original training data and 1e-4 on combined training data, and a categorical cross-entropy loss function with accuracy as the metric. The model trained for several epochs out of 100 on the combined dataset and the original training dataset.

The phase 2 is the same as that of MobileNetV2.

ResNet50. Phase 1: Freeze the entire base model by setting trainable to False. Followed by a classification head consisting of a GlobalAveragePooling2D layer, a dense layer with 512 neurons with ReLU activation function, followed by a dropout layer with 0.5 (50%) probability, and a dense layer with softmax activation function to classify 20 rice varieties. The model is compiled with an Adam optimizer, learning rate of 1e-3 on original training data and 1e-4 on combined training data, and a categorical cross-entropy loss function with accuracy as the metric. The model trained for several epochs out of 100 on the combined training dataset and the original training dataset.

The phase 2 is the same as that of MobileNetV2.

Table 5: Confusion matrix.

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

DenseNet121. Phase 1: Freeze the entire base model by setting trainable to False. Followed by a classification head consisting of a GlobalAveragePooling2D layer, a dense layer of 512 neurons with ReLU activation function, followed by a dropout layer with 0.5(50%) probability, and a dense layer with softmax activation function to classify 20 rice varieties. The model is compiled with an Adam optimizer, learning rate of 1e-3 on original training data and 1e-4 on combined training data, and a categorical cross-entropy loss function with accuracy as the metric. The model trained for several epochs out of 100 on the original training dataset and the combined training dataset.

The phase 2 is the same as that of MobileNetV2.

3.2.7. Model evaluation

The models were evaluated on the test set using accuracy, precision, recall, and F1-score.

Confusion matrix. This provides a summary of prediction results. where TP is true positive, FP is false positive, FN is false negative, and TN is true negative.

Accuracy. Accuracy is the proportion of total correct predictions. It is appropriate for balanced datasets but can be misleading for imbalanced datasets.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (1)$$

Precision. This focuses on false positives. That is among all the predicted positive instances, how many are actually positive?

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (2)$$

Recall. This focuses on false negatives. That is among all the actual positive instances, how many did we correctly identify?

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (3)$$

F1-score. This is the harmonic mean of recall and precision. It is useful as a single metric for imbalanced data. It balances precision and recall.

$$F_1 = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (4)$$

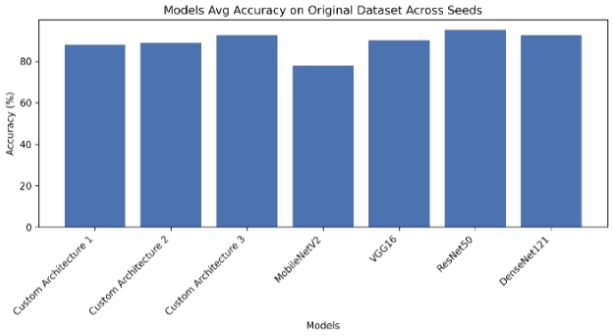


Figure 5: Model accuracy on the original dataset.

4. Results

The results of this research work are presented in this section under the following themes; Comparative analysis of custom-built architectures vs Transfer learning models, and Comparative analysis of Custom built models based on Image resolutions. These two themes will help to address the research gaps stated earlier and also highlight this research novelty.

The results presented here are average and standard deviation results of three independent runs using random seeds 42, 43, and 44.

4.1. Comparative analysis of custom architectures vs. transfer learning models

The comparison here is between custom-built architecture and transfer learning models on rice dataset (original and combined dataset).

Based on the outcome of Table 6 above, the best performing models are ResNet50 (95.18%), DenseNet121 (92.70%) and Custom Architecture 3 (92.41). An unexpected result is the relatively low performance of MobileNetV2 (77.80%).

Custom Architecture 1 achieved an accuracy of 87.94% while Custom Architecture 2 achieved 88.93%. Custom Architecture 3 was the best performing custom model, with an accuracy of 92.41%. The performance of the custom architectures suggests an effect of model depth on classification performance, as demonstrated by the progression from the shallow model (Custom Architecture 1) with 87.94%, to the moderately deep model (Custom Architecture 2) with 88.93%, and finally to the deeper model (Custom Architecture 3) with 92.41%.

For the transfer learning models, ResNet50 achieved an accuracy of 95.18%, while DenseNet121 achieved 92.70%. MobileNetV2 achieved the lowest accuracy among all the pretrained models.

Figure 5 shows the performance of all models on the original dataset, where ResNet50 achieved the highest accuracy.

Based on Table 7, increasing the dataset size through augmentation improved classification performance.

For the custom architectures, performances improved with augmented dataset, which suggest that data augmentation may improve model performance. Custom architecture 1 achieved the highest performance among the custom models.

Table 6: Model performance on the original dataset.

Model	Accuracy (%)	Macro Precision (%)	Macro Recall (%)	Macro F1 (%)
Custom Architecture 1	87.94 ± 3.46	88.56 ± 3.01	87.48 ± 3.64	87.32 ± 3.77
Custom Architecture 2	88.93 ± 0.93	89.34 ± 1.31	88.78 ± 1.13	88.15 ± 0.69
Custom Architecture 3	92.41 ± 1.00	91.58 ± 1.15	91.59 ± 1.13	91.30 ± 1.16
MobileNetV2	77.80 ± 9.54	82.24 ± 7.71	80.32 ± 7.88	77.98 ± 9.09
VGG16	90.14 ± 1.28	90.25 ± 1.69	89.67 ± 1.47	89.19 ± 1.40
ResNet50	95.18 ± 0.12	94.94 ± 0.27	94.32 ± 0.22	94.35 ± 0.18
DenseNet121	92.70 ± 1.62	93.24 ± 1.51	92.39 ± 1.63	92.31 ± 1.58

Table 7: Model performance on the combined dataset.

Model	Accuracy (%)	Macro Precision (%)	Macro Recall (%)	Macro F1 (%)
Custom Architecture 1	98.51 ± 1.13	98.54 ± 1.06	98.50 ± 1.06	98.40 ± 1.17
Custom Architecture 2	96.10 ± 0.54	96.28 ± 0.53	96.52 ± 0.19	96.03 ± 0.33
Custom Architecture 3	96.53 ± 2.34	96.58 ± 2.06	96.41 ± 2.25	96.19 ± 2.41
MobileNetV2	90.36 ± 3.38	90.67 ± 3.01	89.86 ± 3.52	89.81 ± 3.35
VGG16	97.87 ± 0.56	97.76 ± 0.67	97.56 ± 0.75	97.57 ± 0.74
ResNet50	96.85 ± 1.38	97.02 ± 1.00	96.36 ± 1.70	96.35 ± 1.60
DenseNet121	93.85 ± 1.52	94.38 ± 1.05	93.40 ± 1.44	93.27 ± 1.40

For the pretrained models (transfer learning), the models generally benefited from the increased amount of training data.

In general, increasing the training data through augmentation was associated with improved classification performance across the models.

In addition, accuracy is not the only metric used, below is the confusion matrix for custom architecture 3. Macro-averaging was used because the dataset was approximately balanced, ensuring that each class contributes equally to the overall model performance.

4.1.1. Capacity vs. data relationship

This analysis examines model capacity vs data relationship. Error reduction was also used to measure model improvements when comparing against a ceiling of 100%.

Table 9 shows custom architecture 1 has the highest error reduction rate while DenseNet121 has the lowest.

The positive signs in the change column indicate that all the models improved when trained on the combined dataset compared with the original dataset. On the original dataset ResNet50 outperformed all the other models while custom architecture 1 outperformed all models on the combined dataset.

Among the custom architectures, Custom Architecture 3 is the deepest of the three but has the lowest parameter capacity (8.1M / 3800) with 2140 parameters per sample. Custom Architecture 2 is smaller than Custom Architecture 3 in terms of depth but has a larger capacity (19.3 M / 3800), corresponding to 5072 parameters per sample. Custom architecture 1 which is the shallowest of the three has the largest parameter capacity with 11686 (44.4 M / 3800) parameters per sample on the original dataset and was outperformed by the two other architectures. When the dataset size increased sevenfold, Custom Architecture 1's ratio drops to 1618 parameters per sample. This reduction in the parameter per sample coincided with it largest observed gain in accuracy (+10.57), with Custom Architecture 1 achieving 98.51% on the combined dataset. The parameter counts do not include the pre-trained models because they don't train from scratch unlike the custom architectures.

Figure 7 illustrates the results on the combined dataset, where Custom Architecture 1 outperformed all models.

4.2. Comparative analysis of custom architectures by image resolution

In this section, three custom-built architectures are compared based on image resolution (high, medium, and low). For fair comparison, the data split, training, validation, and test set were the same for all the experiments. The custom architectures, loss function (categorical cross-entropy), learning rate (1e-3 for original training set, and 1e-4 for combined training dataset), optimizer (Adam), batch size (32), number of epochs (100), early stopping criteria were kept unchanged. The only thing that changed during these experiments was the image resolution.

Table 10 shows the effect of image size. Three resolutions were used 224×224 (high resolution), 160×160 (medium resolution), and 128×128 (low resolution). Image resolution affects performance but its impact depends on both the model depth, capacity, and dataset size.

On the original dataset, resolution barely matters and the trend is not consistent. The best resolution is not the same for all architectures. Architecture 1 highest performance is at 160×160 (90.07%), Architecture 2 also peaks at 160×160 (91.42%, though 128×128 is nearly identical at 91.35%), and

Table 8: Custom Architecture 3 (seed 42) classification report.

Class	precision	recall	f1-score	Support
10_Lal_Aush	1.000000	0.944444	0.971429	18.000000
11_Jirashail	0.944444	0.944444	0.944444	18.000000
12_Gutisharna	1.000000	1.000000	1.000000	43.000000
13_Red_Cargo	1.000000	1.000000	1.000000	27.000000
14_Najirshail	1.000000	1.000000	1.000000	18.000000
15_Katari_Polao	1.000000	0.965517	0.982456	29.000000
16_Lal_Biroi	1.000000	1.000000	1.000000	18.000000
17_Chinigura_Polao	1.000000	1.000000	1.000000	24.000000
18_Amon	1.000000	1.000000	1.000000	37.000000
19_Shorna5	1.000000	1.000000	1.000000	25.000000
1_Subol_Lota	0.950000	1.000000	0.974359	19.000000
20_Lal_Binni	1.000000	1.000000	1.000000	19.000000
2_Bashmoti	0.944444	1.000000	0.971429	17.000000
3_Ganjiya	0.952381	1.000000	0.975610	20.000000
4_Shampakatari	1.000000	1.000000	1.000000	21.000000
5_Katarivog	1.000000	1.000000	1.000000	20.000000
6_BR28	0.944444	1.000000	0.971429	17.000000
7_BR29	1.000000	0.900000	0.947368	20.000000
8_Paijam	1.000000	1.000000	1.000000	21.000000
9_Bashful	1.000000	1.000000	1.000000	39.000000
accuracy	0.989362	0.989362	0.989362	0.989362
macro avg	0.986786	0.987720	0.986926	470.000000
weighted avg	0.989806	0.989362	0.989315	470.000000

Table 9: Overall comparison.

Model	Original (%)	Combined (%)	Change	Error Reduction
Custom Architecture 1	87.94	98.51	+10.57	87.65%
Custom Architecture 2	88.93	96.10	+7.17	64.77%
Custom Architecture 3	92.41	96.53	+4.12	54.28%
MobileNetV2	77.80	90.36	+12.56	56.58%
VGG16	90.14	97.87	+7.73	78.40%
ResNet50	95.18	96.85	+1.67	34.65%
DenseNet121	92.70	93.85	+1.15	15.75%

Table 10: Custom-architecture performance by image size.

Models	Accuracy (%) (224×224) Original	Accuracy (%) (224×224) Combined	Accuracy (%) (160×160) Original	Accuracy (%) (160×160) Combined	Accuracy (%) (128×128) Original	Accuracy (%) (128×128) Combined
Custom Architecture 1	87.94 ± 3.46	98.51 ± 1.13	90.07 ± 1.81	96.88 ± 2.86	88.58 ± 0.33	93.12 ± 2.39
Custom Architecture 2	88.93 ± 0.93	96.10 ± 0.54	91.42 ± 0.81	95.67 ± 3.51	91.35 ± 2.28	94.18 ± 4.48
Custom Architecture 3	92.41 ± 1.00	96.53 ± 2.34	90.92 ± 2.46	87.73 ± 3.96	92.55 ± 0.77	88.58 ± 3.37

Architecture 3 peaks at 128×128 (92.55%). There is no clear trend but they all performed well.

Based on the above Figure 8, the performance measured in accuracy is good across all the three architectures when using the three resolutions.

On the combined dataset, resolution matters a lot with

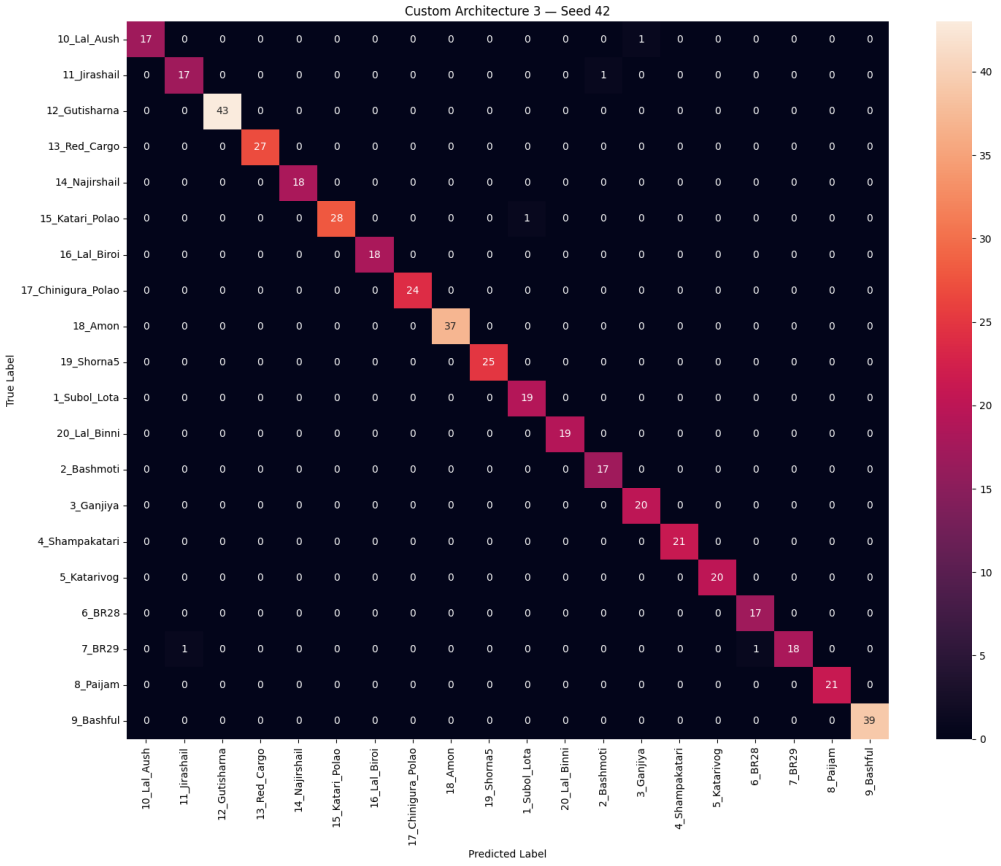


Figure 6: Confusion matrix of Custom Architecture 3.

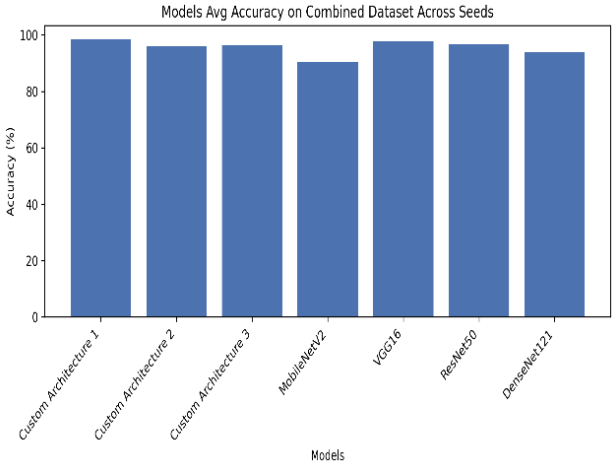


Figure 7: Model accuracy on the combined dataset.

all three architectures achieving their highest performance at 224×224 resolution but with different sensitivity. Architecture 3 is the most fragile, from 96.53% to 87.73% to 88.58%, that is an 8percentage-point drop from full to reduced resolution, and its variance roughly triples (±2.34 → ±3.96/±3.37). Architecture 1 is moderately sensitive with a 5.4percentage-point drop (98.51% → 93.12%). Architecture 2 is the most resolution-robust with a 1.9percentage-point drop from 96.10% to 94.18%.

Table 11: Image-size performance on the original dataset.

Model	224×224	160×160	128×128
Custom Architecture 1	87.94 ± 3.46	90.07 ± 1.81	88.58 ± 0.33
Custom Architecture 2	88.93 ± 0.93	91.42 ± 0.81	91.35 ± 2.28
Custom Architecture 3	92.41 ± 1.00	90.92 ± 2.46	92.55 ± 0.77

Table 12: Image-size performance on the combined dataset.

Model	224×224	160×160	128×128
Custom 1	98.51 ± 1.13	96.88 ± 2.86	93.12 ± 2.39
Custom 2	96.10 ± 0.54	95.67 ± 3.51	94.18 ± 4.48
Custom 3	96.53 ± 2.34	87.73 ± 3.96	88.58 ± 3.37

Using the combined dataset as presented in Figure 9, the experiments reveals that performance (accuracy) of the three architectures using 224×224 is the best. However, performance

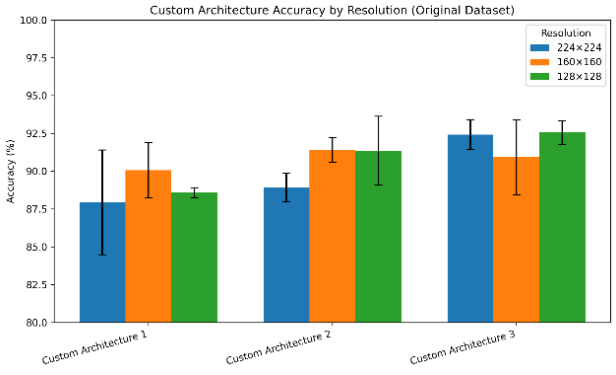


Figure 8: Model accuracy by image resolution using the original dataset.

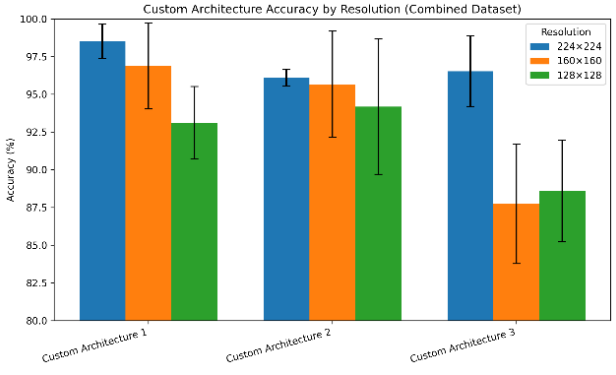


Figure 9: Model accuracy by image resolution using the combined dataset.

at 160×160 and 128×128 were also very good and close to 224×224 for Custom Architecture 1 and 2.

4.2.1. Resolution vs. model complexity

Experimental results show that custom architecture 1, which is shallow performed better on 224×224 (high) resolutions. Though its performance was also good on the other resolutions. The moderate architecture which is custom architecture 2 performed well irrespective of the resolutions, its performance is stable. The deepest architecture which is the custom architecture 3 performed well with all resolutions and small dataset.

4.2.2. Resolution sensitivity (original vs. combined)

This experiment reveals that when there is limited dataset then high resolution like 224×224 barely matters. However, if there is enough dataset then the issue of resolution becomes very important. 224×224 achieved the best result in this work. Finally, optimal resolution depends on the size of the dataset and the model architecture.

5. Discussion

In this research, two datasets (original and combined) were used. Findings from the experiment with the original dataset show that custom architecture 3 outperformed the two other custom architectures (accuracy) in terms of rice variety classification. For the combined dataset, results show the importance of data augmentation in increasing model accuracy especially for models with deeper architectures. To achieve a good performance, there is a need for the dataset size to align with the model complexity.

Model performance improved when trained on combined dataset (large) as compared to the original dataset (small), highlighting the importance of augmenting data when it is limited. All the models improved when trained with the combined dataset. The model that improved most is Custom Architecture 1. VGG16 showed the next-largest improvement. Other models like MobileNetV2, ResNet50, MobileNetV2, DenseNet121, Custom Architecture 2 and Custom Architecture 3 also performed well and showed improvements with increased data. All the custom architectures built, performed well and improved when trained on the combined dataset. The best custom architecture is custom architecture 1 which is shallow and has the largest capacity to successfully classify rice varieties with an accuracy of 98.51% on combined dataset. It is also noticed that models like Custom Architecture 3 and transfer learning models perform well on the original dataset which is small.

The two datasets were also used on the three custom architectures to analyse the effect of image resolutions, three categories of resolutions (high, medium, and low) were considered. Results show that higher resolution is more effective when the dataset is large; this is evident in the observed performance, and they all achieved their best performance with this resolution. However, for Custom Architectures 1 and 2 performance drops as the resolution is reduced on combined data. With small data (original data), the difference in performance across the three resolutions became less pronounced. It was also observed that custom architecture 3 performed well with the low resolution by achieving an accuracy of 92.55% which is the best it has achieved. This is also similar with custom architecture 2 achieving 91.42% with medium resolution 160×160. These results indicate that with small data, resolutions become less important to some models. Hence, optimal resolutions depend on the size of the dataset and the model architecture. Lower resolutions can provide good and comparable performance when there is enough data.

Hoai *et al.* [13] performed a comparative study of rice variety classification using handcrafted descriptors and deep learning models. VRNICE dataset of size 10508 was used with some deep learning models among them are ResNet50, VGG16, MobileNetV2, and DenseNet121. According to them, ResNet50 achieved an accuracy of 98.84, DenseNet121 achieved an accuracy of 99.05, MobileNetV2 achieved an accuracy of 97.66% and VGG16 achieved accuracy of 96.94. Dhaneesh *et al.* [16] compared six transfer learning models namely EfficientNetB7, DenseNet201, ResNet152V2, ResNet50, Xception, and VGG19 to classify rice husk and grains into 5 different varieties. Their findings showed that a fine-tuned ResNet50 proposed by them outperformed all the 6 transfer learning models by achieving an accuracy of 98.12%. Comparing Hoe *et al.* 2020, and Dhaneesh *et al.* [16] with this research, the dataset

here is diverse with 20 different varieties and it is also large (combined dataset). Prova *et al.* [39] used the same dataset as this work but with different models. The models they used are Attention based CNN, Random Forest, SVC, Logistic regression, KNN, XGBOOST, MLP, and CNN+LSTM.

They proposed a technique called CBAM attention mechanism. This technique was used to extract features. These features are used as input by the RF, SVC, LR, KNN, XGBOOST, MLP, and CNN+LSTM to classify rice varieties into 20 different categories. They concluded that the proposed technique enhanced their classifiers in classifying rice varieties into different classes which resulted in their best model (KNN classifier) achieving an accuracy of 99.35%. The only similarity between their work and this research is the dataset. Apart from that, most of the techniques they used are not the same with what was used for this work. Some of the major differences are their resizing of their images into 28×28 pixels for computational reasons, data spilt using 75:25 ratio, attention mechanism used to extract features, and different models were used. The present experiments used 224×224, 160×160, and 128×128. Data were split using 80:10:10 ratio. Features were extracted automatically by the deep learning models because of their inbuilt ability unlike the traditional machine learning models (like RF, SVC, KNN, LR etc.) that required handcrafted or pre-extracted features.

Furthermore, the models used here are all deep learning models. The performance of these models suggests that they benefited from the increased in training data through data augmentation.

6. Conclusion

This experiment addresses the problem of data by using the large and diverse Bangladesh rice dataset with 20 varieties and a total of 28,380 images. Three custom models and four transfer learning models were analysed and compared using the original dataset (smaller) and combined dataset (larger). The best custom model on the original dataset is custom architecture 3 and the best on the combined dataset is custom architecture 1. The best transfer learning model on the original dataset is ResNet50 and the best on the combined dataset is VGG16. The comparative analysis shows to a reasonable extent that model performance is highly dependent on the interactions between dataset size and architectural complexity. In this study, simpler and efficient models perform well on smaller datasets while complex architectures such as Custom Architecture 1 outperform other models given that sufficient data is available. Additionally, some models do benefit from data augmentation while others exhibit a high level of sensitivity to augmentation.

Several analyses done on image resolutions also revealed that an increase in dataset size together with high-resolution inputs produced higher classification accuracy.

For future research, precise automatic rice variety classification warrants further attention. In this regard, future work should investigate why some models do not benefit from data augmentation.

Data availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this manuscript.

Funding

This work is supported by the Federal University of Lafia Institution-Based Research Grant from TETFUND. The grant/award number is: FUL/REG/TETF/002/VOL.V1/326.

Acknowledgment

This work is supported by the Federal University of Lafia Institution-Based Research Grant from TETFUND. The grant/award number is: FUL/REG/TETF/002/VOL.V1/326.

References

[1] M. Zhao, Y. Lin & H. Chen, "Improving nutritional quality of rice for human health", Theoretical and Applied Genetics **133** (2020) 1397. https://doi.org/10.1007/s00122-019-03530-x.

[2] J. Chen, Y. Huang & Y. Tang, "Quantifying economically and ecologically optimum nitrogen rates for rice production in south-eastern China", Agriculture, Ecosystems & Environment **142** (2011) 195. https://doi.org/10.1016/j.agee.2011.05.005.

[3] R. Segal & M. Le Nguyet, *Unfair harvest: The state of rice in Asia*, Oxfam, 2019. https://oi-files-cng-v2-prod.s3.eu-west-2.amazonaws.com/asia.oxfam.org/s3fs-public/file_attachments/bp-unfair-harvest-rice-asia-200319-en.pdf.

[4] H. Ashraf, I. Murtaza, N. Nazir, A. B. Wani, S. Naqash & A. M. Husaini, "Nutritional profiling of pigmented and scented rice genotypes of Kashmir Himalayas", Journal of Pharmacognosy and Phytochemistry **6** (2017) 910. https://www.phytojournal.com/archives/2017/vol6issue6/PartM/6-6-142-977.pdf.

[5] T. Tipi, N. Yildiz, M. Nargelecekenler & B. Cetin, "Measuring the technical efficiency and determinants of efficiency of rice (Oryza sativa) farms in Marmara Region, Turkey", New Zealand Journal of Crop and Horticultural Science **37** (2009) 121. https://doi.org/10.1080/01140670909510257.

[6] J. D. Guzman & E. K. Peralta, *Classification of Philippine rice grains using machine vision and artificial neural networks*, World Conference on Agricultural Information and IT, Tokyo, Japan, 2008, pp. 41–48. https://api.semanticscholar.org/CorpusID:127366985.

[7] G. Shi, X. Zhang, G. Qu & Z. Chen, "Classification of rice varieties using SIMCA applied to NIR spectroscopic data", ACS Omega **7** (2022) 46623. https://doi.org/10.1021/acsomega.2c05561.

[8] M. Razavi, S. Mavaddati & H. Koohi, "ResNet deep models and transfer learning technique for classification and quality detection of rice cultivars", Expert Systems with Applications **247** (2024) 123276. https://doi.org/10.1016/j.eswa.2024.123276.

[9] S. J. M. Rad, F. A. Tab & K. Mollazade, *Classification of rice varieties using optimal color and texture features and BP neural networks*, 7th Iranian Conference on Machine Vision and Image Processing, Tehran, Iran, 2011, pp. 1–5. https://doi.org/10.1109/IranianMVIP.2011.6121583.

[10] S. Kamath, J. C. Stephen, S. Suresh, B. Barai, A. Sahoo, K. Radhika Reddy & K. R. Bhattacharya, "Basmati rice: Its characteristics and identification", Journal of the Science of Food and Agriculture **88** (2008) 1821. https://doi.org/10.1002/jsfa.3286.

[11] J. Chen, M. Li, T. Pan, L. Pang, L. Yao & J. Zhang, "Rapid and non-destructive analysis for the identification of multi-grain rice seeds with near-infrared spectroscopy", Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy **219** (2019) 179. https://doi.org/10.1016/j.saa.2019.03.105.

[12] N. M. U. Din, A. Assad, R. A. Dar, M. Rasool, S. U. Sabha, T. Majeed, Z. U. Islam, W. Gulzar & A. Yaseen, "RiceNet: A deep convolutional neural network approach for classification of rice varieties", Expert Systems with Applications **235** (2024) 121214. https://doi.org/10.1016/j.eswa.2023.121214.

[13] D. P. V. Hoai, T. Surinwarangkoon, V. T. Hoang, H.-T. Duong & K. Meethongjan, "A comparative study of rice variety classification based on deep learning and hand-crafted features", ECTI Transactions on Computer and Information Technology **14** (2020) 1. https://doi.org/10.37936/ECTI-CIT.2020141.204170.

[14] M. Koklu, I. Cinar & Y. S. Taspinar, "Classification of rice varieties with deep learning methods", Computers and Electronics in Agriculture **187** (2021) 106285. https://doi.org/10.1016/j.compag.2021.106285.

[15] M. Sliwinska-Bartel, D. T. Burns & C. Elliott, "Rice fraud a global problem: A review of analytical tools to detect species, country of origin and adulterations", Trends in Food Science & Technology **116** (2021) 36. https://doi.org/10.1016/j.tifs.2021.06.042.

[16] P. Dhaneesh, A. V. Prajeesh & V. Manohar, "Optimizing rice variety classification with transfer learning: A robust and scalable model for agricultural applications", Results in Engineering **28** (2025) 107570. https://doi.org/10.1016/j.rineng.2025.107570.

[17] C. Sun, T. Liu, C. Ji, M. Jiang, T. Tian, D. Guo, L. Wang, Y. Chen & X. Liang, "Evaluation and analysis the chalkiness of connected rice kernels based on image processing technology and support vector machine", Journal of Cereal Science **60** (2014) 426. https://doi.org/10.1016/j.jcs.2014.04.009.

[18] T. Liu, W. Wu, W. Chen, C. Sun, C. Chen, R. Wang, X. Zhu & W. Guo, "A shadow-based method to calculate the percentage of filled rice grains", Biosystems Engineering **150** (2016) 79. https://doi.org/10.1016/j.biosystemseng.2016.07.011.

[19] A. R. Pazoki, F. Farokhi & Z. Pazoki, "Classification of rice grain varieties using two artificial neural networks (MLP and neuro-fuzzy)", Journal of Animal and Plant Sciences **24** (2014) 336. https://psa.pastic.gov.pk/SearchArticleView.aspx?S_id=25131&articledetailId=28404.

[20] Y. Abbaspour-Gilandeh, A. Molaee, S. Sabzi, N. Nabipur, S. Shamshirband & A. Mosavi, "A combined method of image processing and artificial neural network for the identification of 13 Iranian rice cultivars", Agronomy **10** (2020) 117. https://doi.org/10.3390/agronomy10010117.

[21] I. Cinar & M. Koklu, "Identification of rice varieties using machine learning algorithms", Journal of Agricultural Sciences **28** (2022) 307. https://doi.org/10.15832/ankutbd.862482.

[22] P. Saxena, K. Priya, S. Goel, P. K. Aggarwal, A. Sinha & P. Jain, "Rice varieties classification using machine learning algorithms", Journal of Pharmaceutical Negative Results **13** (2022) 3762. https://www.researchgate.net/publication/367021922_RICE_VARIETIES_CLASSIFICATION_USING_MACHINE_LEARNING_ALGORITHMS.

[23] P. Lin, X. Li, Y. Chen & Y. He, "A deep convolutional neural network architecture for boosting image discrimination accuracy of rice species", Food and Bioprocess Technology **11** (2018) 765. https://doi.org/10.1007/s11947-017-2050-9.

[24] T. Ahmed, C. R. Rahman & M. F. M. Abid, "Rice grain disease identification using dual phase convolutional neural network-based system aimed at small dataset", arXiv (2020). https://arxiv.org/abs/2004.09870.

[25] H. Li, L. Zhang, H. Sun, Z. Rao & H. J. Ji, "Identification of soybean varieties based on hyperspectral imaging technology and one-dimensional convolutional neural network", Journal of Food Process Engineering **44** (2021) e13767. https://doi.org/10.1111/jfpe.13767.

[26] R. Deng, Y. Jiang, M. Tao, X. Huang, K. Bangura, C. Liu, J. Lin & L. Qi, "Deep learning-based automatic detection of productive tillers in rice", Computers and Electronics in Agriculture **177** (2020) 105703. https://doi.org/10.1016/j.compag.2020.105703.

[27] Y. Meng, W. Yuan, E. U. Aktilek, Z. Zhong, Y. Wang, R. Gao & Z. Su, "Fine hyperspectral classification of rice varieties based on self-attention mechanism", Ecological Informatics **75** (2023) 102035. https://doi.org/10.1016/j.ecoinf.2023.102035.

[28] Y. Meng, Z. Ma, Z. Ji, R. Gao & Z. Su, "Fine hyperspectral classification of rice varieties based on attention module 3D-2DCNN", Computers and Electronics in Agriculture **203** (2022) 107474. https://doi.org/10.1016/j.compag.2022.107474.

[29] D. S. Verma, M. Dafadar, J. K. Mishra, A. Kumar & S. Mahato, "AI-enable rice image classification using hybrid convolutional neural network models", International Journal of Intelligent Systems **2025** (2025) 5571940. https://doi.org/10.1155/int/5571940.

[30] M. A. Al Mamun, S. R. I. Karim, M. I. Sarkar & M. Z. Alam, "Evaluating the efficacy of hybrid deep learning models in rice variety classification", 2024. https://ijarise.org/index.php/ijarise/article/view/60.

[31] P. Shrivastava, K. Singh & A. Pancham, "Classification of grains and quality analysis using deep learning", International Journal of Engineering and Advanced Technology **11** (2021) 244. https://doi.org/10.35940/ijeat.A3213.1011121.

[32] S. Y. Prasetyo, *Optimizing rice variety recognition through transfer learning with pretrained ResNet models*, 4th International Conference of Science and Information Technology in Smart Administration (ICSINTESA), Balikpapan, Indonesia, 2024, pp. 231–235. https://doi.org/10.1109/ICSINTESA62455.2024.10748031.

[33] Z. Kang, R. Fan, C. Zhan, Y. Wu, Y. Lin, K. Li, R. Qing & L. Xu, "The rapid non-destructive differentiation of different varieties of rice by fluorescence hyperspectral technology combined with machine learning", Molecules **29** (2024) 682. https://doi.org/10.3390/molecules29030682.

[34] S. Weng, P. Tang, H. Yuan, B. Guo, S. Yu, L. Huang & C. Xu, "Hyperspectral imaging for accurate determination of rice variety using a deep learning network with multi-feature fusion", Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy **234** (2020) 118237. https://doi.org/10.1016/j.saa.2020.118237.

[35] S. Chen, J. Xiong, W. Guo, R. Bu, Z. Zheng, Y. Chen, Z. Yang & R. Lin, "Colored rice quality inspection system using machine vision", Journal of Cereal Science **88** (2019) 87. https://doi.org/10.1016/j.jcs.2019.05.010.

[36] S. Pradana-López, A. M. Pérez-Calabuig, C. Rodrigo, M. A. Lozano, J. C. Cancilla & J. S. Torrecilla, "Low requirement imaging enables sensitive and robust rice adulteration quantification via transfer learning", Food Control **127** (2021) 108122. https://doi.org/10.1016/j.foodcont.2021.108122.

[37] V. K. Shrivastava, M. K. Pradhan, S. Minz & M. P. Thakur, "Rice plant disease classification using transfer learning of deep convolution neural network", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **XLII-3/W6** (2019) 631. https://doi.org/10.5194/isprs-archives-XLII-3-W6-631-2019.

[38] S. Ghosal & K. Sarkar, *Rice leaf diseases classification using CNN with transfer learning*, IEEE Calcutta Conference (CALCON), Kolkata, India, 2020, pp. 230–236. https://doi.org/10.1109/CALCON49167.2020.9106423.

[39] N. N. I. Prova, "Enhancing agricultural research with an attention-based hybrid model for precise classification of rice varieties", International Journal of Cognitive Computing in Engineering **6** (2025) 412. https://doi.org/10.1016/j.ijcce.2025.02.002.

[40] R. Rajalakshmi, S. Faizal, S. Sivasankaran & R. Geetha, "RiceSeedNet: Rice seed variety identification using deep neural network", Journal of Agriculture and Food Research **16** (2024) 101062. https://doi.org/10.1016/j.jafr.2024.101062.

[41] K. Sharma, G. K. Sethi & R. K. Bawa, "A comparative analysis of deep learning and deep transfer learning approaches for identification of rice varieties", Multimedia Tools and Applications **84** (2025) 6825. https://doi.org/10.1007/s11042-024-19126-7.

[42] M. M. Islam, G. M. S. Himel, M. G. Moazzam & M. S. Uddin, "An image dataset of rice varieties", Mendeley Data, version 3 (2023). https://doi.org/10.17632/3mn9843tz2.3.

[43] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto & H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications", arXiv (2017). https://doi.org/10.48550/arXiv.1704.04861.

[44] K. Simonyan & A. Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv (2014). https://doi.org/10.48550/arXiv.1409.1556.

[45] K. He, X. Zhang, S. Ren & J. Sun, *Deep residual learning for image recognition*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

[46] G. Huang, Z. Liu, L. van der Maaten & K. Q. Weinberger, *Densely connected convolutional networks*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243.