engineering proceedings

MDPI

Proceedings Paper

A Compressed Convolutional Neural Network (CNN) Model for Rice Yield Detection at Ripening Stage Using Weight Pruning †

Usman Umar [1,*], Mohammed Hassan [2], Mohammed Hamada [3] and Habeebah Kakudi [4]

1 Federal Universuty Kashere, Gombe
2 Bayero University, Kano; mhasan.se@buk.edu.ng
3 University of Aizu; hamada@U-aizu.ac.jp
4 Bayero University, Kano; hakakudi.cs@buk.edu.ng
* Correspondence: usmanumar@fukashere.edu.ng

† Presented at the The 4th International Electronic Conference on Applied Sciences, 27 Oct–10 Nov 2023; Available online: https://asec2023.sciforum.net/.

Abstract: This research proposes a compressed Convolutional Neural Network (CNN) model for rice yield detection using weight pruning technique. The initial CNN model achieved an accuracy of 98.10% on a dataset comprising 3150 images of both yield and unyield rice crops. However, it had a large size of 603MB, posing challenges in terms of deployment and storage. To address this issue, we adopted "Magnitude-based Weight Pruning." This technique involves ranking model weights based on their magnitudes and removing the less influential ones. The compressed model achieved a significant reduction in size to 168MB, representing a reduction of approximately 72.3%, while maintaining a reasonable accuracy of 96.67%. The research was conducted in three phases. First, a dataset of 3150 images of yield and unyield rice crops was collected from different farms in Kano metropolis and preprocessed by resizing them to 250X250 pixels. Second, a CNN model with 27 layers was designed and trained using the preprocessed dataset. The model achieved an accuracy of 99% on the training set and 98.10% on the test set. Finally, weight pruning was applied to the trained CNN model to reduce its size. The compressed model exhibited a size of 168MB and an accuracy of 96.67%. The results of this research demonstrate the effectiveness of weight pruning as a viable technique for compressing CNN models without significantly compromising their accuracy. The compressed model, with its reduced size, is well-suited for deployment on resource-constrained devices for rice yield detection applications.

Keywords: Convolutional Neural Network; rice yield detection; weight pruning; image compression

Citation: Umar, U.; Hassan, M.; Hamada, M.; Kakudi, H. A Compressed Convolutional Neural Network (CNN) Model for Rice Yield Detection at Ripening Stage Using Weight Pruning. *Eng. Proc.* **2023**, *52*, x. https://doi.org/10.3390/xxxxx

Academic Editor(s):

Received: date
Revised: date
Accepted: date
Published: date

1. Introduction

Rice cultivation plays a crucial role in ensuring food security for a significant portion of the global population. Accurate and timely detection of rice yield is essential for effective agricultural management [1]. In the past few years, Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in image classification tasks, including crop yield detection [2]. However, the deployment of CNN models for such applications often faces challenges related to their substantial memory and stor-age requirements [3].

To address these challenges, researchers have attempted to propose compressed CNN models suitable for utilization on devices with low computing resources. In this paper, we propose a compressed CNN model for rice grain yield detection at the ripening stage using weight pruning. Our model aims to address the limitations found in current models and achieve comparable performance with the original architecture.

The objectives of our research are to collect a large image dataset of paddy rice, preprocess and prepare it for training, develop and compress a CNN model using weight pruning technique, evaluate its performance using various metrics, and compare the performance of the compressed model with that of the original architecture. Specifically, magnitude weight pruning technique was used for the compression, and the evaluation metrics used to compare the performance of the compressed model with that of the original architecture were the accuracy and size of the models.

Several related works have been conducted on crop yield detection using machine learning techniques such as regression tree, random forest, multivariate regression, association rule mining, and artificial neural networks [4].

However, these models have limitations in terms of their size, memory utilization, and processing capabilities, making them difficult to deploy on resource-constrained devices. Our proposed compressed CNN model will be suitable for utilization on devices with low computing resources, such as drones, mobile phones, and smart cameras. The model will help farmers detect the optimum maturity of rice to prevent pest damage, prevent crop wastage, loss and spoilage on the field, and improve the quality of crops and farmers' income. Additionally, we collected a large dataset of paddy rice from farms, which was used to train and test the model. The dataset will be made available on Kaggle, allowing other researchers to utilize it for further studies.

The methodology section provides a detailed description of the experimental setup, including the hardware and software configuration, and the experiment phases involved in developing the compressed CNN model for rice yield detection.

2. Materials and Method:

This section shows the entire proposed procedure that was adopted in developing the model for a compressed deep learning model for rice yield detection at ripening stage using weight pruning. The entire procedure is segmented into essential phases in the following subsections, commencing with the collection of images for the classification process utilizing convolutional neural networks

2.1. Dataset Collection/Image Acquisition

To properly train and developed a rice yield detection model, a considerable amount of paddy rice images had to be obtained as a starting dataset. Without sufficient and representative training data, the model cannot learn the underlying discriminative patterns required to make robust classifications [5]. 70% of the dataset was used for training and 30% of the dataset for testing.Firstly, a unique dataset of yield and unyield rice crop has been captured using a digital camera and an android phone from different farms from Kura, Garko, and Tudun Wada Area all in Kano metropolis. Fig. 2.1: Shows the Proportion of Yield and Unyield images in the dataset.

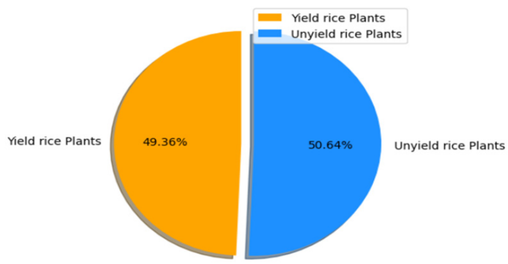

Figure 1. Proportion of Yield and Unyield images in the dataset.

Initially, we captured a total of 4250 images of both yield and unyielding paddy rice from the field. Due to significant background distortion, extraneous elements, and the presence of duplicate images, we conducted a curation process, resulting in a refined dataset comprising 3150 images. Among these, 1580 images belong to the unyielding rice category, while the remaining 1540 images represent yield rice. This dataset will be made available on the Kaggle website for public use. Sample images from the dataset are presented as demonstrated in Figure 1, these images are representative samples from the rice yield detection dataset. Specifically, (a) and (b) feature yield crops of paddy rice, while (c) and (d) display unyielding paddy rice crops.

(a) **(b)**

(c) **(d)**

Figure 2. Sample Images of Yield and Unyield images from the dataset.

2.3. Image Pre-processing and Labelling

Images were captured with different smart phones and digital camera of different resolutions, different formats, and different quality. To enhance feature extraction, the images slated for use as the dataset for the CNN model underwent preprocessing to achieve greater uniformity.

The dataset images the were preprocessed by cropping, resizing, normalized and labeling.

2.3.1. Image Cropping

Image cropping was manually performed for the images which requires removal of unwanted parts.

2.3.2. Image Resizing

Images were captured in different sizes with different cameras and different smart phones. Therefore, to improve computational efficiency, maintain spatial information and mitigate overfitting, image resize is needed. Images were resized to 250x250 pixels to make all the dataset to be of equal size, which will also help in feature extraction to get the balanced features.

2.3.3. Labeling

Each image in the dataset was meticulously labeled to indicate whether it represented a yield or unyield rice crop. This labeling is crucial for supervised learning, as it provides the ground truth information required for training the convolutional neural network (CNN) model. Expert agronomists assisted in this crucial step to ensure accurate and reliable labeling.

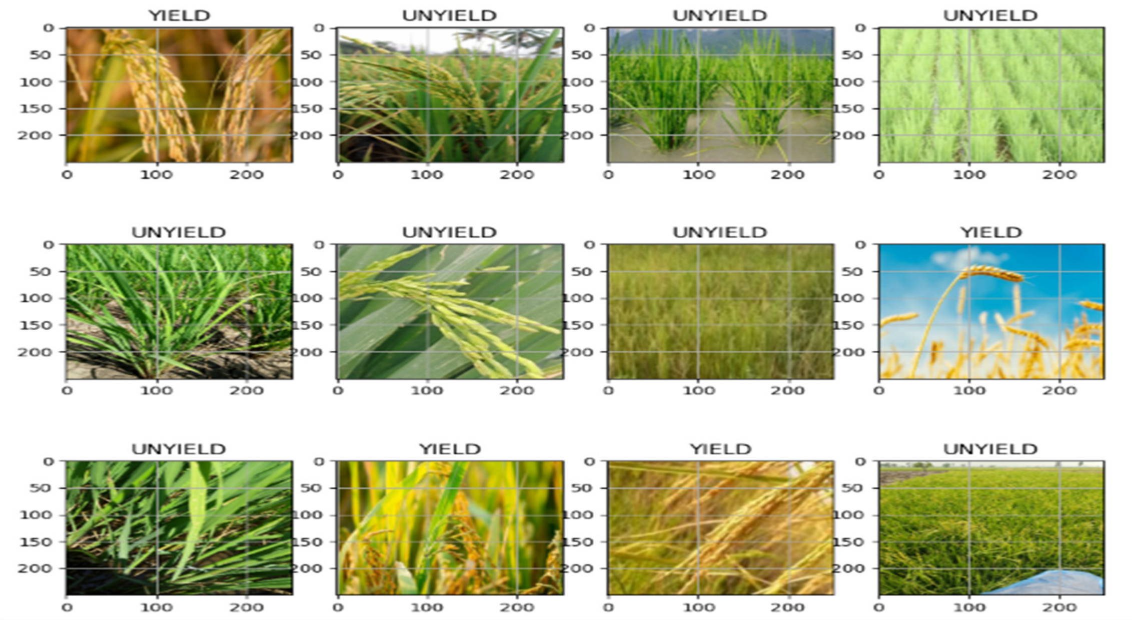

Figure 3. Proportion of Yield and Unyield images in the dataset.

2.4. Feature Extraction

The captured images were in RGB (Red, Green, and Blue) format, there are total of 3150 images of rice dataset. 1540 and 1580 samples of yield and unyield images respectively which was annotated with the help of an experienced agronomists. A convolutional neural network (CNN) was used to extract features from the images. The CNN was trained on the processed dataset, and the weights of the CNN were optimized to achieve best performance.

2.5. Model Formulation

The workflow of the compressed CNN model for rice yield detection is shown in Figure 4 below: The workflow of the compressed CNN model for rice yield detection involves the following steps:

- **Data Collection**: Images of yield and unyield rice crops were captured from different rice farms in Kano metropolis using digital cameras and mobile phones.
- **Data Preprocessing:** The rice yield dataset was cleaned and resized to the same size. The data was then transformed and prepared for analysis using Keras and TensorFlow frameworks.
- **Model Building:** The CNN model was built using TensorFlow and Keras inbuilt functions. It consisted of convolution, activation, pooling, and fully connected layers.
- **Image Augmentation**: The ImageDataGenerator function was used to perform image augmentation, generating new images with various transformations. Parameters such as rotation, zoom, shift, and batch size were specified.
- **Dataset Enhancement:** Before passing the dataset to the ImageDataGenerator function, mathematical operations like normalization and scaling were performed. This increased the dataset size and introduced more variation to prevent overfitting.
- **Model Training:** The augmented dataset was used to train the CNN model, which was then saved.
- **Model Compression:** The output of the augmented CNN model was compressed using weight pruning technique.
- **Model Evaluation**: The compressed model was trained, tested, and saved. Both the compressed and Base CNN models were evaluated based on their performance and memory utilization.

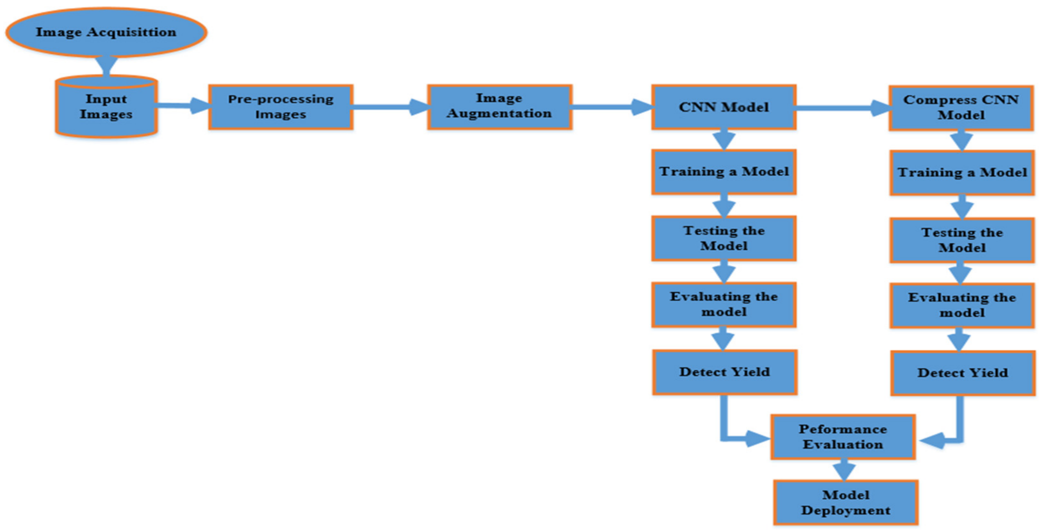

Figure 4. Proposed Model framework.

2.6. Experimental Setup

This section outlines an experimental setup for developing, compressing, and comparing Convolutional Neural Network (CNN) models aimed at detecting rice yield during the ripening stage. The experiment was conducted on Google Colab, which provides access to GPUs and TPUs for efficient model training and collaborative research. The setup includes hardware and software details, such as the operating system and specific software versions. The CNN model architecture was defined, and the Rice Yield Dataset was prepared by dividing it into training and testing subsets. Data augmentation techniques were applied during training to increase dataset diversity. The setup encompasses multiple phases in this research endeavor:

1. Development CNN Model for Rice Yield Detection:

- A specialized CNN model was created for the specific task of rice yield detection.
- The Rice Yield Dataset, consisting of 3150 labeled images of both yield and unyielding rice plants, was configured and loaded.
- The dataset was partitioned, with 70% allocated for training and 30% for testing, to create distinct subsets for these purposes.
- All images were resized uniformly to dimensions of 250 x 250 pixels.

2. **Defining the CNN model architecture**: The architecture of the model was outlined in Table 1 below:

Table 1. The CNN model architecture for rice yield detection.

LAYER NUMBER	LAYER TYPE	OUTPUT SHAPE	NUMBER OF PARAMETERS
1	Input	(250, 250, 3)	0
2	Conv2D	(250, 250, 32)	896
3	Activation (ReLU)	(250, 250, 32)	0
4	BatchNormalization	(250, 250, 32)	128
5	MaxPooling2D	(83, 83, 32)	0
6	Dropout	(83, 83, 32)	0
7	Conv2D	(83, 83, 64)	18496
8	Activation (ReLU)	(83, 83, 64)	0
9	BatchNormalization	(83, 83, 64)	256
10	Conv2D	(83, 83, 64)	36928
11	Activation (ReLU)	(83, 83, 64)	0

12	BatchNormalization	(83, 83, 64)	256
13	MaxPooling2D	(41, 41, 64)	0
14	Dropout	(41, 41, 64)	0
15	Conv2D	(41, 41, 128)	73856
16	Activation (ReLU)	(41, 41, 128)	0
17	BatchNormalization	(41, 41, 128)	512
18	Conv2D	(41, 41, 128)	147584
19	Activation (ReLU)	(41, 41, 128)	0
20	BatchNormalization	(41, 41, 128)	512
21	MaxPooling2D	(20, 20, 128)	0
22	Dropout	(20, 20, 128)	0
23	Flatten	51200	0
24	Dense	1024	52429824
25	Activation (ReLU)	1024	0
26	BatchNormalization	1024	4096
27	Dropout	1024	0
28	Dense (Output)	1	1025

Description of the model Architecture:

The CNN model architecture for rice yield detection is designed for binary image classification tasks using TensorFlow and Keras libraries. The model takes RGB images of size 250x250 pixels as input and has several convolutional layers with ReLU activation, batch normalization, and dropout layers for regularization. MaxPooling2D layers are used to downsample feature maps, and the Flatten layer converts 2D feature maps into a 1D vector. Fully connected layers refine these features before the final output prediction is made. The architecture is designed to capture meaningful information from input images to predict rice yield accurately.

3. The Model Compilation and Training

In the model compilation and training phase, the rice yield detection model was configured with the Adam optimizer and Binary Cross-Entropy loss function. Various evaluation metrics, including Binary Accuracy, Precision, Recall, F1-Score, and Area Under the Curve (AUC), were chosen to assess performance. To prevent overfitting, an EarlyStopping callback was employed during training, which was conducted for 25 epochs. Additionally, key evaluation metrics like Accuracy, Precision, Recall, F1-Score, and Model Size were highlighted as essential for assessing the model's performance and feasibility in deployment.

3. Result and Discussion

This section presents the findings of a study focused on the impact of a novel approach to model compression using a curated rice yield image dataset. The goal is to compress the rice yield model to make it suitable for resource-constrained devices like mobile phones and drones while maintaining accuracy.

The study introduces a compressed Convolutional Neural Network (CNN) model, leveraging weight pruning, to accurately predict rice grain yield during the ripening stage. The initial (base) model achieved an impressive accuracy of 98.10%, but after weight pruning, the compressed model achieved a slightly lower accuracy of 96.67%. Despite this reduction, the compressed model maintains a high level of performance. **Figure 5:** titled "Performance of the Models on Paddy Rice Detection," is presented below: This graph provides an overview of the evaluation performance metrics for the model used in the study.

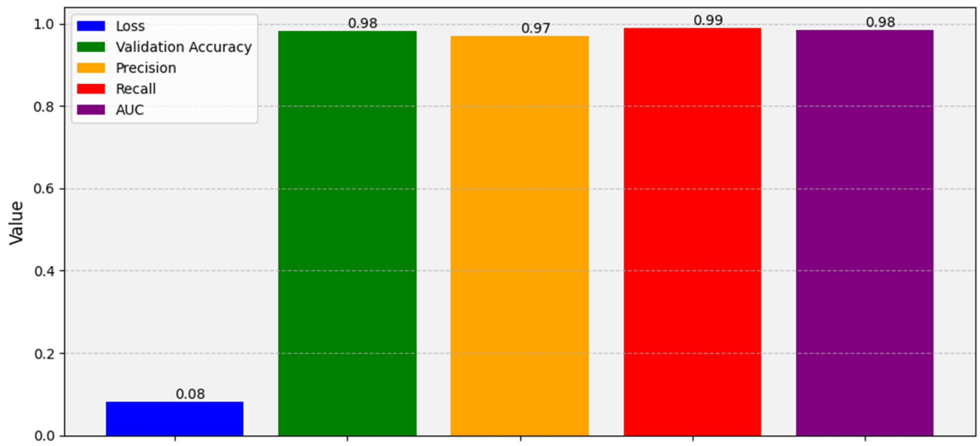

Figure 5. Performance of the proposed Models on Paddy Rice Detection Model.

Additionally, the size of the base model was reduced significantly from 603MB to 168MB through compression. This reduction enables more efficient storage and deployment on devices with limited resources. Table 2**:** titled "Comparison of the Rice Yield Model and the Compressed Model," is presented below: This Table provides an overview of result of compression.

Table 2. Comparison of the Rice Yield Model and the Compressed Model result.

Models	**Accuracy**	**Model Size**
Ric Rice Yield Model	**98.10%**	**603MB**
Pruned Model	**96.667%**	**168MB**

4. Conclusion and Future Work

In summary, our study focused on developing an innovative method for rice yield detection using a Convolutional Neural Network (CNN) model, leveraging a unique dataset of rice yield images. Key findings include the successful creation of this novel dataset, extensive data preprocessing, and the division of the dataset into training and testing subsets.

The CNN model's performance was exceptional, achieving high accuracy (98%) and significant metrics such as precision (97%), recall (99%), and an AUC of 98%. This highlights the effectiveness of CNNs in learning complex relationships within rice images.

Through weight pruning techniques, we compressed the CNN model while maintaining commendable accuracy (96.67%) and significantly reducing its size, from 608MB to 168MB. This demonstrates the feasibility of deploying such models on resource-constrained devices.

Future research endeavors should prioritize the optimization of techniques that strike a balance between accuracy and model size within the pruned model framework. Exploration of advanced compression methods like quantization and knowledge distillation can provide valuable insights into enhancing the model's efficiency.

Efforts should also focus on the seamless integration of the compressed CNN model into existing agricultural technologies and devices. This integration has the potential to empower farmers with improved yield prediction capabilities, ultimately enhancing agricultural practices and productivity.

In conclusion, our study contributes significantly to the field of rice yield prediction by proposing an efficient CNN model using weight pruning techniques. While there is a slight accuracy reduction compared to the base model, the compressed model maintains high performance. Additionally, the substantial reduction in model size enables efficient deployment on resource-constrained devices, with further advancements expected to benefit agricultural practices and productivity.

Author Contributions:

Funding:

Institutional Review Board Statement:

Informed Consent Statement:

Data Availability Statement:

Conflicts of Interest:

References

1. H. Chen, M. Fang, Y. Chen, Y. Xu, and C. Tao, "A Hyperspectral Image Classification Method Using Multifeature Vectors and Optimized KELM," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 14, p. 2781, 2021.
2. M. Hasan and J. Shin, "Online Kanji Characters Based Writer Identification Using Sequential Forward Floating Selection and Support Vector Machine," Appl. Sci., vol. 12, p. 10249, 2022.
3. S. S. Sawant and P. Manoharan, "Unsupervised band selection based on weighted information entropy and 3D discrete cosine transform for hyperspectral image classification," Int. J. Remote Sens., vol. 41, pp. 3948–3969, 2020.
4. Y. Song, X. Cai, X. Zhou, B. Zhang, H. Chen, Y. Li, W. Deng, and W. Deng, "Dynamic hybrid mechanism-based differential evolution algorithm and its application," Expert Syst. Appl., vol. 213, p. 118834, 2023.

A. M. Roy, "Adaptive transfer learning-based multiscale feature fused deep convolutional neural network for EEG MI multiclassification in brain–computer interface," Eng. Appl. Artif. Intell., vol. 116, p. 105347, 2022.

5. K. Pius, Y. Li, E. A. Agyekum, T. Zhang, Z. Liu, P. T. Yamak, and F. Essaf, "SD-UNET: Stripping down U-Net for Segmentation of Biomedical Images on Platforms with Low Computational Budgets," Diagnostics, vol. 10, p. 110, 2021.
6. Y. Li, L. Gross, Z. Li, X. Li, X. Fan, and W. Qi, "Automatic Building Extraction on High-Resolution Remote Sensing Imagery Using Deep Convolutional Encoder-Decoder with Spatial Pyramid Pooling," IEEE Access, vol. 7, pp. 128774–128786, 2019.
7. M. O. Lawal, "Tomato detection based on modified YOLOv3 framework," Sci. Rep., vol. 11, 2021.
8. Y. Wu, G. Wan, L. Liu, Z. Wei, and S. Wang, "Intelligent Crater Detection on Planetary Surface Using Convolutional Neural Network," in Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021, pp. 1229–1234.
9. M. Zhang, W. Li, and Q. Du, "Diverse Region-Based CNN for Hyperspectral Image Classification," IEEE Trans. Image Process., vol. 27, pp. 2623–2634, 2018.
10. Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation," arXiv, vol. 39, pp. 1856–1867, 2019.
11. Y. W. Hu, J. Leu, M. Faisal, and S. W. Prakosa, "Analysis of Model Compression Using Knowledge Distillation," IEEE Access, vol. 10, pp. 85095–85105, 2022.
12. Z. Wang, S. Lin, J. Xie, and Y. Lin, "Pruning Blocks for CNN Compression and Acceleration via Online Ensemble Distillation," IEEE Access, vol. 7, pp. 175703–175716, 2019.
13. Y. Zhou, G. G. Yen, and Z. Yi, "Evolutionary Shallowing Deep Neural Networks at Block Levels," IEEE Trans. Neural Netw. Learn. Syst., pp. 1–13, 2021.
14. H. Song, H. Mao, W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," in Proceedings of the 4th International Conference on Learning Representations, ICLR 2016—Conference Track Proceedings, San Juan, Puerto Rico, 2–4 May 2016, pp. 1–14.