Available online at http://scik.org
Commun. Math. Biol. Neurosci. 2023, 2023:132
https://doi.org/10.28919/cmbn/8278
ISSN: 2052-2541

TRANSFER LEARNING USING MOBILENET FOR RICE SEED IMAGE CLASSIFICATION

WAHYUDI AGUSTIONO[1], FIRDA AYU SAFITRI[2], WAHYUDI SETIAWAN[1,*], CAROLINE CHAN[3]

[1]Department of Information Systems, University of Trunojoyo Madura, Bangkalan, East Java 69192, Indonesia

[2]Department of Informatics, University of Trunojoyo Madura, Bangkalan, East Java 69192, Indonesia

[3]University of Newcastle, Newcastle Business School, Newcastle 2308, Australia

Abstract: Rice is the world's primary source of carbohydrates, especially in Asia. Quality rice requires good seed breeding. In this research, we classified rice seeds. The experiment using public data consists of five classes. Each class contains 2,000 images. The total amount of image data is 10,000. Classification uses mobileNet, which consists of 13 depthwise separable convolutions consisting of depthwise (DW) and pointwise (PW) convolutional layers. Each DW and PW is followed by batch normalization and Rectified Linear Unit activation. At the end, there is Global Average pooling and two dense layers. The trial uses transfer learning with initial weights from imageNet. The first to twelfth convolutional layers freeze. That is, they do not train the weights in them. On the 13th or last convolutional layer, fine-tuning is carried out. Experimental data is divided into training, validation, and testing. The testing results show that accuracy is 99.55%, precision 99.55%, recall 99.08%, and f1-score 99.31%.

Keywords: transfer learning; mobileNet; classification; rice seeds; convolutional neural network.

2020 AMS Subject Classification: 92B10.

1. INTRODUCTION

Rice is a staple food for half the world's population, especially in Asia, Africa, and Latin America. Rice comprises approximately 80% starch, 12% water, and 7% protein [1]. Rice has various types.

---

*Corresponding author
E-mail address: wsetiawan@trunojoyo.ac.id
Received October 12, 2023

Breeding superior seeds to produce quality rice is necessary. Seeds play an important role in rice growth. Seeds carry genetic characteristics for successful plant growth.

The selection of quality seeds aims to increase rice productivity. Seed breeding can be done by selecting sources based on shape, smell, and color. Seed selection based on visual conditions can be done if the number is limited. However, if the number of seeds increases, identifying them will be increasingly difficult.

In this study, classification was carried out based on the shape of the rice seed. Similar research has been carried out, as shown in Table 1.

Table 1. Previous research on rice seed classification

author	**year**	**data**	**class**	**feature**	**classification method**	**accuracy**
Guzman et al.	2008	52	5	morphology	NN	70
Silva & Sonnadara	2013	450	9	Textures Morphology color	NN	92
Cinar & Koklu	2019	3810	2	morphology	LR.MLP,SVM,DT,RF, NB,kNN	LR 93.02
Kiratinaranapuk et al.	2020	50.000	14	Shape, color, texture	LR, LDA, kNN, SVM VGG16, VGG19, Xception, Inception InceptionResNetv2	SVM 90.61 InceptionResNetV 2 95.15
Koklu et al.	2021	75.000	5	-	ANN, DNN, CNN VGG16	99.87, 99.5, 100
Tonael et al.	2021	140	2	GLCM	kNN, SVM	SVM 92.85
Wu et al.	2021	750	3	-	VGG16	95
Qadri et al.	2021	10800	6	Textures, color	Logistic Model T Tree	97.4
Jin et al.	2022	1900	10		SVM, LR, RF Lenet, Googlenet, Resnet	ResNet 86.08

Research on corn seed classification has been carried out using machine learning and deep learning methods. Guzman uses machine learning (ML) approaches, including classifying rice seeds using morphological features with 52 images and five classes—classification using Neural networks. Test results show 70% accuracy [2]. Next, Silva and Sonnadara classified nine classes with 450 images. The research used texture, morphology, and color feature extraction—neural network classification. Experimental results show an accuracy of up to 92% [3]. Research by Cinar & Koklu they were using ML methods, namely Logistic Regression (LR), Multilayer perceptron (MLP), Support Vector Machine (SVM), Decision Tree (DT, Random Forest (RF), Naïve Bayes (NB), and k-Nearest Neighbor (kNN). Testing using 3,810 images divided into two classes of rice seeds. Feature extraction using morphology. Test results show the best Linear Regression accuracy at 93.12% [4]. The following research by Tonael et al. used 140 image data with two classes. Extraction of Gray Level Co-occurrence Matrix (GLCM) features and classification using comparison of kNN and SVM. The best test results on SVM have an accuracy of 92.85% [5]. Furthermore, Qadri et al. used 6 class data with 10,800 images. Extraction features using texture and color—classification using a Logistic Model Tree. Test results show 97.4% accuracy [6]. Kiratinaranapuk et al. used LR, LDA, kNN, and SVM machine learning with shape, color, and texture features—experimental data 50,000 with 14 classes. Classification results show that SVM has an accuracy of up to 90.61% [4]. Following research, Jin et al. used SVM, LR, and RF machine learning methods [7].

Research using machine learning classification requires initializing the features that must be used for the classification process. The system can conduct the classification mining process with features that must be defined first. Apart from machine learning, there is also deep learning, which was used in previous research, including Kiratinaranapuk et al. using VGG16, VGG19 [8], Xception [9], Inception [10], InceptionResnetV2. The best accuracy results are up to 95.15% using InceptionResnetV2 [4]. Furthermore, research conducted by Koklu et al. used five classes with 75,000 data. Classification uses NN, Deep Neural Network, and Convolutional Neural Network VGG16. Classification results show accuracy of up to 99.87%, 99.5%, and 100%, respectively [11]. Research by Jin et al. also uses deep learning classification methods such as LeNet [7], GoogleNet [12], and ResNet [13]. Experimental results show that ResNet produces accuracy values of up to 86.08% [7].

Research using deep learning does not require initializing feature types. The model works in a black-box manner that performs input and output without having to understand the type of feature used. Generally, deep learning for classification consists of the first layer for feature extraction and the second for classification. In this research, we classified corn seeds using the deep learning method.

2. RESEARCH METHOD

This section explains the research stages that must be carried out to obtain rice seed classification results. Figure 1 shows the proposed system.

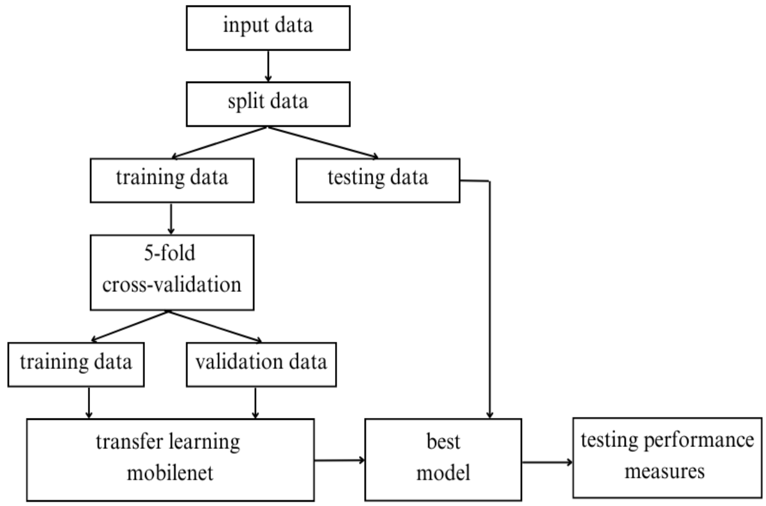


Figure 1. Proposed system

2.1 Dataset

The dataset uses data from previous research from Moklu et al., which has five rice seeds: arborio, basmati, jasmine, karacadag, and ipsala [11]. The amount of data used is 2,000 per class. So, the total data is 10,000 images. Next, split the data with a ratio of 80:20 for training vs. testing. The training data uses 5-fold cross-validation, which divides the data into training and validation data. So, the number of training is 6,400, validation is 1,600, and testing is 2,000 images.

2.2 MobileNet

MobileNet is a lightweight version of CNN from Google. The characteristic of mobileNet is that it has Deptwise Separable Convolution, which functions to reduce model size and complexity. The size model has smaller parameters. Meanwhile, complexity has fewer multiplications and additions (multi-adds) [14]–[16].

Depthwise Separable Convolution is depthwise convolution plus pointwise convolution:

1. Depthwise Convolution is a channel-wise kernel×kernel spatial convolution. If the data has five channels, it will have five kernel×kernel spatial convolutions.
2. Meanwhile, pointwise convolution is a 1×1 convolution to change dimensions.

The mobileNet architecture can be seen in Table 2. Meanwhile, Figure 2 shows the layers after convolution and depthwise separable.

2.3 Transfer learning

Transfer learning utilizes feature representations from pre-trained models without training a new model from scratch. Pre-trained models are usually trained on large datasets that are benchmarks for computer vision. The weights from the model can be reused in other computer vision tasks. Pre-trained models can perform new functions for image classification or integrate the training process on new models. Pre-trained models save training time and lower generalization errors. Transfer learning is suitable when used on small training data. The weights from the pre-trained models are used to initialize the new model [17]–[19].

Table 2. MobileNet Architecture

Layer /stride	**Filter Shape**	**Input Size**
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
conv1 (Conv2D)	(None, 112, 112, 32)	864
conv_dw_1 (DepthwiseConv2D)	(None, 112, 112, 32)	288
conv_pw_1 (Conv2D)	(None, 112, 112, 64)	2048
conv_pad_2(ZeroPadding2D)	(None, 113, 113, 64)	0
conv_dw_2 (DepthwiseConv2D)	(None, 56, 56, 64)	576
conv_pw_2 (Conv2D)	(None, 56, 56, 128)	8192
conv_dw_3 (DepthwiseConv2D)	(None, 56, 56, 128)	1152
conv_pw_3 (Conv2D)	(None, 56, 56, 128)	16384
conv_pad_4(ZeroPadding2D)	(None, 57, 57, 128)	0
conv_dw_4 (DepthwiseConv2D)	(None, 28, 28, 128)	1152
conv_pw_4 (Conv2D)	(None, 28, 28, 128)	32768
conv_dw_5 (DepthwiseConv2D)	(None, 28, 28, 256)	2304
conv_pw_5 (Conv2D)	(None, 28, 28, 256)	65636
…	…	…
conv_dw_13 (DepthwiseConv2D)	(None, 7, 7, 1024)	9216
conv_pw_13 (Conv2D)	(None, 7, 7, 1024)	1048576
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1024)	0
dense (Dense)	(None, 1024)	1049600
dense_1 (Dense)	(None, 5)	5125
Total params: 4,283,589 (16.34 MB) Trainable params: 1,054,725 (4.02 MB) Non-trainable params: 3,228,864 (12.32 MB)		

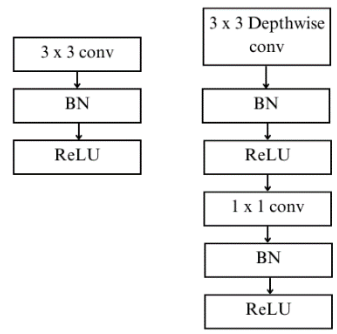


Figure 2. Batch Normalization and Rectified Linear Unit after each convolutional layer

Transfer learning is usually followed by fine-tuning, which functions to improve model performance. If you retrain the entire model, overfitting will likely occur. This solution can be provided by retraining the model using a low learning rate. Fine-tuning is usually done on the bottom layer of the model. However, if the model fails to identify it, it can fine-tune the low-level features of the convolutional layer [20]–[22]. The stages of transfer learning are shown in Figure 3.

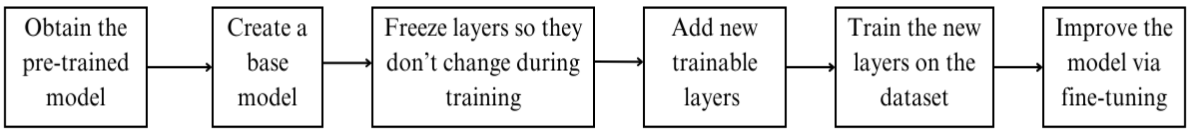


Figure 3. Transfer Learning steps [23]

The stages of transfer learning are as follows:

1. Obtain pre-trained models. The transfer learning process begins by getting a model that has undergone previous training. Pre-trained models generally take imageNet weights.
2. Create a base model. This section uses a base model, for example, mobileNet. The base model is only used in the final output layer for the new role. Therefore, remove the old final output layer. Then, add a final output layer that is compatible with the problem.
3. Freeze layers so they don't change during training. Freeze the low feature layer in the initial convolutional layers. The weights do not need to be re-initiated. Learning has been carried out previously on the pre-trained model used.
4. Add new trainable layers. The next step is adding a new trainable layer, bringing the old features to the new dataset. The new trainable layer is essential because the model is pre-trained without a final output layer.
5. Train the new layers on the dataset. The pre-trained model is different from the classification model that will be used. Generally, a pre-trained model such as ImageNet has 1,000 outputs, while the classification model that will be used consists of five classes. So, the model must be trained with a new output layer. Therefore, usually, a new dense layer is added. The new dense layer units correspond to the number of output classes.
6. Improve the model via fine-tuning. Fine-tuning performs unfreezing on certain parts of the base model. Usually, the convolutional layer produces high features in the final layers with a shallow learning rate. A low learning rate aims to improve performance and prevent overfitting.

Figure 4 shows the transfer learning process and fine-tuning on certain parts of the mobileNet.

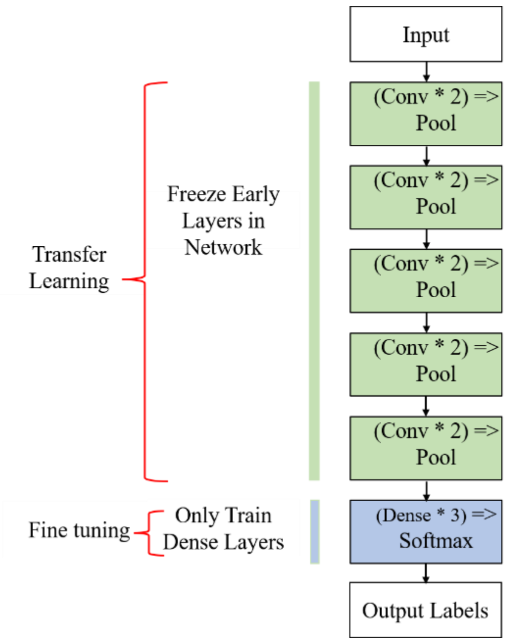


Figure 4. Transfer learning and fine-tuning

In this research, transfer learning was carried out by freezing the entire depthwise separable convolutional layer. Meanwhile, fine-tuning carries out training on classified rice seed data. The classification aims to identify images in five groups of rice types.

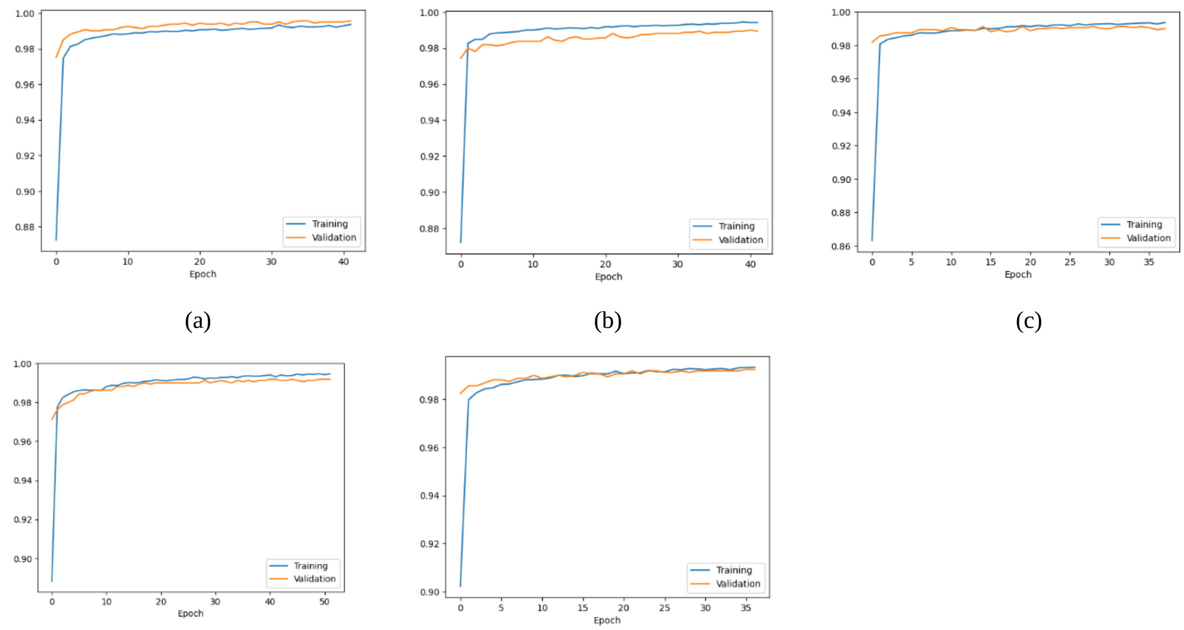
3. MAIN RESULTS

3.1 Testing environment

Windows 11 Pro 64-bit Operating System, Processor Intel(R) Core (TM) i5-10210U CPU @ 1.60GHz (8 CPUs), ~2.1GHz, Python Programming Language with Tensorflow library, Keras, Numpy, OpenCV, Scikitlearn, Matplotlib, and Pandas implemented using Google Colab Pro. Additionally, initialize the hyperparameter values as max-epoch = 100, batch-size = 10, learning rate = 0.0003, and moment = 0.9.

3.2 Result

The experiment began with a training and validation process with 5-fold cross-validation [24]–[26]. The accuracy results of the training and validation process are shown in Figure 5.



(a) (b) (c)

(d) (e)

Figure 5. (a) – (e) Results of training accuracy and data validation in 5-fold cross-validation (fold-1 until fold-5)

Figure 5 shows the success of the training and validation process. The validation results show that there is no overfitting in the data. The experimental results show that fold-1 leads to the best validation accuracy results. Apart from accuracy, performance is measured using precision, recall, and f1-score. Figure 6 shows the performance of each measure.

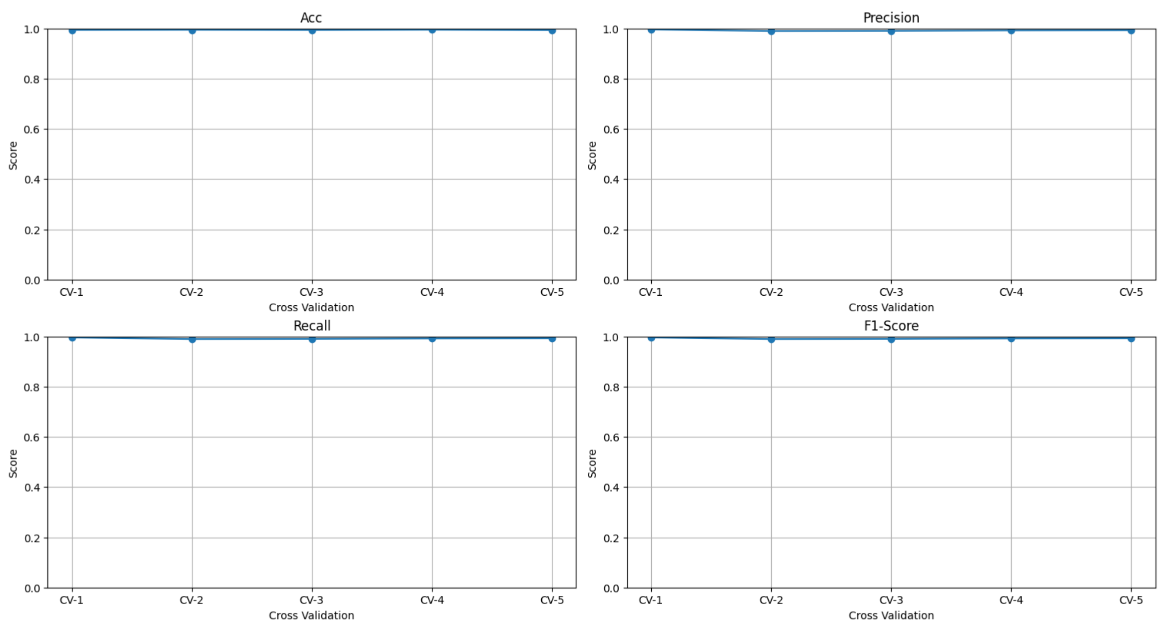


Figure 6. Performance Measures of training and validation result

Figure 6 shows the performance measures (accuracy, precision, recall, and f1-score) of 5-fold cross-validation. The average precision, recall, and f1-score performance is almost 100%. Furthermore, the measurement performance using cross-validation is shown in Tables 3a to 3e to make it more transparent.

Table 3a. Performance measures of validation data fold-1

	precision	**recall**	**f1-score**	**support**
arborio	1.0	0.99	1.0	320
basmati	0.99	1.0	0.99	320
ipsala	1.0	1.0	1.0	320
jasmine	1.0	0.99	0.99	320
karacadag	0.99	1.0	1.0	320
accuracy			1.0	1600
average	1.0	1.0	1.0	1600

Table 3b. Performance measures of validation data fold-2

	precision	**recall**	**f1-score**	**supp**
arborio	0.98	1.0	0.99	320
basmati	0.98	0.99	0.99	320
ipsala	1.0	1.0	1.0	320
jasmine	1.0	0.99	0.99	320
karacadag	0.99	0.97	0.98	320
accuracy			0.99	1600
average	0.99	0.99	0.99	1600

Table 3c. Performance measures of validation data fold-3

	precision	**recall**	**f1-score**	**support**
arborio	0.99	0.98	0.99	320
basmati	0.98	0.99	0.99	320
ipsala	1.0	1.0	1.0	320
jasmine	1.0	0.99	0.99	320
karacadag	0.98	0.99	0.98	320
accuracy			0.99	1600
average	0.99	0.99	0.99	1600

Table 3d. Performance measures of validation data fold-4

	precision	**recall**	**f1-score**	**supp**
arborio	0.98	0.99	0.99	320
basmati	0.99	0.99	0.99	320
ipsala	1.0	1.0	1.0	320
jasmine	0.99	1.0	1.0	320
karacadag	0.99	0.97	0.98	320
accuracy			0.99	1600
average	0.99	0.99	0.99	1600

Table 3e. Performance measures of validation data fold-5

	precision	**recall**	**f1-score**	**support**
arborio	0.99	0.99	0.99	320
basmati	0.98	1.0	0.99	320
ipsala	1.0	1.0	1.0	320
jasmine	1.0	0.99	1.0	320
karacadag	0.98	0.98	0.98	320
accuracy			0.99	1600
average	0.99	0.99	0.99	1600

The results of measuring the performance of the five folds show that fold-1 produces the best performance regarding the accuracy of each fold, as shown in Figure 7.

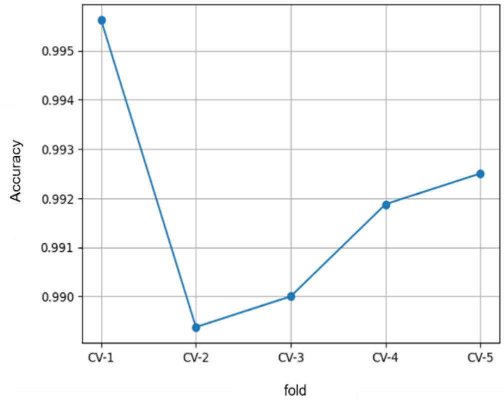


Figure 7. Accuracy of each fold cross-validation (CV)

For training, because the system uses Colab Pro, it can learn relatively quickly, around two to three minutes, as shown in Table 4. The fastest training time for fold-5 is 2 minutes 43 sec. Meanwhile, the longest training time on fold-4 is 3 minutes and 39 seconds.

Table 4. Time of training data

fold	Training Time
1	0:03:20.426846
2	0:03:03.998708
3	0:02:44.535499
4	0:03:39.649333
5	0:02:43.564186

The next stage is testing data 20%. Testing uses the best model obtained in fold-1 during the training and validation process. The test results are shown in Figure 8, which is the confusion matrix of the testing results.

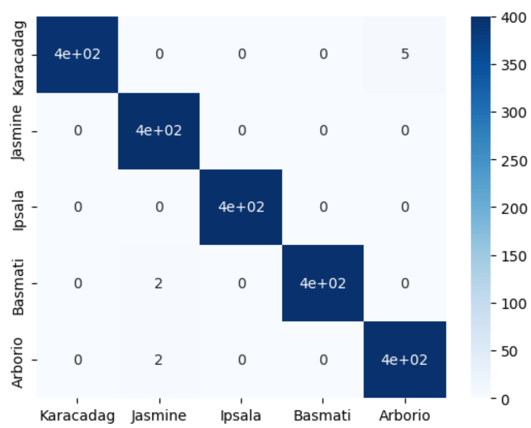


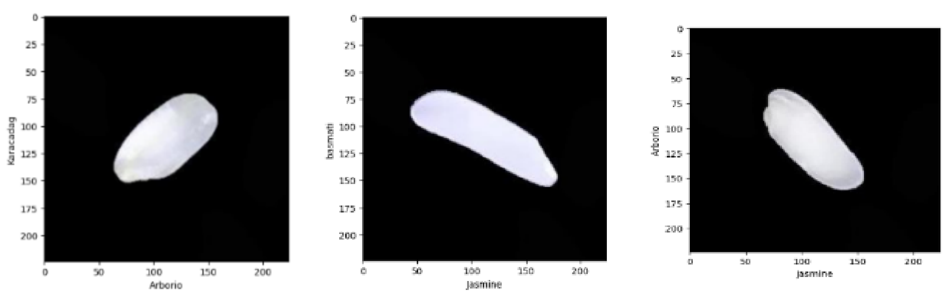
Figure 8. Confusion matrix of testing data

The test results showed only nine identification errors out of 2,000 testing images. The misidentified image is found in Arborio, Basmati, and Karacadag seed types. Those identified incorrectly were in the Basmati and Arborio classes. From Figure 8, precision Recall, f1-score can be calculated as shown in Table 4. Visualization of the test results is shown in Figure 9. The x-axis is the predicted label, while the y-axis is the actual label. Figure 9 shows the results of random testing.

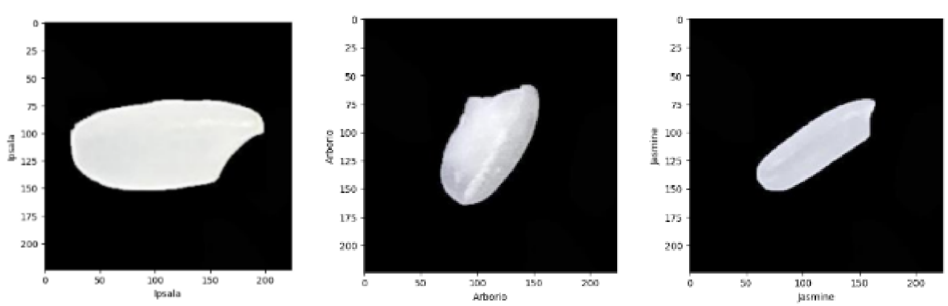
Table 4. Testing Result of Rice Seeds Classification

Seed type	**Precision**	**Recall**	**F1-score**	**Accuracy**
Karacadag	0.9875	1	0.993710692	$= \frac{1991}{2000}$
Jasmine	1	0.990099	0.995024876	
Ipsala	1	1	1	$= 0.9955$
Basmati	0.995	1	0.997493734	
Arborio	0.995	0.96368	0.979089791	
Average	**0.9955**	**0.990756**	**0.993063819**	

The results of performance measurement testing data are accuracy 99.55%, precision 99.55%, recall 99.08%, f1-score 99.31%



(a)



(b)

Figure 9. (a) false identification; (b) valid identification

4. CONCLUSION

In this research, we have classified rice seeds. Due to system limitations, the experimental data used was only 10,000 images divided into five classes. The use of transfer learning is beneficial for implementations with limited data. Transfer learning (TL) uses a model based on imageNet. The TL process freezes all depthwise separable convolutional layers. Fine-tuning carries out training on a dataset for image classification of 5 types of rice seeds.

The dataset is divided into training, validation, and testing. Training and testing data have a ratio of 80:20. Meanwhile, training and validation use 5-fold cross-validation. The training and validation results show the best model at fold-1. Meanwhile, the test showed performance of more than 99%.

ACKNOWLEDGMENT

This research was funded by the Independent Research Scheme of International Collaborative Research at Trunojoyo University, Madura, in 2023.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] G.R. Sadimantara, W. Nuraida, N.W.S. Suliartini, et al. Evaluation of some new plant type of upland rice (Oryza sativa L.) lines derived from cross breeding for the growth and yield characteristics, IOP Conf. Ser.: Earth Environ. Sci. 157 (2018), 012048. https://doi.org/10.1088/1755-1315/157/1/012048.

[2] J.D. Guzman, E.K. Peralta, Classification of Philippine rice grains using machine vision and artificial neural networks, in: World conference on agricultural information and IT, Tokyo, Japan, 24-27 August 2008, pp 41-48, (2008).

[3] C.S. Silva, U. Sonnadara, Classification of rice grains using neural networks, Proc. Techn. Sessions, 29 (2013), 9-13.

[4] K. Kiratiratanapruk, P. Temniranrat, W. Sinthupinyo, et al. Development of paddy rice seed classification process using machine learning techniques for automatic grading machine, J. Sensors. 2020 (2020), 7041310. https://doi.org/10.1155/2020/7041310.

[5] D.A. Tonael, Y.R. Kaesmetan, M.I.J. Lamabelawa, Selection of superior rice seed features using deep learning

method, Indones. J. Artif. Intell. Data Min. 4 (2021), 78-88.

[6] S. Qadri, T. Aslam, S.A. Nawaz, et al. Machine vision approach for classification of rice varieties using texture features, Int. J. Food Prop. 24 (2021), 1615-1630. https://doi.org/10.1080/10942912.2021.1986523.

[7] B. Jin, C. Zhang, L. Jia, et al. Identification of rice seed varieties based on near-infrared hyperspectral imaging technology combined with deep learning, ACS Omega. 7 (2022), 4735-4749. https://doi.org/10.1021/acsomega.1c04102.

[8] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, pp. 1-14, (2015).

[9] C. Francois, Xception: deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1251-1258, (2017).

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, et al. Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016: pp. 2818-2826. https://doi.org/10.1109/CVPR.2016.308.

[11] M. Koklu, I. Cinar, Y.S. Taspinar, Classification of rice varieties with deep learning methods, Computers Electron. Agric. 187 (2021), 106285. https://doi.org/10.1016/j.compag.2021.106285.

[12] C. Szegedy, W. Liu, Y. Jia, et al. Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Boston, MA, USA, 2015: pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594.

[13] H. Mujtaba, Introduction to ResNet or residual network, 2020. https://www.mygreatlearning.com/blog/resnet.

[14] W. Sae-Lim, W. Wettayaprasit, P. Aiyarak, Convolutional neural networks using MobileNet for skin lesion classification, in: 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), IEEE, Chonburi, Thailand, 2019: pp. 242–247. https://doi.org/10.1109/JCSSE.2019.8864155.

[15] L.G. Falconi, M. Perez, W.G. Aguilar, Transfer learning in breast mammogram abnormalities classification with mobilenet and nasnet, in: 2019 International Conference on Systems, Signals and Image Processing (IWSSIP), IEEE, Osijek, Croatia, 2019: pp. 109–114. https://doi.org/10.1109/IWSSIP.2019.8787295.

[16] A.G. Howard, M. Zhu, B. Chen, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications, preprint, (2017). http://arxiv.org/abs/1704.04861.

[17] S. Kornblith, J. Shlens, Q. V. Le, Do better ImageNet models transfer better? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2661-2671, (2019).

[18] R. Marlow, S. Kuriyakose, N. Mesaros, H.H. Han, R. Tomlinson, S.N. Faust, M.D. Snape, A.J. Pollard, A. Finn, A phase III, open-label, randomised multicentre study to evaluate the immunogenicity and safety of a booster dose of two different reduced antigen diphtheria-tetanus-acellular pertussis-polio vaccines, when co-

administered with measles-mumps-rubella vaccine in 3 and 4-year-old healthy children in the UK, Vaccine. 36 (2018), 2300-2306. https://doi.org/10.1016/j.vaccine.2018.03.021.

[19] M. Huh, P. Agrawal, A.A. Efros, What makes ImageNet good for transfer learning?, (2016). http://arxiv.org/abs/1608.08614.

[20] A.K. Reyes, J.C. Caicedo, J.E. Camargo, Fine-tuning deep convolutional networks for plant recognition, in: Working Notes of Conference and Labs of the Evaluation forum (CLEF), (2015).

[21] E.C. Too, L. Yujian, S. Njuki, et al. A comparative study of fine-tuning deep learning models for plant disease identification, Computers Electron. Agric. 161 (2019), 272-279. https://doi.org/10.1016/j.compag.2018.03.032.

[22] P. Peng, J. Wang, How to fine-tune deep neural networks in few-shot learning?, preprint, (2020). http://arxiv.org/abs/2012.00204.

[23] D. Mwiti, Transfer learning guide: a practical tutorial with examples for images and text in Keras, (2023). https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras.

[24] T.T. Wong, P.Y. Yeh, Reliable accuracy estimates from k -fold cross validation, IEEE Trans. Knowl. Data Eng. 32 (2020), 1586-1594. https://doi.org/10.1109/TKDE.2019.2912815.

[25] S. Yadav, S. Shukla, Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification, in: 2016 IEEE 6th International Conference on Advanced Computing (IACC), IEEE, Bhimavaram, India, 2016: pp. 78–83. https://doi.org/10.1109/IACC.2016.25.

[26] J.G. Moreno-Torres, J.A. Saez, F. Herrera, Study on the impact of partition-induced dataset shift on k-fold cross-validation, IEEE Trans. Neural Netw. Learn. Syst. 23 (2012), 1304-1312. https://doi.org/10.1109/TNNLS.2012.2199516.