CAUCHY: Jurnal Matematika Murni dan Aplikasi
Volume 10(1) (2025), Pages 278-289
p-ISSN: 2086-0382; e-ISSN: 2477-3344

Enhancing Image Classification of Cabbage Plant Diseases Using a Hybrid Model Convolutional Neural Network and XGBoost

Nabila Ayunda Sovia[1], Ni Wayan Surya Wardhani[1], Eni Sumarminingsih[1], Elvo Ramadhan Shofa[2,*]

[1]Departemen of Statistics, FMIPA, Brawijaya University, Malang, Indonesia
[2]Beijing Institute of Technology, China

Email: nabilaayunda003@student.ub.ac.id, elvoramadhanshofa2005@gmail.com*

ABSTRACT

Classifying imbalanced datasets presents significant challenges, often leading to biased model performance , particularly in multiclass classification. This study addresses these issues by integrating Convolutional Neural Networks (CNN) and XGBoost, leveraging CNN's exceptional feature extraction capabilities and XGBoost's robust handling of imbalanced data. The Hybrid CNN-XGBoost model was applied to classify cabbage plants affected by pests and diseases, which are categorized into five classes, with a significant imbalance between healthy and affected plants. The dataset, characterized by severe class imbalance, was effectively handled by the proposed model. A comparative analysis demonstrated that the CNN-XGBoost approach, with a Balanced Accuracy of 0.93 compared to 0.53 for the standalone CNN, significantly outperformed the standalone model, particularly for minority class predictions. This approach not only enhances the accuracy of plant disease and pest diagnosis but also provides a practical solution for farmers to efficiently identify and classify cabbage plants, contributing to more effective agricultural management.

Keywords: Cabbage classification; Convolutional Neural Network; Imbalance Data; XGBoost

INTRODUCTION

Cabbage, is an important crop in agricultural economies and is commonly grown in highland areas, where outbreaks can lead to numerous symptoms of diseases and pest attacks, which leading to significant losses for farmers if not properly managed [1]. Traditionally, farmers have relied on visual inspection to identify and differentiate between pests and diseases in cabbage plants to monitor their crops. This task is labor intensive and requires modern, technology driven solutions to support effective crop management [2]. One approach to improving cabbage farming is the use of CNN as modern technology that can mimic human visual recognition to enhance farmers ability to manage their crops. CNNs are known for their strong feature extraction abilities and high accuracy in image classification tasks, because of their convolution technique that

Submitted: December 18, 2024 Reviewed: December 31, 2024 Accepted: February 10, 2025
DOI: https://doi.org/10.18860/cauchy.v10i1.30866

extract detailed feature in image [3,4].

However, despite their advantages, CNN face challenges when applied to agricultural datasets in real world scenarios, as these datasets are often imbalanced. For example, images of pests are usually much more common than images of diseases, resulting in uneven data distribution that makes it difficult for the model to accurately identify less common categories [5]. CNN often focus more on the majority classes in imbalanced datasets, which reduces their effectiveness in such situations, as mentioned by Dablain et al. [6]. To overcome this issue, innovative methods are needed to ensure better balance in classification performance. One effective solution is to use ensemble techniques, which are known for their ability to handle imbalanced data well [7].

Among these techniques, XGBoost is particularly useful due to its gradient-boosting algorithm, which helps address class imbalance while improving the accuracy of classification [8]. The combination of CNNs and XGBoost leverages the strengths of both methods, which CNN excel in extracting meaningful features from image data, while XGBoost reduce variants in imbalanced data to enhance classification accuracy [9]. Previous studies have demonstrated the potential of this Hybrid approach. Jiao et al. applied CNN-XGBoost with APSO optimization on balanced data, achieving a remarkable accuracy of 99% across three different datasets [10]. Similarly, Gao et al., in a study on imbalanced multiclass data using a comparative ensemble technique with SMOTE resampling, reported that the combination of CNN and XGBoost achieved nearly 80% accuracy, though challenges in generalizing to minority classes were noted [11]. These studies underscore the effectiveness of CNN-XGBoost but also reveal gaps in its application to highly imbalanced multiclass datasets.

This research addresses these gaps by applying the CNN-XGBoost framework to cabbage image datasets with severe class imbalance, where pest related images vastly outnumber disease related ones. The aims to improve classification accuracy and strengthen the model's ability to generalize to minority classes. By optimizing the CNN-XGBoost methodology, the research aims to provide new insights into handling imbalanced agricultural datasets and advancing Precision agriculture.

METHODS

Dataset

This research utilizes primary data. The dataset focuses on cabbage plants impacted by diseases and pests in 2023, specifically from Poncokusumo, Malang. The data comprises images of cabbage plants, which were collected from fields ranging in size from 1.000 to 3.000 square meters. These images capture plants that were 70 to 80 days old at the time of observation, before harvest. The dataset includes a total of 242 images, categorized into five distinct types based on the specific diseases and pests affecting the cabbage. Notably, the data has already undergone preprocessing and cleaning to ensure its quality. The type of each disease and pest, along with the number of cabbage plants affected by them, can be seen in Table 1.

Table 1. Data Information on Cabbage Plants Affected by Various Pests and Diseases

No	Type	Code	Count
1	Insect Pests	P3	170
2	Foliar Disease	P6	35
3	Brassica rot Disease	P7	12

4	Bacterial Soft Rot	P8	15
5	Clubroot disease	P9	10

As shown in Table 1, the dataset used in this study is highly imbalanced, which requires the application of specific methods to address this issue. Figure 1 displays images corresponding to each disease and pest.

(a) **(b)** **(c)** **(d)** **(e)**

Figure 1. Cabbage Affected by Diseases and Pests (a) P3, (b) P6, (c) P7, (d) P8, (e) P9

Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN) are nonlinear computations used for tasks such as regression or classification [12]. That consist of Layers, and process input data through mathematical operations involving parameters and biases Simple Neural Network model represented by Equation (1).

$$Y_i = \boldsymbol{W}_i \, X_{i-1} + b_i \tag{1}$$

Where $\boldsymbol{W}_i$ represents the weights between Layers, which are multiplied by the input from the previous Layer (X_{i-1}) and then added with bias b_i. This process is followed by applying a nonlinier activation function to handle high dimensional data effectively [13].

Processing image data needed feature extraction phase, which forming the core of the network architecture. Several architercture exist for running CNN, and the latest one, ConvNeXt-tiny, offers good performance with a small number of parameters [14]. Figure 3 illustrates the structure of the ConvNeXt-tiny architecture in this study.

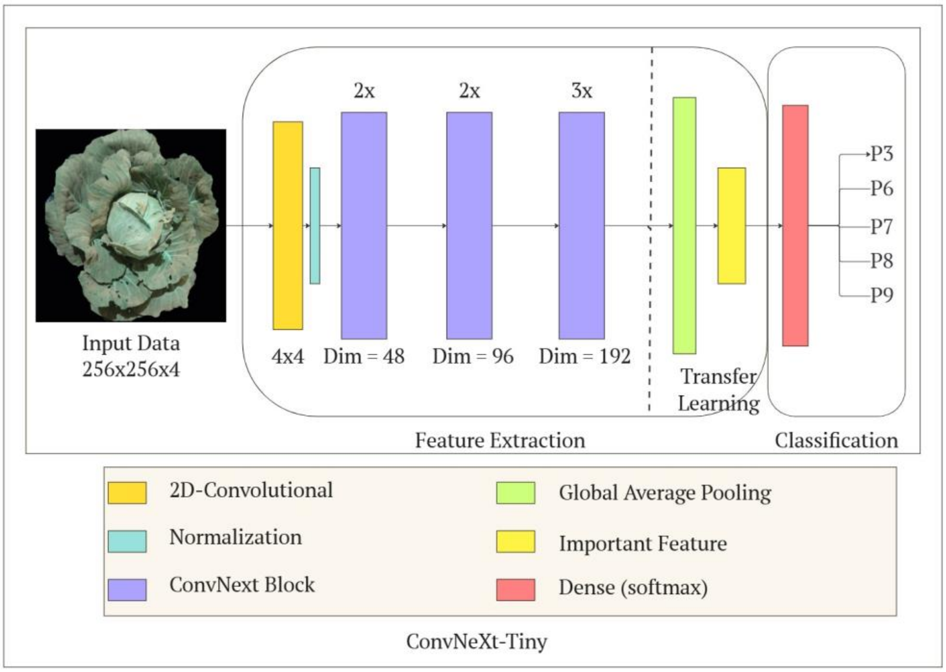


Figure 2. Illustration of ConvNext-Tiny used in this Study

Based on the illustration of ConvNeXt-Tiny in Figure 2, the CNN-based classification algorithm consists of two primary components, which is a feature extraction Layer and a classification Layer. The following section will explain the Layers in a CNN.

- **Feature Extraction Layer**

  - **Convolution Layer**

    The feature extraction Layer contains a 2D-Convolutional Layer that extracts local patterns like edges, textures, or simple shapes. This process is using filter as shown in Equation (2).

$$Y_i = F(X_{i-1}, X_i) \tag{2}$$

A filter F, consisting of small n x n matrices, slides over the input image denoted by (X_{i-1}), performing dot products and summing the results into a single value (X_i), then it ise mapped onto a new fature representation (feature map). This process is illustrated in Figure 3.

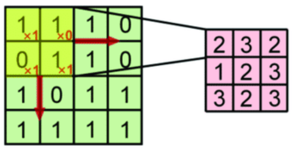


Figure 3. Illustration of the Convolutional Process

The convolutional process illustrated in Figure 4 shows a yellow 2x2 matricx undergoing dot product operation with a 2x2 filter, resulting in a single value of 2, which is the mapped onto a new feature representation depicted as a pink 9x9 matrix [14].

- **Nonlinierity Function**

  Next, nonlinearity is introduced to the model using GELU (Gaussian Error Linear Unit) as the activation functions. This nonlinearity allows the network to capture complex relationships in the data that go beyond simple linear mappings [15]. The activation function used in ConvNeXt is GELU, shown in Equation (3)

$$\mathrm{GELU}(\mathrm{X}) = 0{,}5\mathrm{X}\left(1 + \tanh\left[\frac{2}{\pi}\sqrt{\mathrm{X} + 0{,}44715\ \mathrm{X}^3}\right]\right) \tag{3}$$

- **Downsampling**

  After transforming the data to non linier using GELU, then the data will be applied into down sampling Layer to reduce the spatial dimensions (width and height) of the feature maps while simultaneously doubling the number of channels (Dim = 48 → 96 → 196) [14].

- **Global Average Pooling**

  The high-dimensional output from the down sampling process (195) is then transformed using a Global Average Pooling Layer, which takes the output feature maps and reduces each channel to a single representative value through an averaging operation. This process is illustrated in Figure 4.

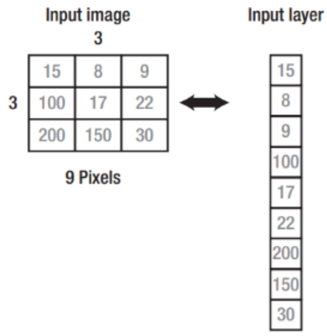


Figure 4. Illustration of Dimension Transformation

The Layer calculates the average for each feature map, and the resulting vector is directly used as an important feature, either transferred to another model through transfer learning or, in the ConvNeXt-Tiny architecture, fed into the Softmax Layer for classification.

- **Classification Layer**
  - **Softmax Activation Layer**

The Dense (Softmax) Layer (red block) outputs probabilities for each class (P3, P6, P7, P8, and P9), allowing the network to determine the most likely category for the input data. The Softmax activation function is employed in the output Layer for multi-class classification, as shown in Equation (5) [16].

$$f(Z) = \frac{e^{Z_k}}{\sum_{k=0}^{K} e^{Z_k}} \tag{5}$$

Where Z represents the output from the Hidden Layer fed into the activation function, and k denotes the number of classes.

XGBoost

XGBoost employs a tree-boosting approach, correcting the errors of the previous model then each tree is added sequentially [17]. The model is formed by minimizing the objective function through training the model using an iterative function, so the objective function and the t-th iterative function can be written in Equation (6) [10].

$$O^{(t)} = \sum_{i=1}^{n} l\left(Y_i, \hat{Y}_i^{(t-1)} + f_t(X_i)\right) + \Omega(f_t) \tag{6}$$

Loss function denoted by l, calculates the gap between the results of the classification and the actual value and Ω is the penalty term (regularization) used to prevent overfitting. The first decision tree model is formed. The K-th boosted tree model with as the output is defined in Equation (7).

$$\hat{Y}_i = \sum_{k=1}^{K} f_k(X_i) \tag{7}$$

Model Evaluation

To assess the performance of a multi-class classification model, a generalized confusion matrix, an extended version of the traditional confusion matrix, is used. As shown in Table 2 [18].

Table 2. Generalized Confusion Matrix

Actual (A)	Prediction (P)			
	Class 1	**Class 2**	**...**	**Class-k**
Class 1	X_{11}	X_{12}	...	X_{1k}
Class 2	X_{21}	X_{22}	...	X_{2k}
⋮	⋮	⋮		⋮
Class-k	X_{k1}	X_{1k}	...	X_{kk}

Diagonal elements (True Positives) indicate correct predictions for each class, where predicted and actual values match. Off-diagonal elements indicate incorrect predictions, with False Positives being the number of incorrect predictions in a column, and False Negatives being those in a row. The model's accuracy can be determined using the data derived from the confusion matrix. The form of Accuracy is shown in Equation (8).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{8}$$

However, since the data used in this study has a severe imbalance, accuracy alone is not sufficient to measure the performance of classification. To ensure the balanced performance assessment of the classification across all classes, including rare ones, another assessment method called the Macro Average and weighted average of the F1-score is used. This metric averages F1-scores calculated for each class independently, regardless of sample size. To calculate it, Precision, Recall, and F1-score are needed, as shown in equations (9), (10), and (11) [19].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{F1} - \text{score} = 2 \text{ x } \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

The Macro Average of the F1-score can be written in Equation (12) [19].

$$\text{Macro Average F1} - \text{score} = \frac{1}{K}\sum_{\text{i}=1}^{K} \text{F1} - \text{score} \tag{12}$$

K denoted the number of classes. The weighted average of the F1-score as shown in Equation (13) [20].

$$\text{Weighted Average F1} - \text{score} = \sum_{\text{i}=1}^{K} (\text{Weight x F1} - \text{score}) \quad (13)$$

To provide a fairer assessment of the model on imbalanced data, model evaluation is conducted using Balancded Accuracy. Balancded Accuracy treats each class equally, giving rare classes the same attention as the majority class [21]. Balancded Accuracy is shown in Equation (14).

$$\text{Balancded Accuracy} = \frac{1}{K} \sum_{\text{i}=1}^{K} \text{Recall} \quad (14)$$

Proposed Model

In this study, the data is split into an 80:20 ratio for training and testing. Two classification models are explored, which is a CNN for feature extraction, combined with XGBoost as the classifier. A comparison between the base CNN and the combined model is conducted, with the model achieving the highest accuracy being deemed the most effective for the classification task. The research process is outlined in Figure 5.

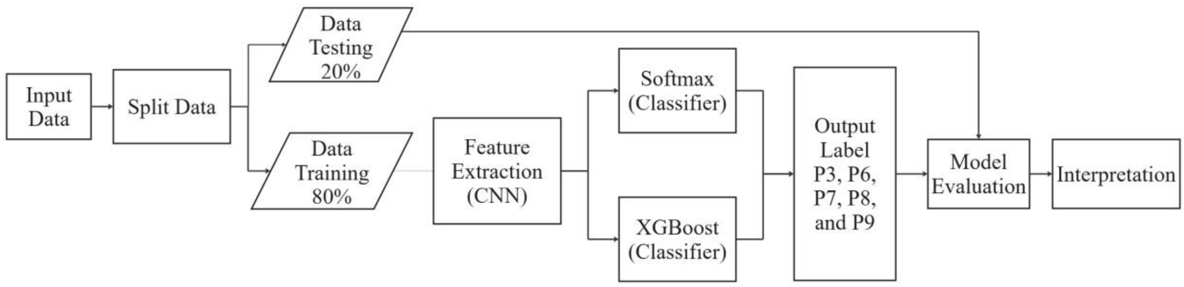


Figure 5. Research Steps

To optimize the CNN, the Adam optimizer and callbacks are applied to adjust the learning rate based on loss and accuracy, with a patience of 3 epochs before reducing the rate. Early stopping is employed with a patience of 7 epochs to prevent overfitting. For XGBoost, the maximum tree depth of 6 controls complexity, and a subsample ratio of 1 utilizes the full dataset. A learning rate of 0.3 ensures faster training, with hyperparameter choices informed by prior research, as shown in Table 3.

Table 3. Hyperparameter Tunning

Model	Hyper-parameter	Range
CNN	Learning rate (Adam Optimizer)	0.00025
	Epoch	100
	Batch size	32
	Patience (Callbacks)	5
	Patience (Early Stopping)	7
XGBoost	Learning rate	0.3
	Max tree depth	6
	Subsample ratio	1

RESULTS AND DISCUSSION

Imbalanced Data

To perform classification, training and testing sets are prepared by dividing data, as detailed in Figure 6.

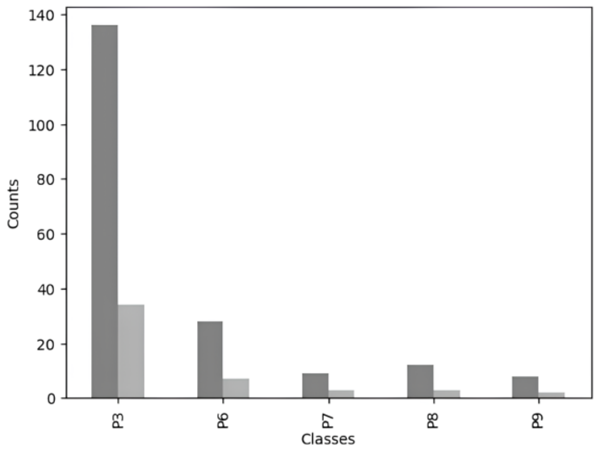


Figure 6. Data Distribution

Figure 6 illustrates the distribution of training dan testing datasets, highlighting the severe class imbalance. The largest class (P3 : Insect Pests) had 136 training samples and 34 testing samples, while the smallest clas (P9: Clubroot Disease) had only 8 training dan 2 testing samples. It is evident that the data is highly imbalanced. This imbalance necessitated the use of robust methods to ensure fair classification across all classes.

Extraction Feature

This study utilizes a standalone CNN model for feature extraction using the ConvNeXt-Tiny architecture. The process begins with Equation (1) as a simple neural network and applies convolution using filters described in Equation (2). This leads to the convolutional neural network process, resulting in important features extracted from the training data through Global Average Pooling. Descriptive statistics for the first five features are presented in Table 4.

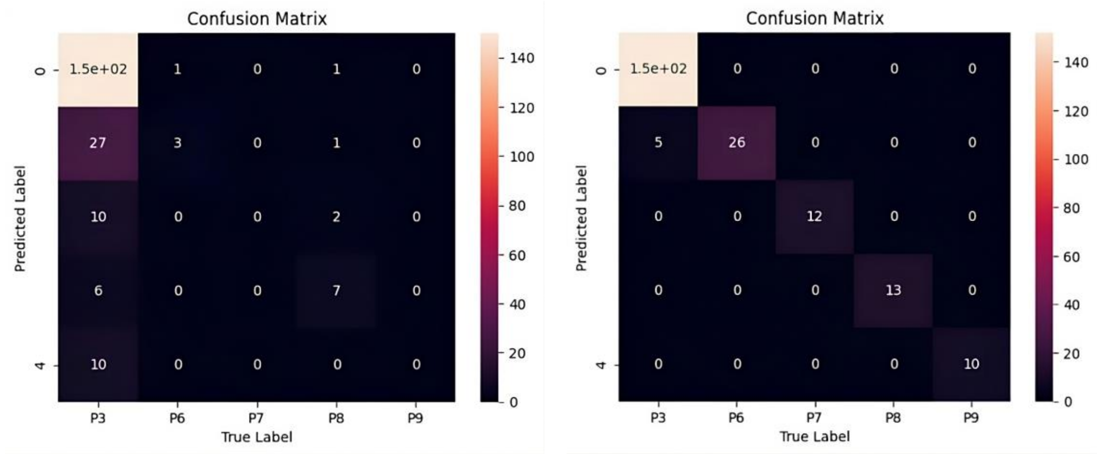
Table 4. Descriptive Statistics of Extracted Features

Feature	Minimun	Mean	Maximum	Variance
f_1	-0.227	0.115	2.105	0.165
f_2	-0.615	2.055	2.676	0.557
f_3	-0.710	0.941	1.341	0.218
f_4	-0.348	0.618	1.071	0.219
f_5	0.110	0.337	0.832	0.013

Classification Performance

The extracted features are then used in the classification process. For CNN, the classification is performed using the Softmax activation function as described in Equation (4), while XGBoost performs classification by combining decision trees with boosting as described in Equations (5) and (6). The model's performance is evaluated based on its accuracy in classifying the data, as shown in the Generalized Confusion Matrix.

The following presents the generalized confusion matrix, which illustrates the classification accuracy for each class. Two confusion matrix tables are provided, the first shows the classification accuracy of the standalone CNN model, while the second represents the Hybrid CNN-XGBoost model, allowing for a comparison of the models' performance in classifying imbalanced multiclass data. The results are shown in Figure 5.



(a) **(b)**

Figure 5. Confusion Matrix, (a) CNN, (b) CNN-XGBoost

In Figure 5, the confusion matrix color gradient on the main diagonal becomes progressively darker, indicating that for each class there is an increase in correctly predicted labels. Meanwhile, the number of prediction errors outside the main diagonal quickly decreases. Values outside the diagonal of the confusion matrix indicate misclassifications. The CNN model shows poorer performance compared to the CNN-XGBoost model, which only misclassifies 5 instances of P3. This means that the CNN-XGBoost model produces a better result, especially in handling imbalanced data.

The results from the Generalized Confusion Matrix are then used as input for evaluation using Precision, Recall, and F1-score for each class. The evaluation results for each class of the CNN model are shown in Table 5.

Table 5. CNN Model Evaluation Class

Model CNN	**Class**				
	P3	**P6**	**P7**	**P8**	**P9**
Precision	0.86	0.54	0.67	0.79	1.00
Recall	0.94	0.60	0.17	0.73	0.20
F1-score	0.90	0.57	0.27	0.76	0.33

As shown in Table 4, the class with balanced data (P3) achieved the highest accuracy in both positive and negative cases. On the other hand, imbalanced classes (P6 through P9) exhibited lower accuracy compared to P3. This indicates that the CNN model is not yet capable of handling imbalanced data effectively, necessitating further improvements using Hybrid techniques.

Next, classification was performed using the Hybrid CNN-XGBoost model. The model demonstrated significantly improved performance across all classes, as shown in Table 6.

Table 6. CNN-XGBoost Model Evaluation Class

Model CNN-XGBoost	Class				
	P3	P6	P7	P8	P9
Precision	0.96	0.97	1.00	1.00	1.00
Recall	0.99	0.86	0.83	1.00	1.00
F1-score	0.98	0.91	0.91	1.00	1.00

As shown in Table 6, there is an improvement in classification accuracy, especially for imbalanced data (P3 to P9), with the highest accuracy observed in the minority classes P8 and P9, reaching 1.00. This indicates that the XGBoost method with a maximum tree depth of 6 can enhance the sensitivity of imbalanced data in classification tasks.

In the case of imbalanced data, evaluating the model using accuracy alone is insufficient. Therefore, to provide a fairer assessment, additional metrics were considered. Macro Average of the F1-score, Weighted Average of the F1-score, and Balanced Accuracy were employed. The evaluation results are presented in Table 7.

Table 7. Comprehensive Model Evaluation Metric

Model	Accuracy	F1-Score	Macro Average F1-Score	Weighted Average F1-score	Balancded Accuracy
CNN	0.71	0.81	0.53	0.79	0.53
CNN-XGBoost	1.00	0.97	0.96	0.97	0.93

Table 7 shows the capability of the CNN-XGBoost classifier model to classify imbalanced data with multiple classes, with a particular focus on Balancded Accuracy, which treats each class equally. The result shows that the base CNN model's Balancded Accuracy is 0.53, which is 0.40 lower than the CNN-XGBoost model. This demonstrates that the CNN model struggles with classifying imbalanced data, as supported by previous studies. In contrast, the Hybrid CNN-XGBoost model significantly outperforms the CNN model, with the highest accuracy reported at 1.00 and a Balancded Accuracy of 0.93. This indicates that the CNN-XGBoost model handles extremely imbalanced data more effectively and provides superior classification results across five classes. The comparison clearly illustrates the enhanced performance of the CNN-XGBoost model over the CNN model in various aspects, highlighting its capability to manage complex and imbalanced datasets.

Discussion

The findings of this study clearly demonstrate the effectiveness of the Hybrid CNN-XGBoost model in addressing challenges posed by imbalanced multiclass data. The standalone CNN model struggled with minority classes, resulting in poor performance metrics such as low precision, recall, and F1-scores. In contrast, the Hybrid model achieved significant improvements, particularly for minority classes like P8 and P9, which reached perfect precision, recall, and F1-scores. This improvement can be attributed to the XGBoost component, which enhances classification by focusing on underrepresented data during training. Additionally, the use of metrics beyond accuracy, such as Macro Average F1-score, Weighted F1-score, and Balanced Accuracy, provided a more comprehensive evaluation. The Hybrid model excelled in these metrics, achieving a

Balanced Accuracy of 0.93 compared to 0.53 for the CNN. This highlights its ability to handle imbalanced datasets more effectively. The confusion matrix further supports this, showing fewer misclassifications and better prediction consistency across all classes for the Hybrid model.

CONCLUSION

This study demonstrates the limitations of a standalone CNN model in classifying imbalanced multiclass datasets and highlights the advantages of integrating CNN with XGBoost. The findings indicate that the single CNN-based model struggles to effectively classify minority data, while it performs better with majority classes. In contrast, combining CNN with the ensemble method XGBoost has proven to be a robust approach for addressing imbalanced data, aided by hyperparameter tuning during the classification process. XGBoost has shown strong performance in handling multiclass classification, as reflected by high accuracy across various evaluation metrics.

These results underscore the potential of the CNN-XGBoost approach for agricultural diagnostics, particularly in identifying pests and diseases in cabbage plants. Future studies could expand this model's application to larger datasets, additional crops, and other imbalanced classification challenges. Moreover, incorporating advanced CNN architectures, such as YOLO, or employing resampling techniques like B-SMOTE, may further enhance classification performance.

REFERENCES

[1] Prabaningrum, L., & Moekasan, T. K. (2020). Incidence and diversity of insect pests and their natural enemies in control threshold-based cabbage cultivation. AAB Bioflux, 12(1), 12–21. http://www.aab.bioflux.com.ro

[2] Wardhani, N. W. S., Lestantyo, P., & Rahmi, N. S. (2023). Decision support system as an element of webbased integrated pest control on cabbage plants. E3S Web of Conferences, 450. https://doi.org/10.1051/e3sconf/202345002001

[3] Reya, S. S., Malek, M. D. A., & Debnath, A. (2022). Deep Learning Approaches for Cabbage Disease Classification. 2022 International Conference on Recent Progresses in Science, Engineering and Technology (ICRPSET), 1–5. https://doi.org/10.1109/ICRPSET57982.2022.10188553

[4] Myna, A. N., Manasvi, K., Pavan, J. K., Rakshith, H. S., & Yukhta, D. J., (2023). Classification and Detection of Cabbage Leaf Diseases from Images Using Deep Learning Methods. Automation, Control and Intelligent Systems, 11(1), 1–7. https://doi.org/https://doi.org/10.11648/j.acis.20231101.11

[5] Skendžić, S., Zovko, M., Živković, I. P., Lešić, V., & Lemić, D. (2021). The impact of climate change on agricultural insect pests. In Insects (Vol. 12, Issue 5). https://doi.org/10.3390/insects12050440

[6] Dablain, D., Jacobson, K. N., Bellinger, C., Roberts, M., & Chawla, N. V. (2023). Understanding CNN Fragility When Learning with Imbalanced Data. Machine Learning, 1–26. https://doi.org/10.1007/s10994-023-06326-9

[7] Liu, L., Wu, X., Li, S., Li, Y., Tan, S., & Bai, Y. (2022). Solving the class imbalance problem using ensemble algorithm: application of screening for aortic dissection. BMC Medical Informatics and Decision Making, 22(1), 1–16. https://doi.org/10.1186/s12911-022-01821-w

[8] Velarde, G., Sudhir, A., Deshmane, S., Deshmunkh, A., Sharma, K., & Joshi, V. (2023). Evaluating XGBoost for Balanced and Imbalanced Data: Application to Fraud Detection.

http://arxiv.org/abs/2303.15218 Rahman, M., Prodhan, R., Shishir, Y., & Ripon, S. (2021). Analyzing and Evaluating Boosting-Based CNN Algorithms for Image Classification. 2021 International Conference on Intelligent Technologies (CONIT), 1–6. https://doi.org/10.1109/CONIT51480.2021.9498328

[9] Rahman, M., Prodhan, R., Shishir, Y., & Ripon, S. (2021). Analyzing and Evaluating Boosting-Based CNN Algorithms for Image Classification. 2021 International Conference on Intelligent Technologies (CONIT), 1–6. https://doi.org/10.1109/CONIT51480.2021.9498328

[10] Jiao, W., Hao, X., & Qin, C. (2021). The image classification method with cnn-xgboost model based on adaptive particle swarm optimization. Information (Switzerland), 12(4), 1–22. https://doi.org/10.3390/info12040156

[11] Gao, X., Jamil, N., Ramli, M. I., & Ariffin, S. M. Z. S. Z. (2024). A Comparative Analysis of Combination of CNN-Based Models with Ensemble Learning on Imbalanced Data. International Journal on Informatics Visualization, 8(1), 456–464. https://dx.doi.org/10.62527/joiv.8.1.2194

[12] Fleuret, F. (2023). The Little Book of Deep Learning (Vol. 1). University of Geneva. https://fleuret.org/public/lbdl.pdf

[13] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). Classification. In An Introduction to Statistical Learning: with Applications in R (pp. 129–195). Springer US. https://doi.org/10.1007/978-1-0716-1418-1_4

[14] Todi, A., Narula, N., Sharma, M., & Gupta, U. (2023). ConvNext: A Contemporary Architecture for Convolutional Neural Networks for Image Classification. Proceedings - 2023 3rd International Conference on Innovative Sustainable Computational Technologies, CISCT 2023. https://doi.org/10.1109/CISCT57197.2023.10351320

[15] Nguyen, A., Pham, K., Ngo, D., Ngo, T., & Pham, L. (2021). An Analysis of State-of-the-art Activation Functions For Supervised Deep Neural Network. 2021 International Conference on System Science and Engineering (ICSSE), 215–220. https://doi.org/10.1109/ICSSE52999.2021.9538437

[16] Kouretas, I., & Paliouras, V. (2019). Simplified Hardware Implementation of the Softmax Activation Function. 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), 1–4. https://doi.org/10.1109/MOCAST.2019.8741677

[17] Fadhlullah Kh.TQ, M., & Wahyono, W. (2024). Classification of Tuberculosis Based on Chest X-Ray Images for Imbalance Data using SMOTE. International Journal of Computing and Digital Systems, 15(1), 981–993. https://doi.org/10.12785/ijcds/160171

[18] De Diego, I. M., Redondo, A. R., Fernández, R. R., Navarro, J., & Moguerza, J. M. (2022). General Performance Score for classification problems. Applied Intelligence, 52(10), 12049–12063. https://doi.org/10.1007/s10489-021-03041-7

[19] Cullerne Bown, W. (2024). Sensitivity and Specificity versus Precision and Recall, and Related Dilemmas. Journal of Classification, 41(2), 402–426. https://doi.org/10.1007/s00357-024-09478-y

[20] Vujović, Ž. (2021). Classification Model Evaluation Metrics. International Journal of Advanced Computer Science and Applications, 12(6), 599–606. https://doi.org/10.14569/IJACSA.2021.0120670

[21] Byeon, H. (2021). Comparing the Balanced Accuracy of Deep Neural Network and Machine Learning for Predicting the Depressive Disorder of Multicultural Youth. International Journal of Advanced Computer Science and Applications, 12(6), 584–588. https://doi.org/10.14569/IJACSA.2021.0120668