ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20240910214215

Original scientific paper
Received: 2024-09-10, Accepted: 2025-02-21

Detection and Classification of Growth Stages in Rice Using Artificial Neural Networks

Durodola Folasade*, Owoeye Samuel, Kamil-Bello Furqan, Daodu Sakira, Makinde Kayode, Folaranmi Olaniyi

Abstract: This project focused on the development of a machine learning model to classify rice plants based on their growth stages, specifically identifying whether the plants are in the "Raw" or "Ripe" stage. The research was conducted using a dataset obtained from Roboflow, which provided annotated images of rice plants. The dataset was divided into training, testing, and validation subsets to ensure the model's robustness and generalization capability. The project involved a comprehensive data preparation process, which included consolidating images into class-based folders, handling file conflicts, and stratifying the dataset into appropriate splits for training, testing, and validation. Several convolutional neural networks (CNN) architectures were explored, including ResNet50, InceptionV3, and MobileNetV2, each leveraging transfer learning from pre-trained models on the ImageNet dataset. ResNet50 achieved an accuracy of 87.3% with a log loss of 0.33, demonstrating good performance but with some misclassifications between similar classes. InceptionV3 outperformed the other models, achieving an accuracy of 95.1% and a log loss of 0.13, indicating superior classification capability and better calibration of predicted probabilities. MobileNetV2 also performed well with an accuracy of 93.5% and a log loss of 0.22, offering a balance between accuracy and computational efficiency. The results highlight InceptionV3 as the most effective model for this task, with a strong ability to differentiate between the rice growth stages. The findings underscore the importance of model selection and data preparation in developing accurate and reliable machine-learning models for agricultural applications. The project demonstrates the potential of CNNs in improving agricultural practices through precise crop monitoring and classification.

Keywords: convolutional neural networks (CNNs); image classification; InceptionV3; machine learning; MobileNetV2; ResNet50; rice plant

1 INTRODUCTION

Rice is a crucial food source worldwide, particularly in regions such as Asia, Africa, and Latin America, serving as a primary dietary staple for billions of people [1]. With the growing global population and changing climatic conditions, there is increasing pressure on rice production systems to enhance production [2]. Traditional methods of monitoring rice growth stages are labour-intensive, time-consuming, and prone to human error, making them inadequate for meeting the rising demands and ensuring food security [3].

Artificial Intelligence (AI), specifically Artificial Neural Networks (ANNs), has shown significant potential in transforming agriculture. ANNs can process large datasets and detect complex patterns, making them ideal for precision agriculture applications [4]. By employing ANNs, we can develop automated systems capable of accurately detecting and monitoring the growth stages of rice, leading to improved crop management, optimized resource use, and increased yields [5, 6]. This innovation contributes to global food security and promotes sustainable agricultural practices.

In addition to boosting productivity, applying ANNs in rice growth stage detection can significantly reduce the environmental impact of rice farming. Precise identification of growth stages allows for the optimized use of water, fertilizers, and pesticides, reducing wastage and environmental degradation [7]. This precision also minimizes the runoff of harmful chemicals into water bodies, protecting aquatic ecosystems and promoting eco-friendly agricultural practices.

Furthermore, ANNs offer valuable tools for adapting to climate variability and change. As climate change presents unpredictable challenges to agriculture, real-time monitoring of rice growth stages becomes essential. ANNs can help predict and mitigate the effects of adverse weather conditions by providing timely insights, enabling farmers to take proactive measures [8]. This adaptability is crucial for maintaining rice productivity and ensuring food security in the face of changing climatic patterns.

The economic benefits of deploying ANN-based systems in rice agriculture are also noteworthy. Automation of growth stage monitoring can lead to significant cost savings by reducing the need for extensive manual labour and enabling more precise agricultural interventions. For smallholder farmers, who play a significant role in global rice production, these savings can be crucial, allowing them to reinvest in their operations and improve their livelihoods [9].

Despite these advantages, traditional methods of monitoring rice growth stages remain laborious, time-consuming, and prone to errors. These methods do not provide real-time data, which is essential for making timely agricultural interventions. Inaccurate and inconsistent monitoring can lead to suboptimal resource use, affecting crop health and yield [5]. There is a pressing need for an automated, reliable, and efficient system that can accurately detect and monitor rice growth stages, handle variability in growth patterns, and provide farmers with accurate, real-time information to enhance decision-making and improve overall crop management practices.

The integration of ANNs into rice farming offers a comprehensive solution to agronomic, environmental, and economic challenges in rice production. This study aims to utilize artificial neural networks (ANNs) for the prediction of rice growth stages through the examination of phenological data and image-based inputs, specifically targeting the vegetative, reproductive, and ripening phases. The ultimate goal is to equip farmers with practical insights that facilitate improved crop management, optimized resource use, and enhanced productivity.

2 LITERATURE REVIEW

The use of technological advancements and methodologies in utilizing artificial neural networks (ANNs) for detecting rice growth stages can not be over-emphasised.

It highlights the shift from traditional monitoring methods to modern precision agriculture techniques, emphasizing the role of ANNs in enhancing agricultural practices [10].

Precision agriculture represents a significant advancement in farming, using technologies like GPS, remote sensing, and data analytics to monitor and manage crop growth with high precision. These tools enable targeted interventions, such as precise irrigation and fertilization, reducing waste, and promoting sustainable farming practices. Machine learning and AI further enhance these capabilities, offering predictive models and decision support systems for efficient farming [11-13].

ANNs have become a powerful tool in agriculture, particularly for analyzing large datasets and identifying complex patterns. In rice farming, ANNs can accurately detect and classify growth stages by processing vast amounts of data from various sources. This capability is crucial for developing automated systems that provide real-time information, reducing reliance on manual labour and increasing productivity [14, 15].

Convolutional Neural Networks (CNNs), a subset of ANNs, are particularly effective for image recognition tasks, making them ideal for analyzing images of rice at different growth stages. CNNs automatically detect and learn features from images, enabling precise classification and monitoring of rice growth. Training CNNs with diverse image datasets ensures the model's robustness and reliability in real-world applications [16, 17].

The success of an ANN-based system depends on the quality and diversity of the training data. Effective data collection involves gathering images and sensor data from multiple sources, ensuring a comprehensive view of the rice fields. Preprocessing techniques, such as data cleaning and augmentation, prepare the data for training, improving the model's robustness and accuracy [18, 19].

Training an ANN model involves feeding preprocessed data into the network and adjusting model parameters to minimize prediction errors. Selecting an appropriate neural network architecture is crucial for capturing the complex features in rice crop images. Advanced machine learning techniques are used to enhance performance, and the model's accuracy is validated using separate data to ensure reliability in real-world applications [20, 21].

3 METHODOLOGY

Accurately identifying rice growth stages is essential for effective agricultural management, as it directly impacts crop yield, resource utilization, and farming efficiency [6]. Traditionally, determining these stages has relied on manual observation and expert judgment, which can lead to variability and inefficiencies. However, advancements in artificial intelligence and machine learning present an opportunity for a more precise, automated approach to growth stage detection [21]. This chapter outlines the methodology for designing, developing, and implementing an artificial neural network (ANN)-based system to detect rice growth stages. The system aims to provide farmers with a reliable tool to automate the identification process, thereby enabling timely and informed decision-making. By integrating this system into a user-friendly platform, farmers can upload images of their rice crops and receive instant feedback on the current growth stage, optimizing their farming practices and improving productivity.

The proposed rice growth stage detection system uses artificial neural networks to analyze and classify images of rice crops, determining the specific growth stage based on visual features [14]. A convolutional neural network (CNN), tailored for image recognition tasks, serves as the system's core component [22]. The CNN is trained on a diverse dataset of rice crop images at various growth stages, captured under different environmental conditions [23]. The system processes these images to extract crucial features like texture, colour, and shape, which are essential for distinguishing between different growth [24]. The system is integrated into a web-based platform that allows farmers to upload images of their crops. Once uploaded, the CNN processes the image and provides immediate feedback on the rice's current growth stage, along with suggestions for appropriate farming practices based on the identified stage. This real-time, automated analysis significantly reduces the need for manual monitoring and expert intervention [25]. The machine learning workflow is shown in Fig. 1.

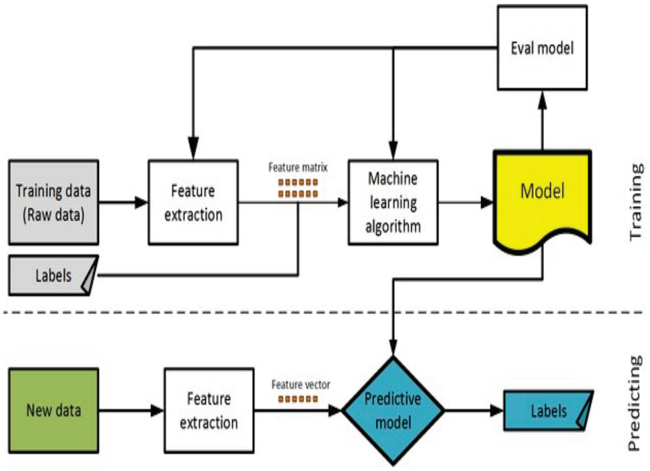


Figure 1 Machine learning workflow

The system's effectiveness hinges on the quality of the dataset used to train the machine-learning models [16]. A comprehensive dataset of rice crop images at various growth stages, captured under diverse environmental conditions, was compiled from multiple sources. To enhance the model's robustness, extensive data augmentation techniques, including random rotation, scaling, and colour adjustments, were employed [26]. Three prominent CNN architectures—ResNet, MobileNet, and InceptionNet—were selected for comparison, each offering distinct advantages in accuracy, computational efficiency, and scalability [27-29]. The training process involved hyperparameter tuning, transfer learning, and regularization techniques to ensure optimal performance. The models were continuously evaluated using metrics such as accuracy, precision, recall, and F1-score to select the best-performing architecture for deployment.

Data collection involved gathering a diverse set of rice images at different growth stages and under varying environmental conditions. The images were sourced from publicly available agricultural datasets, custom-captured images from rice fields, and contributions from agricultural research institutions. This diverse dataset enhances the model's ability to generalize and perform well in real-world applications [30]. Preprocessing steps included image resizing, noise reduction, and data augmentation to ensure compatibility with the neural network architectures [31] [32]. Exploratory Data Analysis (EDA) was conducted to understand the dataset's underlying patterns, relationships, and potential issues [33]. EDA helped identify any imbalances in the dataset and informed decisions on model selection and training strategies.

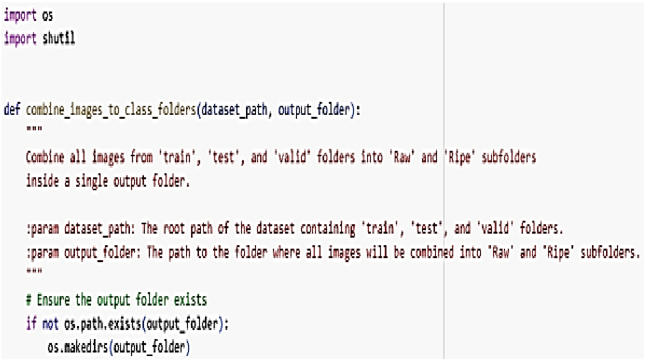
The dataset was split into training, validation, and test sets, with 70% of the data used for training, 20% for validation, and 10% reserved for testing. Normalization was applied to standardize the input values, ensuring that all input features contributed equally to the model's learning process [34]. The test site for system deployment was carefully selected to represent typical rice-growing environments, considering factors such as rice variety, environmental conditions, and field layout [35-37]. The system, including cameras, sensors, and processing units, was installed and configured according to the experimental requirements, with the selected machine-learning models evaluated on a validation set to ensure their accuracy and reliability in classifying rice growth stages.

The system's performance was evaluated through controlled experiments focusing on key metrics such as classification accuracy, model comparison, environmental adaptability, and operational efficiency. The performance of ResNet, MobileNet, and InceptionNet models was compared to determine which provided the best balance of accuracy and computational efficiency. The system's adaptability to varying environmental conditions, including different lighting and weather patterns, was assessed, along with its operational efficiency, including power consumption and maintenance requirements [23]. User feedback was collected from farmers and agricultural experts to evaluate the system's usability, reliability, and effectiveness in supporting rice growth stage monitoring. The experimental trials were conducted over an entire growing season, with continuous monitoring and data logging to capture the system's performance across different growth stages and environmental conditions.

The data collected during the experimental trials was analyzed to identify areas for improvement and optimize the system's performance. The analysis included evaluating the effectiveness of each machine learning model, the accuracy of growth stage classification, and the overall system's reliability and adaptability. The data preprocessing involved cleaning, organizing, and formatting the raw data to facilitate thorough analysis. Trends and correlations between growth stages, environmental conditions, and system performance metrics were identified, leading to model refinement and operational modifications to enhance system performance. The continuous refinement and optimization of the rice growth stage detection system aimed to ensure its effectiveness and reliability in real-world agricultural settings, ultimately providing farmers with a valuable tool to improve crop management practices.

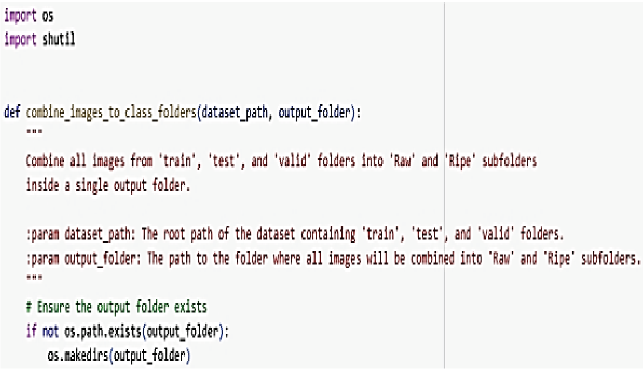
4 IMPLEMENTATION AND RESULT

The implementation of this research involves developing a machine-learning model by following a systematic workflow, beginning with data collection and preparation. A dataset from Roboflow, containing images of rice plants labeled as "Raw" or "Ripe," was used. This data was split into three subsets—training, testing, and validation—to ensure the model's generalization. The dataset preparation, shown in Fig. 2, included consolidating all images into a unified directory structure, separating them into "Raw" and "Ripe" categories for easier processing during the subsequent stages of model training.



```
import os
import shutil



def combine_images_to_class_folders(dataset_path, output_folder):
    """
    Combine all images from 'train', 'test', and 'valid' folders into 'Raw' and 'Ripe' subfolders
    inside a single output folder.

    :param dataset_path: The root path of the dataset containing 'train', 'test', and 'valid' folders.
    :param output_folder: The path to the folder where all images will be combined into 'Raw' and 'Ripe' subfolders.
    """
    # Ensure the output folder exists
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)
```

Figure 2 Data preparation



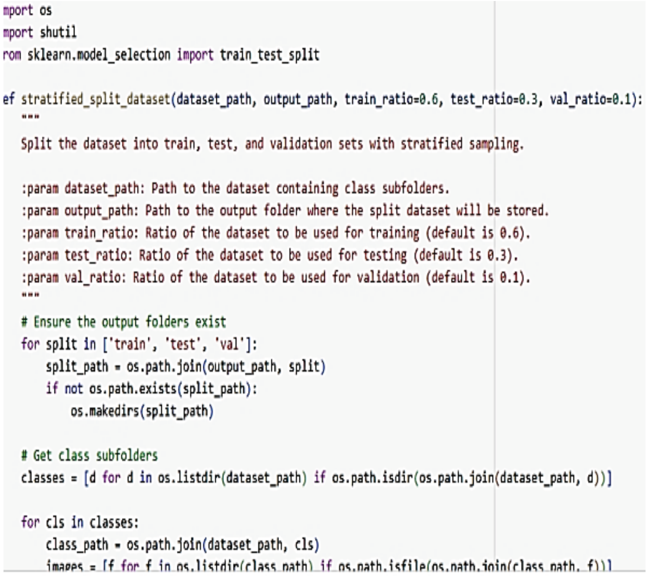
```
import os
import shutil



def combine_images_to_class_folders(dataset_path, output_folder):
    """
    Combine all images from 'train', 'test', and 'valid' folders into 'Raw' and 'Ripe' subfolders
    inside a single output folder.

    :param dataset_path: The root path of the dataset containing 'train', 'test', and 'valid' folders.
    :param output_folder: The path to the folder where all images will be combined into 'Raw' and 'Ripe' subfolders.
    """
    # Ensure the output folder exists
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)
```

Figure 3 Random sampling

Following data consolidation, the workflow handled potential file conflicts, such as duplicate file names, by appending numerical suffixes to ensure unique filenames. This process helped maintain data integrity, preventing data loss due to overwriting. Additionally, random file sampling, shown in Fig. 3 was implemented to create smaller, balanced subsets for testing or validation. Fig. 4 shows the stratified data splitting, which ensured that each class had equal representation across the training, testing, and validation sets, which is critical for building a robust model capable of generalizing well on unseen data.

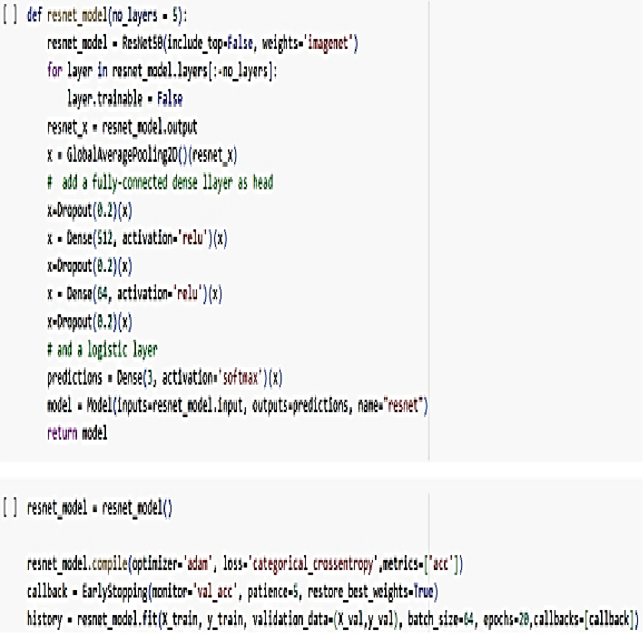
The research implemented multiple deep learning models, including ResNet50, InceptionNet, and MobileNetV2, leveraging pre-trained architectures.

ResNet50 utilized transfer learning, retaining some layers for feature extraction while fine-tuning others to adapt to the new dataset. A Global Average Pooling (GAP) layer was appended, followed by Dense layers with Dropout to avoid overfitting. Similarly, InceptionNet, based on InceptionV3, captured multi-scale features through its inception modules. The model retained valuable feature extraction capabilities while adapting to the new classification task, also incorporating GAP and Dropout layers to prevent overfitting.



```
mport os
mport shutil
rom sklearn.model_selection import train_test_split

ef stratified_split_dataset(dataset_path, output_path, train_ratio=0.6, test_ratio=0.3, val_ratio=0.1):
   """
   Split the dataset into train, test, and validation sets with stratified sampling.

   :param dataset_path: Path to the dataset containing class subfolders.
   :param output_path: Path to the output folder where the split dataset will be stored.
   :param train_ratio: Ratio of the dataset to be used for training (default is 0.6).
   :param test_ratio: Ratio of the dataset to be used for testing (default is 0.3).
   :param val_ratio: Ratio of the dataset to be used for validation (default is 0.1).
   """
   # Ensure the output folders exist
   for split in ['train', 'test', 'val']:
       split_path = os.path.join(output_path, split)
       if not os.path.exists(split_path):
           os.makedirs(split_path)

   # Get class subfolders
   classes = [d for d in os.listdir(dataset_path) if os.path.isdir(os.path.join(dataset_path, d))]

   for cls in classes:
       class_path = os.path.join(dataset_path, cls)
       images = [f for f in os.listdir(class_path) if os.path.isfile(os.path.join(class_path, f))]
```

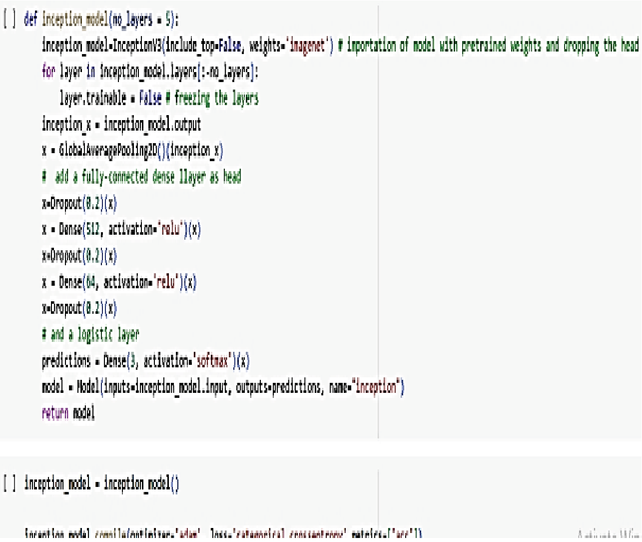
Figure 4 Data splitting



```
[ ] def resnet_model(no_layers = 5):
        resnet_model = ResNet50(include_top=False, weights='imagenet')
        for layer in resnet_model.layers[:-no_layers]:
            layer.trainable = False
        resnet_x = resnet_model.output
        x = GlobalAveragePooling2D()(resnet_x)
        #  add a fully-connected dense llayer as head
        x=Dropout(0.2)(x)
        x = Dense(512, activation='relu')(x)
        x=Dropout(0.2)(x)
        x = Dense(64, activation='relu')(x)
        x=Dropout(0.2)(x)
        # and a logistic layer
        predictions = Dense(3, activation='softmax')(x)
        model = Model(inputs=resnet_model.input, outputs=predictions, name="resnet")
        return model

[ ] resnet_model = resnet_model()


    resnet_model.compile(optimizer='adam', loss='categorical_crossentropy',metrics=['acc'])
    callback = EarlyStopping(monitor='val_acc', patience=5, restore_best_weights=True)
    history = resnet_model.fit(X_train, y_train, validation_data=(X_val,y_val), batch_size=64, epochs=20,callbacks=[callback])
```

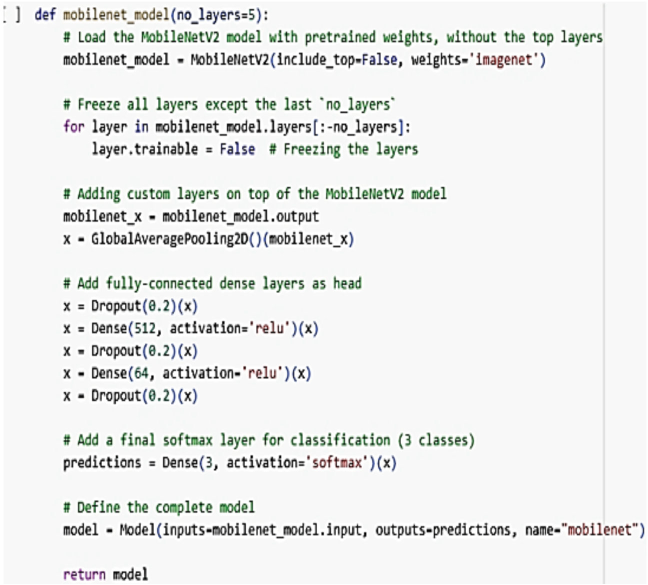
Figure 5 RestNet implementation

A comparative analysis of the performance of three models—ResNet50, InceptionV3, and MobileNetV2—on a rice growth dataset, evaluating their strengths, weaknesses, and overall effectiveness. As shown in Figs. 8 and 9 the ResNet50 model achieved an accuracy of 87.3% on the test set, with an F1 score of 86.9%, indicating a good balance between precision and recall. However, the model struggled with differentiating between certain classes, such as misclassifying instances of 'Vegetative' as 'Reproductive' and confusing 'Ripening' with 'Vegetative.' The training curves showed a smooth increase in accuracy, plateauing at 90–92%, which suggests some level of overfitting despite effective learning and generalization.



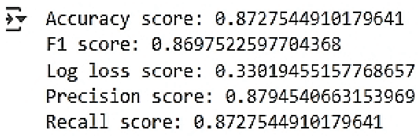
```
[ ] def inception_model(no_layers = 5):
        inception_model=InceptionV3(include_top=False, weights='imagenet') # importation of model with pretrained weights and dropping the head
        for layer in inception_model.layers[:-no_layers]:
            layer.trainable = False # freezing the layers
        inception_x = inception_model.output
        x = GlobalAveragePooling2D()(inception_x)
        #  add a fully-connected dense llayer as head
        x=Dropout(0.2)(x)
        x = Dense(512, activation='relu')(x)
        x=Dropout(0.2)(x)
        x = Dense(64, activation='relu')(x)
        x=Dropout(0.2)(x)
        # and a logistic layer
        predictions = Dense(3, activation='softmax')(x)
        model = Model(inputs=inception_model.input, outputs=predictions, name="inception")
        return model

[ ] inception_model = inception_model()


    inception_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['acc'])
```

Figure 6 Inception net implementation



```
[ ]  def mobilenet_model(no_layers=5):
         # Load the MobileNetV2 model with pretrained weights, without the top layers
         mobilenet_model = MobileNetV2(include_top=False, weights='imagenet')

         # Freeze all layers except the last `no_layers`
         for layer in mobilenet_model.layers[:-no_layers]:
             layer.trainable = False  # Freezing the layers

         # Adding custom layers on top of the MobileNetV2 model
         mobilenet_x = mobilenet_model.output
         x = GlobalAveragePooling2D()(mobilenet_x)

         # Add fully-connected dense layers as head
         x = Dropout(0.2)(x)
         x = Dense(512, activation='relu')(x)
         x = Dropout(0.2)(x)
         x = Dense(64, activation='relu')(x)
         x = Dropout(0.2)(x)

         # Add a final softmax layer for classification (3 classes)
         predictions = Dense(3, activation='softmax')(x)

         # Define the complete model
         model = Model(inputs=mobilenet_model.input, outputs=predictions, name="mobilenet")

         return model
```

Figure 7 Mobile net implementation



```
Accuracy score: 0.8727544910179641
F1 score: 0.8697522597704368
Log loss score: 0.33019455157768657
Precision score: 0.8794540663153969
Recall score: 0.8727544910179641
```

Figure 8 ResNet results breakdown

Fig. 10 shows the confusion matrix revealed minimal misclassifications, demonstrating InceptionV3's superiority in-class differentiation.

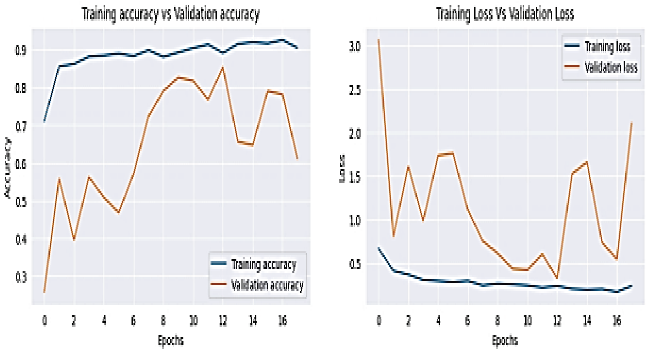
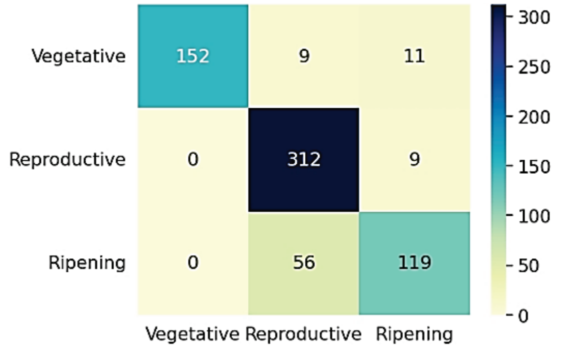


Figure 9 Accuracy and loss plot for the ResNet50



	Vegetative	Reproductive	Ripening
Vegetative	152	9	11
Reproductive	0	312	9
Ripening	0	56	119

Figure 10 Confusion matrix

Figs. 11 and 12 show that the InceptionV3 model outperformed ResNet50 significantly, achieving a test accuracy of 95.1% and an F1 score of 94.8%. Its lower log loss of 0.13 indicated well-calibrated predicted probabilities, making it a highly reliable model for this classification task. Fig. 10 showed the confusion matrix, which revealed minimal misclassifications, demonstrating InceptionV3's superiority in-class differentiation. The training curves for InceptionV3 showed rapid improvement in accuracy, which stabilized at a high level, with the loss curve reflecting excellent fitting and generalization.

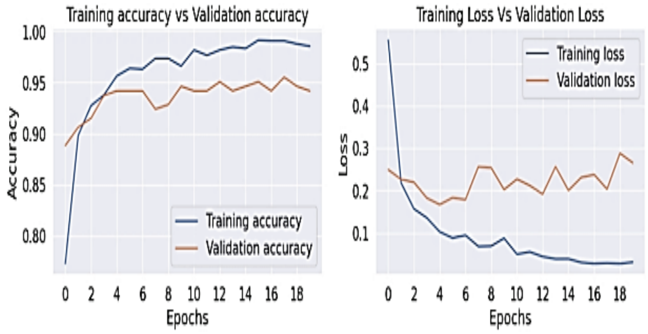
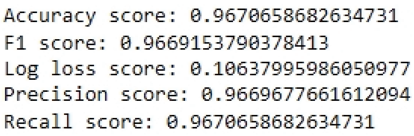


Figure 11 Accuracy and loss plot for the InceptionV3



```
Accuracy score: 0.9670658682634731
F1 score: 0.9669153790378413
Log loss score: 0.10637995986050977
Precision score: 0.9669677661612094
Recall score: 0.9670658682634731
```

Figure 12 Results break down for inception net

The MobileNetV2 model (Figs. 13 and 14) also performed well, with an accuracy of 93.5% and an F1 score of 93.1%, positioning it between InceptionV3 and ResNet50 in terms of performance. The log loss for MobileNetV2 was 0.22, better than ResNet50 but not as good as InceptionV3, indicating some room for improvement in probability calibration. While the confusion matrix highlighted some misclassifications, particularly in the 'Ripening' class, MobileNetV2 showed better stability in model performance compared to ResNet50, with steady decreases in loss and strong generalization capabilities.

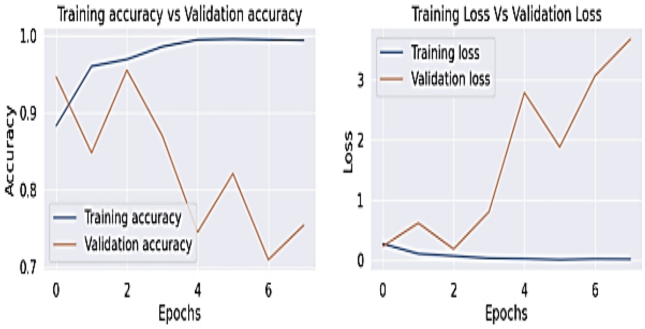
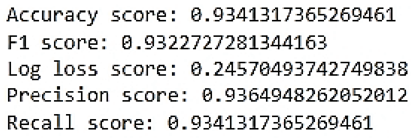


Figure 13 MobileNet accuracy and loss



```
Accuracy score: 0.9341317365269461
F1 score: 0.9322727281344163
Log loss score: 0.24570493742749838
Precision score: 0.9364948262052012
Recall score: 0.9341317365269461
```

Figure 14 MobileNet results

In comparing ResNet50, InceptionV3, and MobileNetV2, InceptionV3 emerges as the most effective model, achieving the highest accuracy (95.1%), F1 score (94.8%), and the lowest log loss (0.13), indicating superior probability calibration and reliable classification with minimal misclassifications. ResNet50 struggles with class differentiation and shows signs of overfitting, making it less reliable overall. Thus, InceptionV3 is the best model, with MobileNetV2 as a strong alternative.

MobileNetV2 was chosen for its efficiency on mobile and embedded devices, employing depthwise separable convolutions to reduce computational demands. The model was pre-trained on ImageNet, and adjustments were made to the final layers to suit the specific task of classifying rice growth stages. The model was compiled with a loss function, optimizer, and evaluation metrics tailored for multi-class classification. The training involved refining the model's predictions over multiple epochs, with validation against a separate dataset to ensure generalization and performance.

In summary, the workflow—from data preparation to model implementation—was designed to maintain data integrity, ensure balanced datasets, and build efficient models for classifying rice growth stages. Each model—ResNet50, InceptionNet, and MobileNetV2—was tailored for the task, utilizing pre-trained architectures and advanced techniques like transfer learning, GAP, and Dropout to achieve accurate and generalizable results.

5 CONCLUSION

This study analyzed and evaluated the performance of three advanced deep learning models—ResNet50, InceptionV3, and MobileNetV2—on rice plant classification. Among these, InceptionV3 proved to be the most effective, achieving the highest accuracy (95.1%) and F1 score (94.8%), along with excellent probability calibration. Its low misclassification rates make it the best model for differentiating between rice plant classes.

MobileNetV2, while slightly less powerful than InceptionV3, still performed well with an accuracy of 93.5% and an F1 score of 93.1%. It strikes a good balance between performance and efficiency, making it suitable for deployment in resource-constrained environments. Despite its higher log loss of 0.22 compared to InceptionV3, MobileNetV2 remains a reliable model.

ResNet50, although effective, lagged behind the other models with an accuracy of 87.3% and an F1 score of 86.9%. Its higher log loss of 0.33 and difficulties in class differentiation indicate that it requires significant refinement to compete with InceptionV3 and MobileNetV2.

Based on these findings, InceptionV3 is recommended as the preferred model for rice plant classification due to its superior accuracy, high F1 score, and well-calibrated probabilities, making it ideal for reliable and precise classification tasks.

To further enhance model performance, increasing the size and diversity of the dataset used for training and testing is essential. A more comprehensive dataset will help improve the models' ability to generalize, reduce misclassifications, and provide more accurate and reliable classification results.

Additionally, techniques such as transfer learning, fine-tuning of pre-trained models, and the development of hybrid models that combine the strengths of different architectures are recommended. Continuous improvement efforts, including advanced data augmentation and robust preprocessing techniques, will further advance the field of rice plant classification, enabling more effective solutions in precision agriculture. In conclusion, the research effectively highlights the transformative potential of AI in agriculture, InceptionV3 is hereby recommended as the most effective model for rice growth stage classification

6 REFERENCES

[1] Muthayya, S., Sugimoto, J. D., Montgomery, S. & Maberly, G. F. (2014). An overview of global rice production, supply, trade, and consumption. *Ann. N. Y. Acad. Sci.*, 1324, 7-14. https://doi.org/10.1111/nyas.12540

[2] Wassermann, R., Jagadish, S. V. K., Heuer, S., Ismail, A., Redona, E., Serraj, R., ... & Sumfleth, K. (2009). Climate change affecting rice production: the physiological and agronomic basis for possible adaptation strategies. *Advances in agronomy, 101*, 59-122. https://doi.org/10.1016/S0065-2113(08)00802-X

[3] Wilfred, O. R. & Consultant, L. (2006). Final survey report on the status of rice production, processing and marketing in Uganda. A report submitted to the Embassy of Japan in Uganda through JICA and Sasakawa Africa Association-Uganda, Kampala.

[4] Sharma, A., Jain, A., Gupta, P. & Chowdary, V. (2020). Machine learning applications for precision agriculture: A comprehensive review. *IEEE Access, 9*, 4843-4873. https://doi.org/10.1109/ACCESS.2020.3048415

[5] Liu, H., Wang, H., Zhou, X. & Li, Z. (2019). Crop growth stage detection using deep learning techniques for smart agriculture. *Sensors, 19*(7), 2008.

[6] Zhang, K., Ge, X., Shen, P., Li, W., Liu, X., Cao, Q., ... & Tian, Y. (2019). Predicting rice grain yield based on dynamic changes in vegetation indexes during early to mid-growth stages. *Remote sensing, 11*(4), 387. https://doi.org/10.3390/rs11040387

[7] Nabavi-Pelesaraei, A., Rafiee, S., Mohtasebi, S. S., Hosseinzadeh-Bandbafha, H. & Chau, K. W. (2018). Integration of artificial intelligence methods and life cycle assessment to predict energy output and environmental impacts of paddy production. *Science of the total environment*, 631, 1279-1294. https://doi.org/10.1016/j.scitotenv.2018.03.088

[8] Kourgialas, N. N., Dokou, Z. & Karatzas, G. P. (2015). Statistical analysis and ANN modelling for predicting hydrological extremes under climate change scenarios: The example of a small Mediterranean agro-watershed. *Journal of Environmental Management, 154*, 86-101. https://doi.org/10.1016/j.jenvman.2015.02.034

[9] Fan, S. & Rue, C. (2020). The role of smallholder farms in a changing world. *The role of smallholder farms in food and nutrition security*, 13-28. https://doi.org/10.1007/978-3-030-42148-9_2

[10] Adewusi, A. O., Asuzu, O. F., Olorunsogo, T., Iwuanyanwu, C., Adaga, E. & Daraojimba, D. O. (2024). AI in precision agriculture: A review of technologies for sustainable farming practices. *World Journal of Advanced Research and Reviews, 21*(1), 2276-2285. https://doi.org/10.30574/wjarr.2024.21.1.0314

[11] Paul, K., Chatterjee, S. S., Pai, P., Varshney, A., Juikar, S., Prasad, V., ... & Dasgupta, S. (2022). Viable smart sensors and their application in data-driven agriculture. *Computers and Electronics in Agriculture, 198*, 107096. https://doi.org/10.1016/j.compag.2022.107096

[12] Schimmelpfennig, D. (2016). Farm profits and adoption of precision agriculture. *USDA Economic Research Report*.

[13] Mulla, D. J. (2013). Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. Biosystems Engineering, 114(4), 358-371. https://doi.org/10.1016/j.biosystemseng.2012.08.009

[14] Alfred, R., Obit, J. H., Chin, C. P. Y., Haviluddin, H. & Lim, Y. (2021). Towards paddy rice smart farming: a review on big data, machine learning, and rice production tasks. *IEEE Access, 9*, 50358-50380. https://doi.org/10.1109/ACCESS.2021.3069449

[15] Javaid, M., Haleem, A., Singh, R. P. & Suman, R. (2022). Enhancing smart farming through the applications of Agriculture 4.0 technologies. *International Journal of Intelligent Networks, 3*, 150-164. https://doi.org/10.1016/j.ijin.2022.09.004

[16] Tan, S., Liu, J., Lu, H., Lan, M., Yu, J., Liao, G., ... & Ma, X. (2022). Machine learning approaches for rice seedling growth stages detection. *Frontiers in Plant Science, 13*, 914771. https://doi.org/10.3389/fpls.2022.914771

[17] Saha, H. N., Roy, R., Chakraborty, M. & Sarkar, C. (2021). IoT-enabled agricultural system application, challenges and security issues. *Agricultural informatics: automation using the IoT and machine learning*, 223-247. https://doi.org/10.1002/9781119769231.ch11

[18] Albahri, A. S., Khaleel, Y. L., Habeeb, M. A., Ismael, R. D., Hameed, Q. A., Deveci, M., ... & Alzubaidi, L. (2024). A systematic review of trustworthy artificial intelligence applications in natural disasters. *Computers and Electrical Engineering, 118*, 109409. https://doi.org/10.1016/j.compeleceng.2024.109409

[19] Zameer, A., Arshad, J., Khan, A. & Raja, M. A. Z. (2017). Intelligent and robust prediction of short-term wind power using a genetic programming-based ensemble of neural networks. *Energy conversion and management, 134*, 361-372. https://doi.org/10.1016/j.enconman.2016.12.032

[20] Patrício, D. I. & Rieder, R. (2018). Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and electronics in agriculture, 153*, 69-81. https://doi.org/10.1016/j.compag.2018.08.001

[21] Shaikh, T. A., Rasool, T. & Lone, F. R. (2022). Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming. *Computers and Electronics in Agriculture, 198*, 107119. https://doi.org/10.1016/j.compag.2022.107119

[22] Li, Y. & Chao, X. (2020). ANN-based continual classification in agriculture. *Agriculture, 10*(5), 178. https://doi.org/10.3390/agriculture10050178

[23] Anami, B. S., Malvade, N. N. & Palaiah, S. (2020). Deep learning approach for recognition and classification of yield affecting paddy crop stresses using field images. *Artificial intelligence in agriculture, 4*, 12-20. https://doi.org/10.1016/j.aiia.2020.03.001

[24] Iqbal, Z., Khan, M. A., Sharif, M., Shah, J. H., Ur Rehman, M. H. & Javed, K. (2018). Automated detection and classification of citrus plant diseases using image processing techniques: A review. *Computers and electronics in agriculture, 153*, 12-32. https://doi.org/10.1016/j.compag.2018.07.032

[25] Subeesh, A. & Mehta, C. R. (2021). Automation and digitization of agriculture using artificial intelligence and the Internet of Things. *Artificial Intelligence in Agriculture, 5*, 278-291. https://doi.org/10.1016/j.aiia.2021.11.004

[26] Mumuni, A. & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array, 16*, 100258. https://doi.org/10.1016/j.array.2022.100258

[27] He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778. https://doi.org/10.1109/CVPR.2016.90

[28] Howard, A. G. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint *arXiv*: 1704.04861.

[29] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818-2826. https://doi.org/10.1109/CVPR.2016.308

[30] Oza, N. C. & Tumer, K. (2008). Classifier ensembles: Select real-world applications. *Information fusion, 9*(1), 4-20. https://doi.org/10.1016/j.inffus.2007.07.002

[31] Tabik, S., Peralta, D., Herrera-Poyatos, A. & Herrera, F. (2017). A snapshot of image pre-processing for convolutional neural networks: a case study of MNIST. *International Journal of Computational Intelligence Systems, 10*(1), 555-568. https://doi.org/10.2991/ijcis.2017.10.1.38

[32] Mumtaz, W., Rasheed, S. & Irfan, A. (2021). Review of challenges associated with the EEG artefact removal methods. *Biomedical Signal Processing and Control, 68*, 102741. https://doi.org/10.1016/j.bspc.2021.102741

[33] Milo, T. & Somech, A. (2020, June). Automating exploratory data analysis via machine learning: An overview. In *Proceedings of the 2020 ACM SIGMOD international conference on management of data*, 2617-2622. https://doi.org/10.1145/3318464.3383126

[34] Burger, W. & Burge, M. J. (2013). *Principles of digital image processing: advanced methods*. Springer Science & Business Media. https://doi.org/10.1007/978-1-84882-919-0

[35] Nelson, A., Setiyono, T., Rala, A. B., Quicho, E. D., Raviz, J. V., Abonete, P. J., ... & Ninh, N. H. (2014). Towards an operational SAR-based rice monitoring system in Asia: Examples from 13 demonstration sites across Asia in the RIICE project. *Remote Sensing, 6*(11), 10773-10812. https://doi.org/10.3390/rs61110773

[36] Ferentinos, K. P., Katsoulas, N., Tzounis, A., Bartzanas, T. & Kittas, C. (2017). Wireless sensor networks for greenhouse climate and plant condition assessment. *Biosystems Engineering, 153*, 70-81. https://doi.org/10.1016/j.biosystemseng.2016.11.005

[37] Wang, R., Cheng, T. & Hu, L. (2015). Effect of wide–narrow row arrangement and plant density on yield and radiation use efficiency of mechanized direct-seeded canola in Central China. *Field Crops Research, 172*, 42-52. https://doi.org/10.1016/j.fcr.2014.12.005

Author's contacts:

Durodola Folasade
(Corresponding Author)
Department of Mechatronics Engineering,
Federal University of Agriculture,
Alabata Road, Abeokuta, Ogun State, 110111, Nigeria
durodolafo@funaab.edu.ng

Owoeye Samuel
Department of Mechatronics Engineering,
Federal University of Agriculture,
Alabata Road, Abeokuta, Ogun State, 110111, Nigeria

Kamil- Bello Furqan
Department of Mechatronics Engineering,
Federal University of Agriculture,
Alabata Road, Abeokuta, Ogun State, 110111, Nigeria

Daodu Sakira
Department of Mechatronics Engineering,
Federal University of Agriculture,
Alabata Road, Abeokuta, Ogun State, 110111, Nigeria

Makinde Kayode
Department of Mechatronics Engineering,
Federal University of Agriculture,
Alabata Road, Abeokuta, Ogun State, 110111, Nigeria

Folarami Olaniyi
Department of Mechatronics Engineering,
Federal University of Agriculture,
Alabata Road, Abeokuta, Ogun State, 110111, Nigeria