BAREKENG: Journal of Mathematics and Its Applications
June 2024 Volume 18 Issue 2 Page 1237–1248
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol18iss2pp1237-1248

ENSEMBLE CNN WITH ADASYN FOR MULTICLASS CLASSIFICATION ON CABBAGE PESTS

Nabila Ayunda Sovia [1*], **Ni Wayan Surya Wardhani**[2]

[1,2]*Departement of Statistics, Faculty of Sciences, Universitas Brawijaya*
Jl. Veteran Malang, Jawa Timur ,65145, Indonesia

*Corresponding author's e-mail: * **nabilaayunda003@student.ub.ac.id***

ABSTRACT

Image classification is a complex process influenced by various factors, one of which is the amount of image data. In the context of cabbage pest classification, data often exhibits a significant class imbalance, where certain pests are more prevalent than others. This imbalance can pose challenges during model training and evaluation, potentially leading to biases in favor of the majority of pests and reduced accuracy in identifying and classifying the less common ones. This research aims to enhance the classification performance for multiclass data specific to cabbage pests. We propose an ensemble learning approach that combines Convolutional Neural Network (CNN), Support Vector Machine (SVM), and Bagging methods. To address the imbalance issue inherent in cabbage pest data, we employ the Adaptive Synthetic Sampling (ADASYN) resampling technique. The CNN acts as the primary image identifier and classifier for various cabbage pests. Subsequently, the CNN model is integrated into SVM and Bagging models to mitigate the challenges of imbalanced data in pest classification. The research outcomes demonstrate that our ensemble approach, in conjunction with the ADASYN resampling technique, achieves an impressive accuracy rate of 97%, signifying its potential for improved cabbage pest detection and classification.

Article History:

Received: 9th January 2024
Revised: 17th February 2024
Accepted: 20th April 2024

Keywords:

ADASYN;
Bagging;
Cabbage;
Convolutional Neural Network;
Support Vector Machine

How to cite this article:
N. A. Sovia and N. W. S. Wardhani., "ENSEMBLE CNN WITH ADASYN FOR MULTICLASS CLASSIFICATION ON CABBAGE PESTS," *BAREKENG: J. Math. & App.*, vol. 18, iss. 2, pp. 1237-1248, June, 2024.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

Cabbage (Brassica oleracea var. capita L.) plays a crucial role in Indonesia's agricultural landscape, sustaining the livelihoods of numerous farmers. Nevertheless, these agriculturalists confront an enduring challenge: the relentless onslaught of pests and diseases that imperil their crops and well-being [1]. According to Abidin, there are symptoms caused by pests and diseases that attack cabbage plants. These symptoms typically appear on cabbage leaves in the form of changes in the color or shape of the leaves [2]. To address this, early detection of pests attacking cabbage can be carried out by identifying the symptoms that appear on the outer parts of the cabbage [3]. However, the true complexity lies in addressing the diversity of diseases and pests affecting cabbage plants, necessitating a multiclass classification approach [2]. One effective means of classifying these disease symptoms involves employing advanced detection tools, such as Convolutional Neural Networks (CNN), on image data. In cabbage fields, the occurrence of severe diseases like rot and clubroot is relatively rare, whereas diseases instigated by insects, such as caterpillars, are frequently encountered. This scenario results in imbalanced data, where one class contains fewer samples (the minority) compared to the other classes that constitute the majority [4].

Previous studies in the field of image classification have primarily concentrated on binary classification tasks, such as distinguishing between crop and non-crop cabbage. For instance, in a study referenced as [4], an ensemble model was introduced, combining bagging and boosting techniques. The primary objective of this study was to differentiate between crop and non-crop cabbage categories. This research encountered the challenge of imbalanced data, comprising 105 images of cropped cabbage and just 25 images of non-cropped cabbage, ultimately achieving a stable accuracy rate of 89%. However, these previous studies predominantly centered around binary classification, which addressed only a fraction of the broader challenges associated with pests and diseases affecting cabbage plants.

The diverse nature of these issues necessitates a more nuanced approach in the form of multiclass classification. The unstructured quality of image data, complicated by the intricate manifestations of pests and diseases in cabbage plants, poses a formidable challenge. Building on the success of prior research, as exemplified by reference [5], which employed a hybrid method to integrate Support Vector Machine (SVM) and CNN for handwritten character recognition, our study recognizes the pivotal role of CNN.

In our research, we leverage the advantages of CNN's accuracy for disease and pest detection in cabbage images, as demonstrated in previous research as demonstrated in previous research [6] [7]. The challenge we confront is imbalanced data, as noted in research conducted by [8]. Despite the excellent results achieved by deep neural networks like CNN on imbalanced data, challenges persist in handling minority data. Prior research addressing CNN on imbalanced data has suggested the use of resampling techniques and the combination of methods to enhance classification performance on such data [8]. Moreover, previous studies [5] have adapted SVM to handle multiclass classification using methods like 'one vs one' and 'Directed Acyclic Graph' (DAG).

In our research, we use Multiclass SVM with the 'one vs one' technique as a high-dimensional feature extractor to identify pests and diseases in cabbage plants. What sets our research apart is the inclusion of hyperparameter optimization, particularly grid search [16], to fine-tune the process. Moreover, we embrace bagging as an ensemble technique, distinguishing it from its previous application in research [4]. While previous studies have used bagging as an ensemble technique, our innovative use of bootstrapped resampling enhances the stability and predictive accuracy of our models. This enhancement ensures the dependability of our results when classifying disease symptoms in affected cabbage plants. Employing bagging proves valuable in aiding both CNN and SVM models by mitigating data overfitting. It will effectively handle issues of variance and bias inherent in CNN [9].

These methods will be combined using ensemble methods and resampling techniques to address imbalanced data [8][9]. Each method has advantages and disadvantages in its application. In cases of imbalanced data, minority class data is more likely to be overlooked by CNN methods, which is why resampling methods are needed to address this issue [10]. ADASYN is a resampling technique used to generate synthetic data, and its advantage lies in its focus on the minority class, which can effectively address issues related to imbalanced data. [11].

In conclusion, our research aims to create a model that combines CNN, SVM, and bagging to classify cabbage plants afflicted by a range of pests and diseases. The unique aspect of our approach lies in the synergy of these methods, artfully addressing data imbalances and enhancing classification accuracy. We aspire to support Indonesian cabbage farmers by providing them with improved tools and strategies to protect their crops and livelihoods. Additionally, we employ a technique called ADASYN to handle data imbalances, building upon the successful methods used in prior studies. The primary distinction in our research lies in the ensemble model and the utilization of primary data, including images of cabbage plants afflicted by various pests.

2. RESEARCH METHODS

2.1 Data

The data used in this research is a cabbage image and the data was collected in Poncokusumo, Malang, with cabbage specimens ranging from 70 to 80 days old, approaching the harvest period. The cabbage specimens used were those affected by pests. These pests exhibit specific characteristics, but the pests to be used are chosen based on their visible external features. There are a total of 242 image data, and the images were randomly captured using a 24.2 MP EOS M100 mirrorless camera. Information regarding cabbage diseases and pests is based on the Abidin journal and research [2]. Five types of pests and diseases that are classified are presented in Table 1.

Table 1. Data Information

Type of Pests	Symbol	Data
Insect Pests	P3	170
Leaf spot disease	P6	35
Black rot disease	P7	12
Soft rot disease	P8	15
Culbroot disease	P9	10

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

Figure 1 (a) Insect Pests, (b) Leaf Spot Disease, (c) Black Rot Disease, (d) Soft Rot Disease, (e) Culbroot Disease.

The following images in Figure 1 illustrate the classification of various diseases and pests affecting cabbage. In Figure 1 (a), cabbage is shown infested with pests, exhibiting apparent health but displaying perforations on its outer leaves. Figure 1 (b) depicts cabbage afflicted with leaf spot disease, resulting in a discoloration of the outer leaves, turning them yellowish brown and causing wilting. In Figure 1 (c), symptoms of black or brown rot are evident, as seen in the cabbage head with brown spots and wilting, ultimately leading to the demise of the plant. Figure 1 (d) represents cabbage affected by soft rot disease, characterized by a blackened and decayed appearance of the cabbage head. Finally, Figure 1 (e) portrays cabbage suffering from root rot, where the absence of a well-formed head indicates the failure of plant growth. The comprehensive descriptions of each condition aim to facilitate the accurate classification and diagnosis of cabbage diseases and pests.

2.2 Image Preprocessing

Before inputting image data into the CNN to improve the quality of the image, noise and variation in the image must be eliminated, and it can be used as image normalization and enhance the accuracy of the processing step [12]. The steps consist of image enhancement and segmentation. First, remove the background to maximize the CNN's ability to identify the parts of the cabbage and form convolutional layers. Then, resize the image data size to make all image sizes uniform and expedite the process. Then, the image will be segmented. Image segmentation helps in identifying object boundaries, such as leaves that have different colors or leaves with holes, and it is used to extract regions of interest from an image [13].

2.3 Splitting Training and Testing Data

The data was split into three sets: training set, validation set, and testing set. The validation set is used as tool for fine tune the model hyperparameters, while the training set is used as tool for the model learning process and enables performance assessment, ultimately yielding the production of the optimal model, and this best performing model underwent evaluation using the testing dataset to gauge its overall performance quality. The partitioning of the train, validation, and test sets was 80:10:10.

2.4 Ensemble Models

Ensemble learning is one way to address the issue of imbalanced data in classification by combining several models to produce the best final model. Three different models are used in this research. The first model is CNN, the second model is SVM, and the third model is bagging.

2.4.1 Convolutional Neural Network (CNN)

This method serves as an automatic feature extractor, where the feature vector, containing the extracted features, is utilized as input for subsequent methods. The input image data is denoted by $x_i^{(k)} \in R^d$, where d represents the image size ($d = W\ x\ H$ = 256 x 256 pixels). The convolutional layer is composed of 2D convolutional kernels or filters, taking on the form of rectangles and characterized by matrices [14]. This research will be using the SqueezeNet architecture, which is explicitly crafted to enhance accuracy while utilizing notably lesser parameters. Notably, SqueezeNet is approximately 510 times smaller than AlexNet.

The structural design integrates 8 Convolutional layers, seamlessly incorporating the Fire Inception and Fire Squeeze blocks. Noteworthy is our modification, opting for 8 fire modules instead of the original 16 and substituting Max Pooling with leaky ReLU activation. This choice is made due to the advancements it brings in multiclass classification [15] The comprehensive SqueezeNet architecture encompasses convolutional layers utilizing the fire module, comprised of a squeeze layer (with singular 1x1 filters) and an expand layer (featuring a blend of 1x1 filters and 3x3 filters) [16]. The convolution operation applied to a feature map with multiple channels is succinctly expressed in Equation (1) [17].

$$x_i^{(k)} = x_{i-1}^{(c)} * w_i^{(c,k)} + b_i^{(k)} \tag{1}$$

Here, w is the weight applied in each layer $w \in R^d$, b is the bias, the variable i signifies the index assigned to the network layer, k is for kernels, and c denotes the channel. The Adam optimizer is employed to iteratively update network weights based on the training data [18]. After convolution, the activation function is applied. The Leaky ReLU activation function is expressed in Equation (2).

$$A_i^{(k)} = \max\left(0; x_i^{(c)}\right) \tag{2}$$

To prevent the risk of overfitting, the Softmax function will be employed. This function is primarily used in multiclass classification problems, serving as an extension of the previous logistic regression method, and is employed to normalize calculations, resulting in probability outputs [17].

2.4.2 Multiclass Support Vector Machine (SVM)

The original purpose of the SVM was binary classification, with the goal of delineating two classes within a provided dataset [19]. For addressing multiclass problems, we adopt the one-vs-one approach, employing features $\vec{x}_i \in R^n$,where $\vec{x}_i$ represents an input feature vector [17], and $y_i \in (1, \dots, m)$ is labels that correspond to every class. This will create binary classifiers for each possible class, and a grid search is utilized to optimize these classifiers. The one vs one method involves constructing classifiers of $m(m-1)/2$, where two classes selected from the m available classes are trained on each classifier. This approach enhances the SVM model's performance by leveraging the extracted features, ensuring robust and effective classification outcomes [20].

2.4.3 Bagging

Bagging (Bootstrap Aggregating) is a fundamental ensemble learning technique employed to address imbalanced data in classification problems [21]. It involves creating random training sample subsets through bootstrapping. These subsets serve as the training data for multiple models. In bagging, the final prediction is determined by aggregating the predictions from these models through a voting mechanism. Employing bagging proves valuable in aiding both CNN and SVM models by mitigating data overfitting. It effectively handles issues of variance and bias inherent in CNN [22].

2.4.4 Ensemble CNN, SVM, Bagging, and ADASYN

The following flowchart shows the process of ensemble methods in Figure 2. The process begins with input data that has already undergone preprocessing. It proceeds by utilizing a CNN model for feature extraction. The resulting features are then input into an SVM model, where a grid search is performed to fine-tune hyperparameters. To bolster the model's robustness and alleviate overfitting, the SVM model is integrated with other models using bagging. This integration ultimately yields a set of classifications based on these methods and are presented in Figure 2.

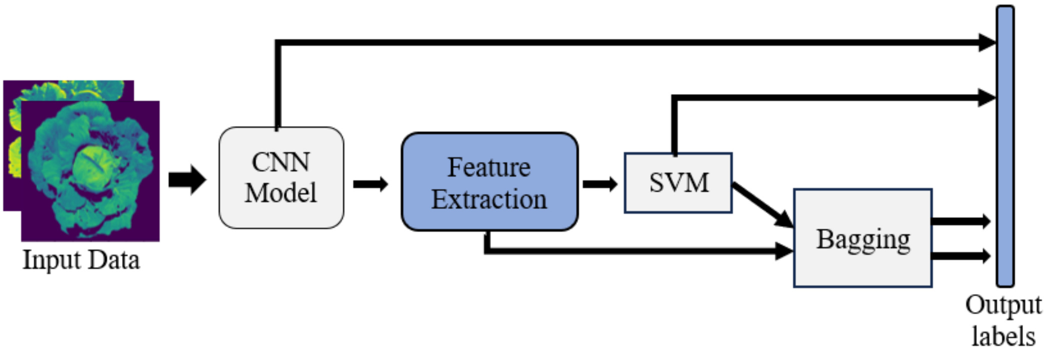


Figure 2. Flowchart of the Ensemble Model of CNN, SVM, and Bagging

2.5 Dealing with Imbalance Data

As previously noted, the dataset comprises 242 images distributed across five distinct class categories. To mitigate the data imbalance within each category, we are planning involves implementing the Adaptive Synthetic Resampling Technique (ADASYN). The primary aim of ADASYN is to address data imbalance by creating supplementary synthetic samples tailored specifically for the minority class, which often poses challenges to the learning process. This involves identifying minority samples that are particularly difficult to address, determined by calculating the number of nearest neighbors. Subsequently, ADASYN generates synthetic samples in a manner that approximates the challenging-to-learn regions based on the neighbor count [11].

2.6 Evaluation Metrics

In the testing phase, the assessment of the model efficacy relies on performance indicators like accuracy, F1 score, sensitivity, and specificity. These metrics serve as crucial measures for evaluating classifier performance, and their computation is derived from the information encapsulated in the confusion matrix [4].

Table 2 showcases the confusion matrix that represents the classification model performance. In this representation, TP (true positives), FN (false negatives), FP (false positives), and TN (true negatives) are employed to signify different aspects of the classification outcomes [23]. In the scenario of addressing a multi-class classification task, the confusion matrix provides a detailed examination of the model's performance by revealing how predicted and actual class distributions align for each class in the dataset. The confusion matrix is presented in Table 2.

Table 2. Confusion Matrix

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

Table 3 depicts of each performance metric linked to the model. Accuracy, calculated using Equation (3) with the confusion matrix, covers both accurate and erroneous classifications, providing a thorough assessment [14]. Within the domain of machine learning, the loss function acts as a quantifier for the model's efficacy on the training data. Equation (4) illustrates that it captures a certain level of information loss, thereby influencing the overall completeness of the outcome [14]. Sensitivity, defined as the test capability to accurately identify a cabbage as a pest class [24], is depicted in Equation (5), showcasing the probability of cabbages testing positive when signs of disease are present. Conversely, specificity, denoting the test ability to inaccurately categorize a cabbage as a pest class, is articulated in Equation (6), elucidating the probability of testing negative when pests are indeed absent [24]. Given the imbalanced nature of the dataset, the F1 score is computed to provide a balanced assessment. As delineated in Equation (8), the F1 score serves as a weighted average of precision and sensitivity (recall) [14]. Table 3 will present the equations for performance measurements.

Table 3. Performance Measures

Name	Representation	
Accuracy [25]	$\frac{TP+TN}{TP+FP+FN+TN}$	(3)
Loss [14]	$1 - Accuracy$	(4)
Sensitivity [25]	$\frac{TP}{TP+FN}$	(5)
Specificity [25]	$\frac{TN}{FP+TN}$	(6)
Precision [4]	$\frac{TP}{TP+FP}$	(7)
F1-score [4]	$2\ x\ \frac{Precision * Sensitivity}{Precision+Sensitivity}$	(8)

Validation graph acts as diagnostic tools, offering insights into whether iterative training enhances a model effectiveness. Graphically, this graph represents training and validation epochs along the horizontal axis, while the vertical axis is defined by quality metrics such as accuracy and loss, as outlined in Equations (3) and Equation (4).

3. RESULTS AND DISCUSSION

In this part, we introduce the experimental findings derived from our study, which include the application of various proposed methods for classifying cabbage pests, along with their respective performance measures. The stage of image preprocessing showcases the results of improving and segmenting image data. Following conventional machine learning practices, the data will then undergo the division into training and testing subsets, revealing the challenge of imbalanced data distribution. We address this imbalance using the resampling method ADASYN. Subsequently, we compare the results of each model when using the ADASYN resampling technique and when not using it. This comparison allows us to draw conclusions based on the effectiveness of the resampling technique.

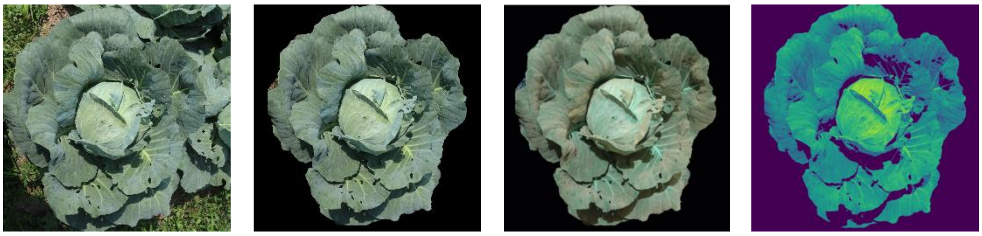
3.1 Image Pre-processing

The preprocessing of images before conducting classification using the CNN method is crucial to prepare optimal input data. The following outlines the image preprocessing process illustrated in **Figure 3**.

(1) Background Removal: As we can see in **Figure 3** (b) that the background of the cabbage is becoming black and the shape of the cabbage is clearly visible.

(2) Image Sharpening and Resizing: The data obtained from the camera has a very large size of 1600x1600 pixels. If the image is too large, the modeling process will take a long time. To make the process faster, the image data is then resized to 256x256 pixels with three channels.

(3) Segmentation: In this study, segmentation uses k-means based method to make the segmentation more detail and change the colors into RGB.



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

Figure 3 (a) Original Cabbage Image, (b) Cabbage Image after Background Removal, (c) Cabbage Image after sharpening and resizing, (d) Segmented Cabbage Image

3.2 Split into Training and Testing

Figure 4 illustrates the data distribution after it has been partitioned into training and testing subsets. Following this step, to create a validation set the training data will undergo additional partitioning. It becomes evident that the data displays an imbalance, with a significant difference in the amount of data between the minority classes and majority classes. In order to improve the model's performance, resampling is required.

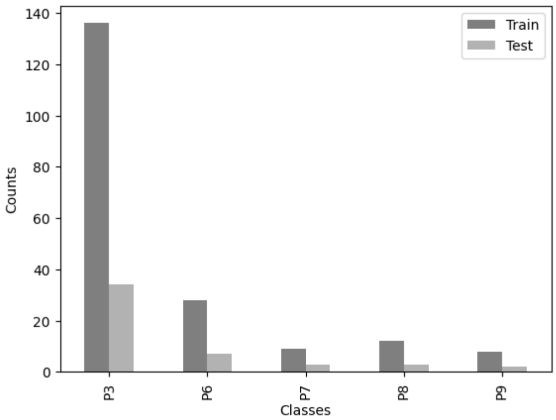


Figure 4. Training and Testing Data Class Distribution

3.3 Hyperparameter

Table 4 presents a variety of hyperparameter configurations. An ‘epoch’ represents the number of times the model undergoes training on the complete dataset. The ‘batch size’ indicates the number of training examples used in each iteration, with common choices falling within the range of $B \in \{32, 64, 128, 256$ In the context of imbalanced data, it is often more advantageous to select a smaller batch size. The 'learning rate' governs the step size at each iteration of the optimization algorithm. Adam as ‘optimizer’ will adjust the learning rates for each parameter individually. The ‘neighbor parameter’ is set to four, indicating that ADASYN selects four minority samples from each neighborhood for the generation of new samples.

Table 4. Hyperparameter Settings

(Hyper) parameter	Settings
Epochs	100
Batch Size	32
Learning rate	$2.5e^{-5}$; $1e^{-5}$; $1e^{-5}$
Optimizer	Adam
Neighbor for ADASYN	4

Table 5 displays the hyperparameters employed for SVM and grid search. These hyperparameters include the 'C' value, where a higher value aims to minimize training data misclassification, while a lower C value aims to achieve a smoother classification by allowing for more margin errors. The 'kernels' are functions that execute a conversion from a space with low dimensions, namely the input space or feature space, to a space with higher dimensions. Finally, the parameter 'gamma' holds significance within the framework of SVM, particularly for the RBF kernel, which defines the extent to which a single training example's influence extends.

Table 5. Grid Search Space

(Hyper) parameter	Settings
C	[10; 100; 1,000]
Kernels	[linear]; [rbf]
Gamma	[0.001; 0.0001

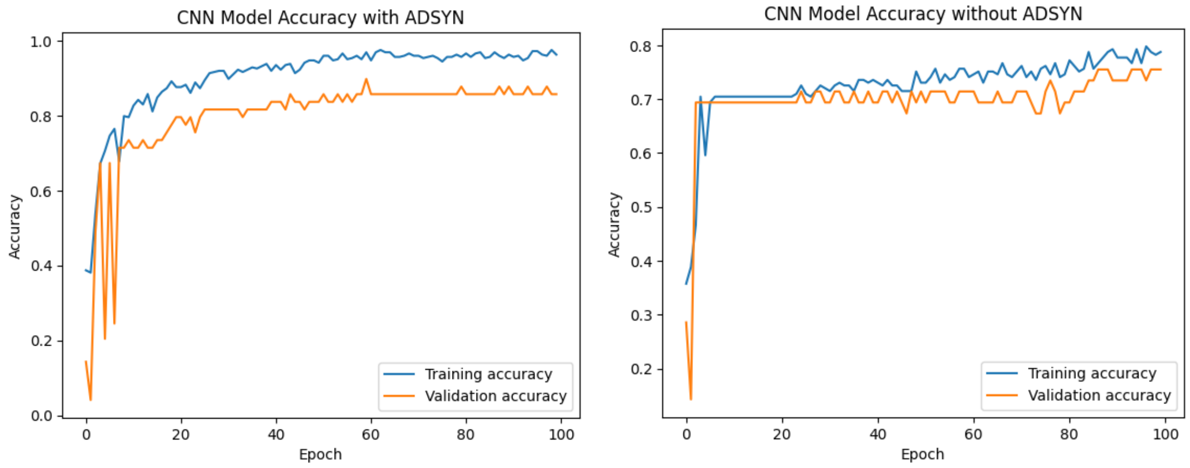
The structure of the CNN architecture significantly influences the model's accuracy, as illustrated in Table 6. The SqueezeNet architecture was applied to process images with dimensions of 256 x 256 pixels and three channels (RGB - Red Green Blue). In the given context, the "-" symbol is used to indicate that no specific or relevant value needs to be computed for a particular attribute.

Table 6. SqueezeNet Architecture

Layer Type	Output Size	Kernel Size/Stride	Number of Filters
Input	256 x 256 x 3	-	-
Convolution	128 x 128 x 96	7x7, stride 2	96
Leaky ReLU	64 x 64 x 96	3x3, stride 2	-
Fire Module (1)	64 x 64 x 128	1x1 (squeeze); 1x1 (expand); 1x1 (expand)	16; 64; 64
Fire Module (2)	64 x 64 x 128	1x1 (squeeze); 1x1 (expand); 1x1 (expand)	16; 64; 64
Leaky ReLU	32 x 32 x 128	3x3, stride 2	-
Fire Module (3)	32 x 32 x 256	1x1 (squeeze); 1x1 (expand); 1x1 (expand)	32; 128; 128
Fire Module (4)	32 x 32 x 256	1x1 (squeeze); 1x1 (expand); 1x1 (expand)	32; 128; 128
Leaky ReLU	16 x 16 x 256	3x3, stride 2	-
Fire Module (5)	16 x 16 x 384	1x1 (squeeze); 1x1 (expand); 1x1 (expand)	48; 192; 192
Fire Module (6)	16 x 16 x 384	1x1 (squeeze); 1x1 (expand); 1x1 (expand)	48; 192; 192
Fire Module (7)	16 x 16 x 512	1x1 (squeeze); 1x1 (expand); 1x1 (expand)	64; 256; 256
Fire Module (8)	16 x 16 x 512	1x1 (squeeze); 1x1 (expand); 1x1 (expand)	64; 256; 256
Convolution	16 x 16 x 1,000	1x1	1,000
Global Average Pooling	1x1x1,000	-	-
Softmax	1x1x1,000	-	-

3.4 Result

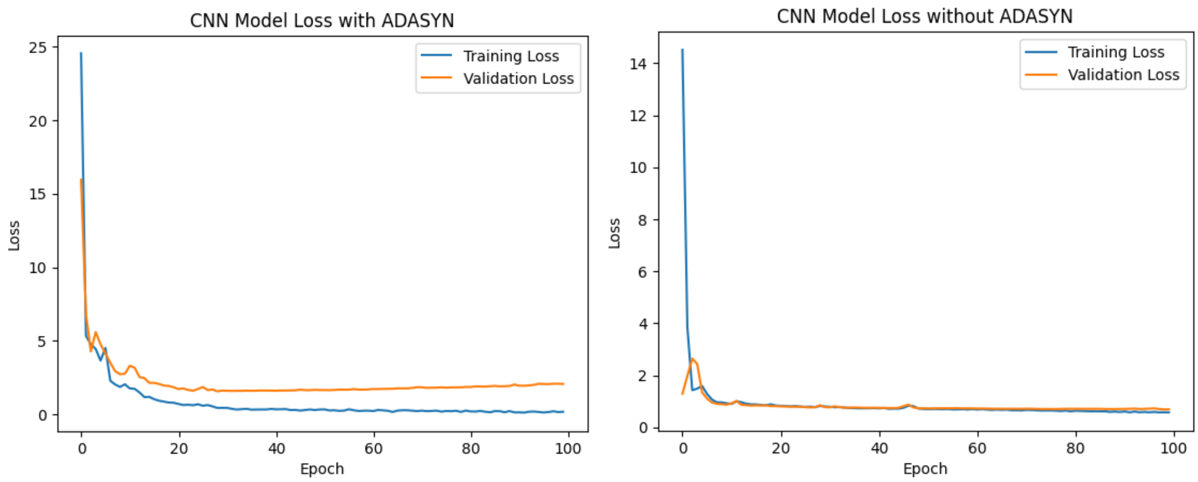
To assess the outcomes, the graph representing training and validation for the CNN model are depicted in Figure 5 and Figure 6.



(a) **(b)**

Figure 5. (a) Accuracy Graph of CNN with ADASYN, (b) Accuracy Graph of CNN without ADASYN

Figure 5 shows the accuracy graph for both training and validation, along with the corresponding loss graph. These graphs are compared between the scenarios with and without the use of ADASYN. It is obvious that the two models are generalized, but both validation accuracy is below training accuracy, which means that there is a possibility of overfitting. Accuracy with ADASYN seems more stable than without ADASYN, which means that ADASYN has helped to enhance the stability of the model's performance. The utilization of ADASYN results in a more consistent accuracy curve. This implies that the model is better equipped to generalize to unseen data and is less susceptible to overfitting. **Figure 6** illustrates the loss graph for both the model with ADASYN and the model without ADASYN. The pattern reveals that when the validation loss curve is below the training loss curve, overfitting occurs. However, similar to the accuracy curve, the loss curve is more stable when ADASYN is employed compared to when it is not.



(a) **(b)**

Figure 6. (a) Loss Graph of CNN with ADASYN, (b) Loss Graph of CNN without ADASYN

Table 6. Table Loss

Method	Loss
CNN with ADASYN	0.524
CNN without ADASYN	0.616

From **Table 6**, we can interpret that the CNN model with ADASYN has a lower loss value compared to the CNN model without ADASYN. A reduced loss value signifies enhanced performance, signifying the model's increased efficiency in minimizing the disparity between predicted and actual values.

Therefore, the CNN model with ADASYN is likely to be more accurate and reliable in its predictions than the CNN model without ADASYN.

Table 7. Comparison Accuracy of Training and Validation

	Method	Training	Validation
with ADASYN	CNN	0.71	0.71
	CNN + SVM	0.1	0.90
	CNN + Bagging	0.99	0.91
	CNN + SVM+ Bagging	0.89	0.75
without ADASYN	CNN	0.75	0.75
	CNN + SVM	0.99	0.87
	CNN + Bagging	0.98	0.85
	CNN + SVM + Bagging	0.89	0.82

Table 7 presents a comparison of accuracy between the training and validation datasets. The results reveal that during the training phase, every model exhibited a notable capability in effectively categorizing almost all classes, with the CNN + SVM ensemble model incorporating ADASYN achieving the highest accuracy at 0.99, alongside a validation accuracy of 0.91. Conversely, the CNN model with ADASYN exhibited the lowest accuracy. Notably, the CNN model displayed similar training and validation accuracies in both ADASYN and non-ADASYN cases. However, the ensemble models showcased higher training accuracies and validation accuracies with ADASYN, except for Ensemble CNN + SVM + Bagging, which exhibited slightly lower validation accuracies in both scenarios, with and without ADASYN. This suggests that while the ADASYN technique contributes to enhancing the capacity of the model to adapt and acquire knowledge from data imbalances during the training process., it may not necessarily result in improved generalization to unseen data.

For further measurements, the performance is then assessed with the proposed method. **Table 8** shows the accuracy, F1 score, sensitivity (recall) and precision. The result of ensemble shows a higher result when using ADASYN than without ADASYN. With the highest result in precision CNN + SVM + bagging with ADASYN with 99% and accuracy of 97%. The method using ADASYN gives improvements in ensemble, whereas the method without ADASYN has a high result but is not stable in CNN + SVM. Comparing the results with and without ADASYN, we can see that ADASYN has helped to improve the sensitivity of the CNN model from 0,87 to 0,96, indicating that the model is better at detecting the minority class. The F1 scores for the methods with and without ADASYN are comparable, with some methods without ADASYN even outperforming their counterparts. The methods with ADASYN have higher recall values, indicating better performance in correctly identifying positive cases. This is again due to the balancing effect of ADASYN on the imbalanced datasets. The methods without ADASYN have slightly higher precision values, indicating better performance in correctly identifying negative cases. This could be because the methods without ADASYN are not influenced by the synthetic samples generated by ADASYN, which might introduce some noise in the classification process. The approaches employing CNN + SVM showed a decline in performance or remained relatively constant across all metrics. This implies that the ensemble model might be a more effective means of enhancing the performance of the CNN model and Bagging in comparison to the SVM-based models.

Table 8. Comparison of Accuracy, F1 Score, Sensitivity and Specificity

	Method	Accuracy	F1	Recall	Precision
with ADASYN	CNN	0.88	0.85	0.75	0.95
	CNN+SVM	0.97	0.90	0.94	0.98
	CNN + Bagging	0.98	0.98	0.93	0.98
	CNN + SVM + Bagging	0.97	0.81	0.94	0.99
without ADASYN	CNN	0.86	0.78	0.15	0.1
	CNN + SVM	0.96	0.88	0.80	0.1
	CNN + Bagging	0.95	0.96	0.80	0.98
	CNN + SVM + Bagging	0.95	0.87	0.80	0.98

4. CONCLUSIONS

This study highlights the potential of the ADASYN technique to enhance the performance of ensemble CNN with SVM and bagging methods in handling imbalanced datasets. We explored two ensemble methods, one with ADASYN and one without. Our findings show that integrating ADASYN substantially improves stability and generalization, leading to consistently high results in evaluation metrics. This is evident in the results without ADASYN which show values that are lower overall than the results obtained when using ADASYN. For instance, the highest accuracy achieved with ADASYN is in the ensemble CNN and Bagging, reaching 98%, whereas without ADASYN, it is in the ensemble CNN and SVM at 96%. The results obtained using ensemble methods demonstrate higher accuracy, F1 score, recall, and precision compared to using CNN alone, both in cases with ADASYN and without ADSYN. Overall, ensemble methods, particularly when combined with ensemble CNN models, offer better stability in performance. The ensemble CNN model, especially when integrated with ADASYN, achieves the highest accuracy and stability, aligning with previous studies.

In the future, refining and optimizing the ADASYN integration to address potential overfitting and noise issues should be a focus. Exploring strategies to fine-tune ADASYN parameters or incorporating additional data preprocessing techniques can help strike a better balance between mitigating class imbalance and preserving data integrity. Collaboration with the agricultural sector and considering faster processing methods like YOLO could enhance the efficiency of plant disease identification in cabbage crops.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Brawijaya University for funding our research, which has enabled us to conduct this study and contribute to the advancement of knowledge in our field.

REFERENCES

[1] E. S. M. Embaby and D. E. S. Lotfy, "Ecological studies on Cabbage pests," *Int. J. Agric. Technol.*, vol. 11, no. 5, pp. 1145–1160, 2015.
[2] F. M. Abidin, H. Sibyan, and N. Hasanah, "Sistem Pakar Identifikasi Penyakit Sayuran Kubis Menggunakan Metode Foreward Chaining," *J. Eng. Inform.*, vol. 1, no. 1, pp. 14–21, 2022, doi: 10.56854/jei.v1i1.14.
[3] M. Waleed, A. S. Abdullah, and S. R. Ahmed, "Classification of Vegetative pests for cucumber plants using artificial neural networks," *2020 3rd Int. Conf. Eng. Technol. its Appl. IICETA 2020*, pp. 47–51, 2020, doi: 10.1109/IICETA50496.2020.9318890.
[4] N. W. S. Wardhani, M. Y. Rochayani, A. Iriany, A. D. Sulistyono, and P. Lestantyo, "Cross-validation Metrics for Evaluating Classification Performance on Imbalanced Data," *2019 Int. Conf. Comput. Control. Informatics its Appl. Emerg. Trends Big Data Artif. Intell. IC3INA 2019*, pp. 14–18, 2019, doi: 10.1109/IC3INA48034.2019.8949568.
[5] X. X. Niu and C. Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," *Pattern Recognit.*, vol. 45, no. 4, pp. 1318–1325, 2012, doi: 10.1016/j.patcog.2011.09.021.
[6] F. Deng, W. Mao, Z. Zeng, H. Zeng, and B. Wei, "Multiple Diseases and Pests Detection Based on Federated Learning and Improved Faster R-CNN," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, doi: 10.1109/TIM.2022.3201937.
[7] B. Jayanthi, T. A. Priyanka, V. B. Shalini, and R. K. Grace, "Pest Detection in Crops Using Deep Neural Networks," *8th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2022*, pp. 29–32, 2022, doi: 10.1109/ICACCS54159.2022.9785155.
[8] L. Liu, X. Wu, S. Li, Y. Li, S. Tan, and Y. Bai, "Solving the class imbalance problem using ensemble algorithm: application of screening for aortic dissection," *BMC Med. Inform. Decis. Mak.*, vol. 22, no. 1, pp. 1–16, 2022, doi: 10.1186/s12911-022-01821-w.
[9] J. Brandt and E. Lanzén, "A Comparative Review of SMOTE and ADASYN in Imbalanced Data Classification," Uppsala Universitet, 2020.
[10] D. Dablain, K. N. Jacobson, C. Bellinger, M. Roberts, and N. V. Chawla, "Understanding CNN Fragility When Learning With Imbalanced Data," *Mach. Learn.*, no. 0123456789, 2023, doi: 10.1007/s10994-023-06326-9.
[11] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *Proc. Int. Jt. Conf. Neural Networks*, no. 3, pp. 1322–1328, 2008, doi: 10.1109/IJCNN.2008.4633969.
[12] R. Sarki, K. Ahmed, H. Wang, Y. Zhang, J. Ma, and K. Wang, "Image Preprocessing in Classification and Identification of Diabetic Eye Diseases," *Data Sci. Eng.*, vol. 6, no. 4, pp. 455–471, 2021, doi: 10.1007/s41019-021-00167-z.
[13] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm," *Procedia Comput. Sci.*, vol. 54, pp. 764–771, 2015, doi: 10.1016/j.procs.2015.06.090.
[14] H. A. H. Al-Najjar *et al.*, "Land cover classification from fused DSM and UAV images using convolutional neural

networks," *J. Remote Sens.*, vol. 11, no. 12, 2019, doi: 10.3390/rs11121461.

[15] A. Maniatopoulos and N. Mitianoudis, "Learnable Leaky ReLU (LeLeLU): An Alternative Accuracy-Optimized Activation Function," *J. Inf.*, vol. 12, no. 12, 2021, doi: 10.3390/info12120513.

[16] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," in *conference paper at ICLR 2017*, 2016, pp. 1–13. [Online]. Available: http://arxiv.org/abs/1602.07360

[17] W. Gong *et al.*, "A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion," *J. Sensors*, vol. 19, no. 7, 2019, doi: 10.3390/s19071693.

[18] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[19] D. Keerthana, V. Venugopal, M. K. Nath, and M. Mishra, "Hybrid convolutional neural networks with SVM classifier for classification of skin cancer," *Biomed. Eng. Adv.*, vol. 5, no. November 2022, p. 100069, 2023, doi: 10.1016/j.bea.2022.100069.

[20] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning (2nd ed.)*, vol. 102. Springer Text in Statistics, 2021. [Online]. Available: https://www.statlearning.com/

[21] L. Yu and N. Zhou, "Survey of Imbalanced Data Methodologies," 2021, [Online]. Available: http://arxiv.org/abs/2104.02240

[22] G. Ngo, R. Beard, and R. Chandra, "Evolutionary bagging for ensemble learning," *Neurocomputing*, vol. 510, pp. 1–14, 2022, doi: 10.1016/j.neucom.2022.08.055.

[23] J. Nayak, K. Vakula, P. Dinesh, B. Naik, and D. Pelusi, "Intelligent food processing: Journey from artificial neural network to deep learning," *Comput. Sci. Rev.*, vol. 38, p. 100297, 2020, doi: 10.1016/j.cosrev.2020.100297.

[24] Y. Skaik, "Understanding and using sensitivity, specificity and predictive values," *Indian J. Ophthalmol.*, vol. 56, no. 4, p. 341, 2008, doi: 10.4103/0301-4738.41424.

[25] Z. Huang, X. Zhu, M. Ding, and X. Zhang, "Medical Image Classification Using a Light-Weighted Hybrid Neural Network Based on PCANet and DenseNet," *IEEE Access*, vol. 8, pp. 24697–24712, 2020, doi: 10.1109/ACCESS.2020.2971225.