CAUCHY – Jurnal Matematika Murni dan Aplikasi
Volume 11 (2) (2026), Pages 1527–1543
p-ISSN: 2086-0382; e-ISSN: 2477-3344

DeepSMOTE for Tea Leaf Disease Classification Using DenseNet-121 Transfer Learning

Nikita Putri Insani*, Sugiyarto Surono, and Aris Thobirin

Department of Mathematics, Faculty of Applied Science & Technology, Ahmad Dahlan University, Yogyakarta, Indonesia

Abstract

Data imbalance is a common problem in image classification that causes models to tend to learn majority class patterns more effectively than minority classes. This study evaluates DeepSMOTE for balancing tea leaf disease images in DenseNet-121 transfer learning classification. The dataset consisted of 885 tea leaf images from eight classes and was divided into training, validation, and testing sets using stratified splitting. Balancing and augmentation were applied only to the training data. DeepSMOTE was implemented using an autoencoder that generated synthetic images through interpolation in latent space. Its performance was compared with no augmentation, conventional augmentation, and feature-space SMOTE using five repeated experiments. The results showed that DeepSMOTE achieved the highest average test accuracy of 95.73 ± 2.50% among the evaluated methods. The results suggest that latent-space interpolation may have contributed to improved minority-class representation and classification performance. However, several classes still produced misclassifications due to visual similarities between diseases. Since this study used a single dataset with a relatively limited number of test images, the results should be interpreted carefully. Nevertheless, DeepSMOTE may be a promising approach for handling class imbalance in tea leaf disease classification.

Keywords: Class Imbalance; DeepSMOTE; DenseNet-121; Tea Leaf Disease Classification; Transfer Learning.

1. Introduction

The development of visual analysis technology has encouraged the emergence of approaches that are able to mimic the way humans process images, leading to the development of deep learning architectures such as Convolutional Neural Network (CNN). CNN has been shown to be able to understand and classify imagery quite well. However, in conventional CNN architectures, problems such as vanishing gradient and the need for multiple layers to achieve optimal performance are still common [1].

Among the various CNN architectures that exist, DenseNet-121 is one way to solve this problem. This architecture connects each layer with all the previous layers through dense feed-forward connections [2]. This concept keeps the gradient flow stable during training and helps prevent the vanishing gradient problem commonly found in deep neural networks [3]. In addition,

*Corresponding author. E-mail: 2200015031@webmail.uad.ac.id

Received: April 1, 2026 Revised: May 8, 2026 Accepted: June 24, 2026
DOI: https://doi.org/10.18860/cauchy.v11i2.41693

this architecture allows layers to make more effective use of information and supports feature reuse, so that the flow of information in the network becomes more efficient [4]. This advantage makes DenseNet-121 more parameter-efficient than other architectures and enables it to produce more informative feature representations, accelerate model convergence, and maintain training stability on complex datasets [3, 4]. The advantages of DenseNet-121 have also been proven in research on rice leaf diseases classification. The tests conducted showed that DenseNet-121 achieved the highest accuracy of 91.67%, followed by DenseNet-169 with 90%, and DenseNet-201 with 88.33% [5]. This strengthens the reason for choosing DenseNet-121 in this study, because this architecture has been proven to be more accurate than other DenseNet variants.

The performance of the DenseNet-121 architecture can be further improved through transfer learning. Transfer learning utilizes the weights and features of pre-trained models on large datasets to be reused on new and smaller datasets, making the training process more efficient, reducing computational costs, and improving model performance [6]. This approach also expands the model's generalization capability, as visual features learned from large datasets can help identify similar patterns across different datasets even when the domains are not identical [7]. The application transfer learning with DenseNet-121 has been shown to produce high accuracy in medical image processing for brain tumor classification, reaching 96.90% [8].

However, the application of transfer learning with DenseNet-121 still faces challenges, especially when dealing with class imbalance problems. The model has been shown to provide more optimal results on balanced datasets, indicating that balancing datasets plays an important role in improving classification performance [9]. To overcome this problem, one of the oversampling methods that can be used is Synthetic Minority Oversampling Technique (SMOTE). This method produces more diverse synthetic samples than several conventional oversampling techniques, thus improving the model's generalization capability and proving effective on small and medium-sized datasets [10, 11]. A number of studies have shown that the combination of SMOTE and deep learning can improve classification accuracy, such as in breast cancer classification with CNN, which increased from 88.7% to 97.84%, COVID-19 image classification based on DenseNet-201, which increased from 97.9% to 98.64%, and brain tumor classification with DenseNet-201, which increased from 99.51% to 99.65% [12–14].

Conventional SMOTE has become a commonly used oversampling method. Therefore, the use of DeepSMOTE is important as an alternative approach for handling image imbalance problems. DeepSMOTE combines an encoder–decoder mechanism with SMOTE-based oversampling in latent space, so that the resulting synthetic samples not only increase the number of minority class samples but also retain their important feature characteristics [15]. This approach has been shown to produce high-quality synthetic images that are more representative than those generated by traditional oversampling methods [15]. In addition, latent space-based oversampling approaches such as DeepSMOTE have the potential to produce more stable synthetic representations because the generation process is performed after the data are mapped into more structured latent features, thereby helping the model better recognize minority patterns under imbalanced data conditions [15, 16].

The problem of class imbalance does not only occur in medical images. It also occurs in plant diseases images, particularly leaf images which are often the main object in plant disease classification. The health condition of the leaves plays an important role in determining the yield and quality of plantation commodities, including tea. Tea plants are susceptible to various diseases that can lead to a significant decrease in yield and production quality [17]. As one of Indonesia's leading commodities, tea contributes greatly to the country's foreign exchange and labor absorption [18]. The impact of diseases on tea plants not only decreases productivity, but also has the potential to disrupt the economic stability of producing regions. Previous research showed that CNN was able to classify tea leaf diseases with high accuracy of up to 96.65% in cases in Bangladesh, but this approach has not yet applied data balancing techniques so its performance still has the potential to be biased against the majority class [19].

Based on this description, this study focuses on the implementation of DeepSMOTE in tea leaf disease classification using a transfer learning approach with DenseNet-121. The novelty of this study lies in the integration of DeepSMOTE with DenseNet-121 transfer learning to address class imbalance in tea leaf disease datasets. This study aims to analyze whether DeepSMOTE can improve the classification performance and stability of the model on imbalanced data. In addition, this approach is expected to improve the model's ability to recognize minority classes without reducing the overall prediction accuracy. Thus, this research is expected to contribute to the application of deep learning in the agricultural sector, especially in tea commodities in Indonesia which have high economic value.

2. Methods

This study was conducted to analyze the influence of data balancing techniques on the performance of tea leaf disease classification using DenseNet-121 transfer learning. The main approach used in this study is DeepSMOTE, while no augmentation, SMOTE, and conventional augmentation are used as comparison approaches to objectively assess the effectiveness of DeepSMOTE. To ensure reproducibility, all experiments were conducted using a fixed random seed of 42.

2.1. Research Flow

This section presents the stages of research consisting of dataset preparation, pre-processing, data augmentation, model development and training, and evaluation. These stages are displayed in the research flow so that the entire process can be understood clearly and systematically.

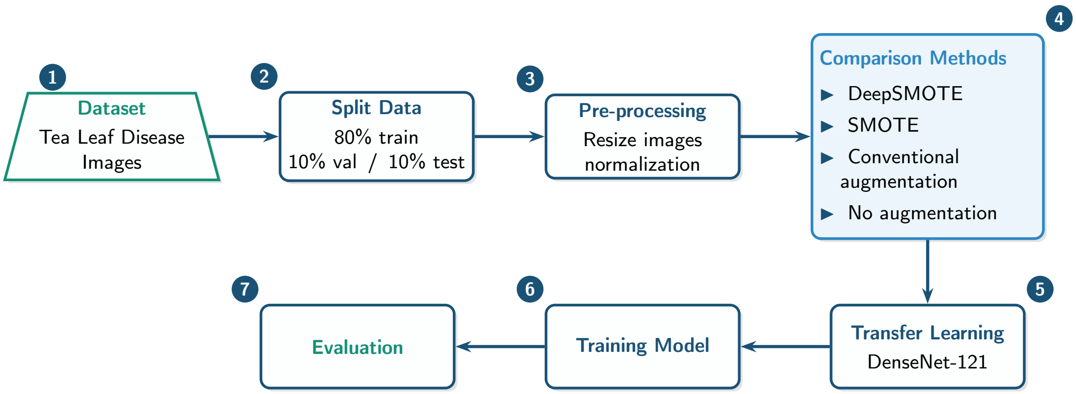


Fig. 1: Research Flow

The experiments were conducted using Google Colab with NVIDIA Tesla T4 GPU acceleration. The implementation was developed in Python using PyTorch, torchvision, scikit-learn, and imbalanced-learn libraries. The library versions used in this study included PyTorch 2.6, torchvision 0.21, scikit-learn 1.6, and imbalanced-learn 0.13.

2.2. Dataset

The dataset used was Identifying Disease in Tea Leaves from Kaggle[1], with a total of 885 images of tea leaves. The data were stratified into 80% training, 10% validation, and 10% testing sets. The split dataset was generated once using a fixed random seed and was kept unchanged throughout the entire study, including all repeated experiments. The initial distribution of the dataset is shown in Table 1.

Table 1 shows that the dataset has an uneven distribution, with the number of images per class ranging from 74 to 143 images. The Healthy class has the least amount of data, while the White Spot is the class with the most data. This imbalance has the potential to cause models to

[1] https://www.kaggle.com/datasets/shashwatwork/identifying-disease-in-tea-leafs

Table 1: Initial distribution of datasets prior to DeepSMOTE deployment

Class	Amount of data	Training data	Validation data	Test data
Anthracnose	100	80	10	10
Algal Leaf	113	91	11	11
Bird Eye Spot	100	80	10	10
Brown Blight	113	90	11	12
Gray Light	100	80	10	10
Healthy	74	59	8	7
Red Leaf Spot	143	114	14	15
White Spot	142	114	14	14
Sum	**885**	**708**	**88**	**89**

be more likely to recognize patterns from the majority class and ignore the minority class. To maintain the objectivity of the model evaluation, augmentation is applied only to the training data, while the validation and testing data remain using the original image without modification. For the implementation of DeepSMOTE and SMOTE, the number of images per class is set to have a target of 114 images, which is adjusted to the amount of training data in the majority class so that the data distribution is balanced.

2.3. Pre-processing

Preprocessing was carried out to adjust the image format to the needs of two models, namely the autoencoder at the DeepSMOTE stage and DenseNet-121 at the classification stage. All images were first converted to RGB format to ensure consistent three channel input before being processed further. For DenseNet-121, the image was resized to 224 × 224 pixels and normalized using the mean and standard deviation of ImageNet. Mathematically normalization is defined by Eq. (1).

$$z = \frac{X - \mu}{\sigma} \tag{1}$$

where X is the vector of the original feature, μ is the mean, and σ is the standard [20]. This normalization was applied because DenseNet-121 uses ImageNet1K pre-trained weights, so the input distribution needs to follow the ImageNet normalization standard.

Meanwhile, in the DeepSMOTE stage, the images were processed differently because the autoencoder works more stably at the original intensity range of 0-1. Therefore, the images were only resized to 128 × 128 pixels and converted into tensors without ImageNet normalization.

2.4. Data Augmentation

Data augmentation was applied only to the training data and implemented using three different approaches, namely DeepSMOTE, SMOTE, and conventional augmentation. Each approach was trained as a separate model.

2.4.1. DeepSMOTE

In this study, DeepSMOTE was used as a method oversampling to generate synthetic samples for minority classes through interpolation in the latent space of an autoencoder [21]. The implementation follows an adversarial autoencoder-based DeepSMOTE approach, where the autoencoder was trained in advance to learn compressed image representations using only the training dataset. The encoder architecture consists of four convolutional layers with kernel size 4×4, stride 2, and padding 1, followed by Batch Normalization and LeakyReLU activation. The latent representation generated by the encoder has a dimension of 256. Meanwhile, the decoder uses four transposed convolution layers with Batch Normalization and ReLU activation, while the output layer uses Sigmoid activation. The process of encoding and decoding is expressed as Eqs. (2) and (3)[22].

$$z = E(x) \tag{2}$$
$$\hat{x} = D(z) \tag{3}$$

where x denotes the input image, z represents the latent vector produced by the encoder, and $\hat{x}$ is the reconstructed image generated by the decoder. Autoencoder training was carried out by minimizing the combination of reconstruction loss and adversarial loss, according to the adversarial autoencoder architecture used. Reconstruction loss uses Mean Squared Error (MSE), which is mathematically defined by Eq. (4).

$$L_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{4}$$

MSE calculates the average square difference between the original image pixels x_i and the reconstructed pixels $\hat{x}_i$ [23]. Meanwhile, adversarial loss is obtained from the discriminator output on the reconstruction images. The discriminator architecture consists of convolutional layers with LeakyReLU activation and a Sigmoid output layer. The function used is a special form of Binary Cross Entropy (BCE). Mathematically, BCE is defined by Eq. (5) [24].

$$L_{\text{BCE}} = -\left[y \log D(x) + (1-y)\log(1 - D(x))\right] \tag{5}$$

In discriminator-assisted autoencoder architectures, the goal of the decoder is to produce a reconstruction image $\hat{x}$ that is classified as real, so the target is $y = 1$. By simplifying Eq. (5) for $y = 1$, the adversarial loss can be expressed as Eq. (6).

$$L_{\text{adv}} = -\log D(\hat{x}) \tag{6}$$

This simplification is often used on adversarial autoencoders because it provides a more stable and stronger gradient, especially when the discriminator still considers the reconstruction to be false data. Based on the previous two loss components, reconstruction loss and adversarial loss, the entire function of the autoencoder loss is defined by Eq. (7).

$$L_{\text{AE}} = L_{\text{MSE}} + \lambda_{\text{adv}} \cdot L_{\text{adv}} \tag{7}$$

with $\lambda_{adv} = 0.01$. The autoencoder was trained for 200 epochs using the Adam optimizer with a learning rate of 0.0002 and β values of (0.5, 0.999). A batch size of 64 was used during training. After the autoencoder training process was completed, each minority class image was projected into a latent space as defined by Eq. (8).

$$z_i = E(x_i) \tag{8}$$

DeepSMOTE then generated synthetic samples by applying SMOTE in a latent space. For each latent vector z_i, the nearest neighbor were determined using the k-nearest neighbors method with $k = 6$. One neighbor z_{nn} was then randomly selected, followed by linear interpolation as defined by the Eq. (9) [25][26].

$$z_{\text{new}} = z_i + \lambda\left(z_{nn} - z_i\right), \quad \lambda \sim U(0,1) \tag{9}$$

The latent-space interpolation mechanism allows synthetic samples to be generated from compressed feature representations learned by the encoder. Because latent representations capture important structural characteristics of tea leaf diseases, interpolation in this space may produce smoother and more semantically consistent variations compared to direct interpolation in the original image or feature space. This mechanism may help preserve intra-class characteristics while still introducing variability among minority-class samples, potentially improving the representation of minority classes. Unlike conventional augmentation, which relies on predefined image transformations, and feature-space SMOTE, which performs interpolation directly on extracted feature vectors, DeepSMOTE performs interpolation in the learned latent space. The synthetic vectors were then reconstructed into images using the decoder defined by Eq. (10).

$$x_{\text{new}} = D\left(z_{\text{new}}\right) \tag{10}$$

At this stage, the number of synthetic images produced was adjusted so that each class in the training data has a total of 114 images. The synthetic images resulting from DeepSMOTE were then combined with the minority-class training data to balance the class distribution prior to DenseNet-121 training. Although DeepSMOTE improved minority-class representation, the method also introduced additional computational complexity because the encoder-decoder and discriminator networks needed to be trained before the classification stage.

2.4.2. SMOTE

As a comparison method, SMOTE was applied to balance the training data using DenseNet-121-based feature representations. Features were extracted from the training images using the feature extraction layers of a pre-trained DenseNet-121 model, followed by global average pooling to obtain fixed-length feature vectors. Unlike DeepSMOTE, which performs interpolation in autoencoder latent space and reconstructs synthetic images, SMOTE generates synthetic samples directly in feature space through interpolation between minority-class feature vectors.

The synthetic feature vectors were generated using the k-nearest neighbors interpolation mechanism, as expressed in Eq. (11).

$$x_{\text{new}} = x_i + \lambda\left(x_{nn} - x_i\right), \quad \lambda \sim U(0,1) \tag{11}$$

where x_i represents the original feature vector, x_{nn} represents one of its nearest neighbors, and λ is a random value sampled from a uniform distribution.

SMOTE was applied only to the training dataset to prevent data leakage. The generated synthetic feature vectors were then combined with the original training features to form a balanced dataset. These balanced feature representations were classified using a fully connected classifier head consisting of a linear layer with 512 neurons, ReLU activation, dropout with a rate of 0.3, and an output layer corresponding to the number of disease classes.

To provide a fairer comparison with DeepSMOTE, both methods employed DenseNet-121-based feature representations and neural network classification frameworks. However, the two approaches did not share identical pipeline structures. DeepSMOTE generated reconstructed synthetic images through interpolation in the autoencoder latent space, whereas SMOTE generated synthetic samples directly in the DenseNet-121 feature space and classified them using a classifier head. Therefore, the comparison presented in this study should be interpreted as a comparison between two interpolation-based balancing strategies rather than strictly equivalent processing pipelines.

2.4.3. Conventional Augmentation

Conventional augmentation provides image variation through several transformations applied on the fly during training, including random horizontal flipping with probability 0.5, random vertical flipping with probability 0.3, random rotation within a range of 15°, and brightness and contrast adjustment with an intensity factor of 0.2. This process does not permanently store new images, so the amount of training data does not increase and the class distribution remains unchanged.

2.5. Transfer Learning DenseNet-121

At the classification stage, all synthetic images of DeepSMOTE, which were originally 128 × 128 pixels in size, were resized to 224 × 224 using bilinear interpolation to match the input size of the DenseNet-121 model [27]. Meanwhile, the images in the no augmentation and conventional augmentation scenarios did not require additional resizing because they were already processed at 224 × 224 pixels. The DenseNet-121 model in this study uses ImageNet1K pre-trained weight. All layers of the model were fine-tuned during training, while the original classifier layer was replaced with a new fully connected layer according to the number of tea leaf disease classes.

DenseNet-121 utilizes a dense inter-layer connection so that each layer receives all of the output from the previous layer as input [28]. This mechanism allows the network to maximize feature reuse while maintaining strong gradient flow in deep networks. The relationship between layers is expressed in Eq. (12) [29].

$$x_l = H_l\left([x_0, x_1, \ldots, x_{l-1}]\right) \tag{12}$$

where $H_l(\cdot)$ is a composite function consisting of Batch Normalization, ReLU, and convolution. On the dense block layer, these three operations are executed in the order of Batch Normalization → ReLU → 3×3 convolution as part of the feature extraction process.

In DenseNet-121, the layers inside the dense block apply a layer bottleneck, which is 1×1 convolution, to reduce the number of channels so that computing becomes lighter. Thus, one layer in a dense block that uses a bottleneck layer consists of Batch Normalization → ReLU → convolution 1×1 → Batch Normalization → ReLU → convolution 3×3. Each layer in a dense block adds a new feature equal to the growth rate (k). If the number of initial features is k_0, then the number of features that enter the l layer can be defined by Eq. (13) [29].

$$\text{Input}_l = k_0 + k(l-1) \tag{13}$$

On DenseNet-121, the growth rate is 32, so the addition of features to each layer remains controlled. Between dense blocks there is a transition layer responsible for reducing the feature size to maintain computational efficiency. At this stage, the spatial resolution is reduced through average pooling, while the number of channels is reduced using 1×1 convolution. This reduction is controlled by the compression factor $\theta = 0.5$, so the number of feature maps becomes half of the previous size [29]. After all dense blocks are processed, DenseNet-121 uses Global Average Pooling (GAP) to reduce the feature tensor to one average value per channel. GAP makes the number of parameters much smaller because it replaces the fully connected layer which is usually large, making the model lighter. The GAP result feature vector is then mapped to the class score using a new linear layer according to the number of tea leaf disease classes. The score is eventually converted into a class probability through the softmax function.

2.6. Training

During training, the model was optimized using categorical cross entropy loss, which measures the difference between the predictions and the actual label. The function of the CCE is defined by Eq. (14) [24].

$$L = -\sum_{c=1}^{C} y_c \log\left(\hat{y}_c\right) \tag{14}$$

with $y_c = 1$ for the correct class and 0 for the other classes. The training was conducted for 20 epochs using Adam's optimizer with a learning rate of 1×10^{-4} and a batch size of 32. The same training hyperparameters, including learning rate, batch size, optimizer, and number of epochs, were consistently used across all comparison methods to ensure fair evaluation. To improve the reliability of the experimental results, a repeated-experiment protocol was employed. The train, validation, and test split described in Section 2.2 was kept fixed for all experiments. For each oversampling method, the model training process was repeated five times using different random seeds. These seeds affected model parameter initialization and data shuffling during training, while the training, validation, and test samples remained identical across runs. No early stopping or checkpoint selection strategy was applied in this study. Each model was trained for a fixed 20 epochs, and the final epoch model was directly used for evaluation. Training, validation, and testing accuracy were recorded in every run, and the final performance was reported as mean ± standard deviation.

2.7. Evaluation

After the DenseNet-121 transfer learning training process is completed, the model's performance was evaluated using test data to assess generalization capability on unseen. Evaluation was carried out using confusion matrix, which summarizes the numbers of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). This information is used to calculate accuracy, precision, recall, and F1-score metrics. The formula for calculating evaluation metrics is defined by Eqs. (15)–(18) [30].

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{15}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

$$F_1\text{-score} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

3. Results and Discussion

This section presents the experimental results of the proposed tea leaf disease classification framework. The discussion begins with the data balancing results produced by DeepSMOTE, followed by model training performance, classification evaluation, comparison with alternative augmentation methods, and overall performance analysis.

3.1. Data Balancing Results Using DeepSMOTE

A total of 204 synthetic images were generated through the DeepSMOTE process, so that the total number of images in the training data after being added with synthetic images increased from 708 to 912 images with a balanced distribution of 114 images for each class. The amount of synthetic imagery added varies depending on the amount of original data for each class. Details of the additions are presented in Table 2 below.

Table 2: Number of synthetic images resulting from DeepSMOTE

Class	Original data	Synthetic data	Final total (target)
Anthracnose	80	34	114
Algal Leaf	91	23	114
Bird Eye Spot	80	34	114
Brown Blight	90	24	114
Gray Light	80	34	114
Healthy	59	55	114
Red Leaf Spot	114	0	114
White Spot	114	0	114
Sum	**708**	**204**	**912**

The largest addition occurred in Healthy class of 55 synthetic images because it had the least amount of initial training data. While classes with more data such as Algae Leaf and Brown Blight only need about 23-24 synthetic images to achieve balance. The distribution before and after balancing is visualized in Fig. 2.

Some examples of synthetic images generated using the DeepSMOTE process are shown in Table 3 to illustrate the visual characteristics of the oversampling results produced in the latent

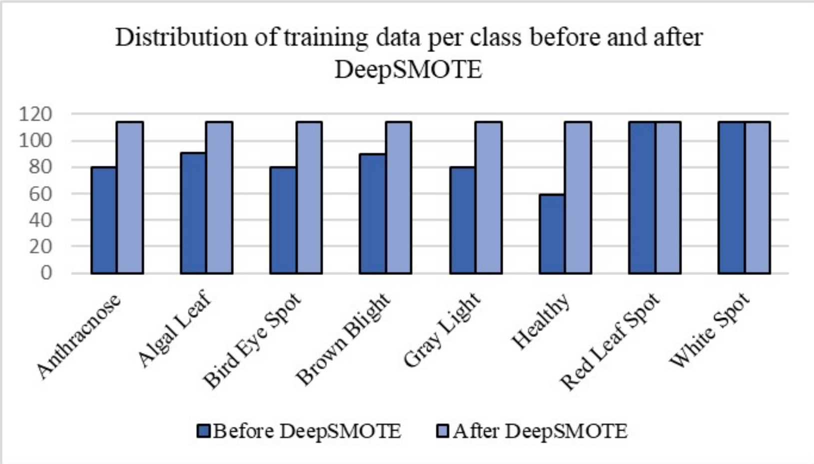
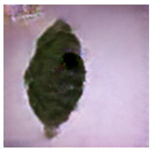
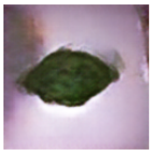
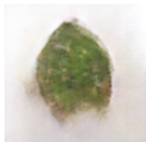
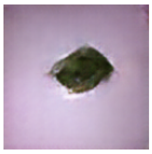
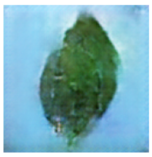


Fig. 2: Distribution of training data per class before and after DeepSMOTE

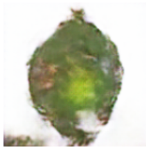
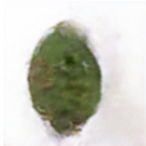
space of the autoencoder. Table 3 presents a comparison between the original and synthetic images.

Table 3: Example of synthesized images using DeepSMOTE

Class	Original image	Synthetic image
Anthracnose		
Algal Leaf		
Bird Eye Spot		
Brown Blight		
Gray Light		
Healthy		

Continued on the next page

Table 3: Example of synthesized images using DeepSMOTE (continued)

Class	Original image	Synthetic image
Red Leaf Spot		
White Spot		

The synthetic images generally preserve important visual characteristics of the original tea leaf images, including lesion patterns, color distribution, and leaf texture. A qualitative visual assessment was conducted by comparing lesion patterns, texture consistency, and color distribution between the original and synthetic images. Although several reconstructed images appear slightly smoother than the original images due to the autoencoder reconstruction process, the major disease characteristics remain recognizable. These observations suggest that latent-space interpolation was able to generate additional samples while maintaining the overall semantic structure of the disease patterns. Compared to conventional augmentation methods such as rotation, flipping, and brightness adjustment, DeepSMOTE generates additional training data through interpolation in the latent space learned by the autoencoder. This approach may help preserve meaningful visual structures during synthetic image generation.

However, several synthetic artifacts were still observed. Some generated images contain smoother textures and less detailed lesion boundaries compared to the original images. Although these artifacts were relatively minor in this study, excessive deviation from the original image distribution may potentially influence model learning. It should be noted that the evaluation of synthetic image quality in this study remained qualitative and was primarily based on visual inspection. More systematic or quantitative image quality assessments, such as similarity-based or perceptual metrics, may provide additional insight into the fidelity of the generated images and could be considered in future work.

3.2. Training Result

The new training data, which contains a combination of the synthetic images of DeepSMOTE results with the original imagery, was then used to train the DenseNet-121 transfer learning classification model. To improve the reliability of the experiment, the DeepSMOTE model training was repeated five times using different random seeds. The training, validation, and testing accuracies from each experiment are presented in Table 4.

Table 4: Repeated experiment results of DeepSMOTE

Run	Train accuracy (%)	Validation accuracy (%)	Test accuracy (%)
Run 1	99.89	93.18	97.75
Run 2	99.89	94.32	91.01
Run 3	99.89	93.18	96.63
Run 4	99.89	95.45	97.75
Run 5	99.89	94.32	95.51

Based on the repeated experiments, the DeepSMOTE model achieved an average training accuracy of 99.89% ± 0.00%, validation accuracy of 94.09% ± 0.85%, and testing accuracy of 95.73% ± 2.50%. These results suggest that the DeepSMOTE approach provided relatively stable classification performance across repeated experiments, although some variability between runs was still observed. In addition, the 95% confidence interval of the test accuracy ranged from 92.26% to 99.20%, indicating that the model performance remained reasonably consistent across repeated experiments, although variations between runs were still observed. The training and validation curves from the final experiment are shown in Fig. 3.

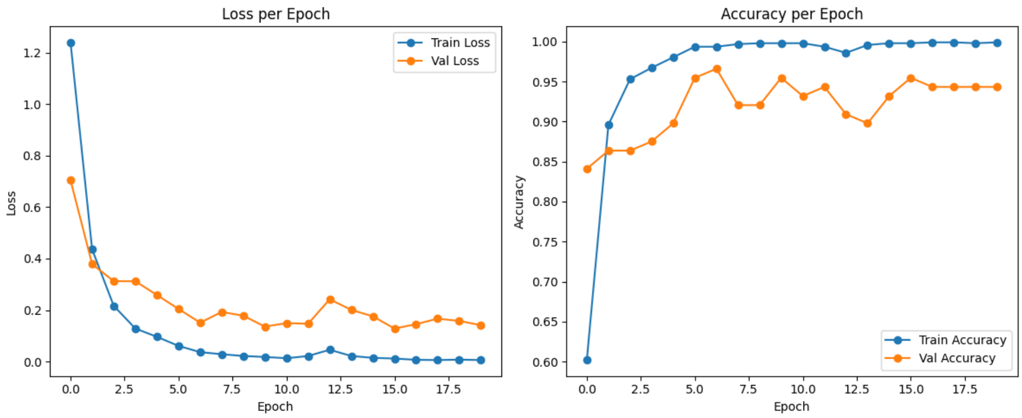


Fig. 3: Training and validation loss and accuracy curves from the final DeepSMOTE experiment

Based on Fig. 3, the training loss decreases rapidly during the early epochs and continues to decrease gradually until the end of training. Validation loss also showed a decreasing trend, although slight fluctuations still appeared in several epochs. Meanwhile, training accuracy increased significantly and reached a very high value in later epochs, while validation accuracy remained relatively stable around 94–96%. Although the model achieved high training accuracy, a gap between training and validation performance was still observed. This indicates that slight overfitting may still occur, although validation performance remains relatively stable.

In addition, the testing dataset only consisted of 89 images, and several classes had relatively small support sizes. Therefore, the high performance obtained in this study should be interpreted cautiously and may not fully represent the generalization capability of the model on larger and more diverse datasets. The classification report from the final DeepSMOTE experiment is presented in Table 5.

Table 5: Classification report of DeepSMOTE on the final run

Class	Precision	Recall	F1-score	Support
Anthracnose	1.0000	0.9000	0.9474	10
Algal Leaf	1.0000	1.0000	1.0000	11
Bird Eye Spot	1.0000	1.0000	1.0000	10
Brown Blight	0.9091	0.8333	0.8696	12
Gray Light	0.9091	1.0000	0.9524	10
Healthy	1.0000	1.0000	1.0000	7
Red Leaf Spot	1.0000	1.0000	1.0000	15
White Spot	0.8667	0.9286	0.8966	14
Accuracy			0.9551	89
Macro avg	0.9606	0.9577	0.9585	89
Weighted avg	0.9566	0.9551	0.9549	89

Based on the classification report, most classes achieved relatively high precision, recall, and F1-score values. Several classes such as Algal Leaf, Bird Eye Spot, Healthy, and Red Leaf Spot achieved perfect scores on all evaluation metrics. However, several classes still showed imperfect performance, particularly Brown Blight and White Spot. Brown Blight achieved a recall of 0.8333 and an F1-score of 0.8696, while White Spot obtained a precision of 0.8667 and an F1-score of 0.8966. These results indicate that some images were still difficult for the model to distinguish consistently. The confusion matrix from the final DeepSMOTE experiment is shown in Fig. 4.

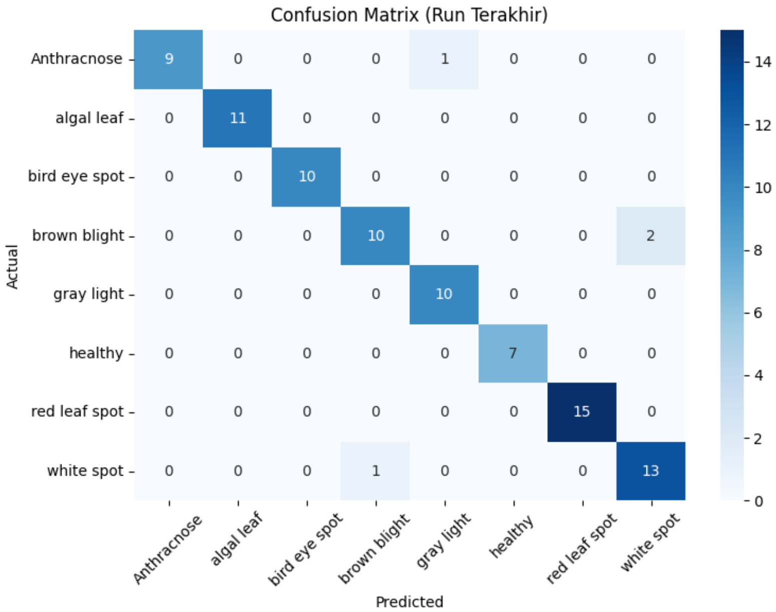


Fig. 4: Confusion matrix of DeepSMOTE on the final run

Most predictions are concentrated on the main diagonal, indicating that the majority of test images were classified correctly. However, several misclassifications still occurred. In particular, two Brown Blight images were predicted as White Spot, while one White Spot image was predicted as Brown Blight. Examples of original images from both classes are presented in Fig. 5.

(a) Brown Blight

(b) White Spot

Fig. 5: Original image samples of (a) Brown Blight and (b) White Spot.

Based on the original image samples, Brown Blight generally shows brownish lesions concentrated near the leaf edges with darker and more irregular infected regions. In contrast, White Spot tends to exhibit lighter lesion areas with more diffuse spot patterns across several parts of the leaf surface. Despite these differences, both classes still share similar color distributions and texture patterns, especially in images with mild infection severity. To further analyze the classification error, an example of a Brown Blight image misclassified as White Spot is presented in Fig. 6.

Fig. 6: Example of a Brown Blight image misclassified as White Spot

The misclassified Brown Blight image contains relatively faint lesions with scattered light-brown spots, visually resembling characteristics observed in White Spot images. In addition, the image appears slightly blurry, reducing the clarity of lesion boundaries and making feature separation more difficult for the model.

Variations in illumination, leaf orientation, image sharpness, and disease severity may further contribute to overlapping feature representations between classes. Overall, the confusion matrix suggests that most samples were classified correctly, although several visually similar disease classes remained difficult to distinguish consistently. The observed errors indicate that overlapping visual characteristics between diseases may still limit classification performance.

3.3. Evaluation of Comparative Methods

To assess the effectiveness of the DeepSMOTE method, a comparison was made with the no-augmentation, conventional augmentation, and SMOTE approaches.

3.3.1. No Augmentation

The no augmentation approach achieved an average test accuracy of 89.89% ± 2.56%, with a 95% confidence interval of [86.33%, 93.45%]. On the final run, the model achieved 91% accuracy. Several classes still showed relatively lower performance, such as Algal Leaf with a recall of 0.64 and F1-score of 0.78, and White Spot with an F1-score of 0.80. These results indicate that without balancing or augmentation, the model still experienced difficulty in learning feature representations evenly across all classes.

3.3.2. Conventional Augmentation

The conventional augmentation approach achieved an average test accuracy of 93.48% ± 1.65%, with a 95% confidence interval of [91.19%, 95.78%]. On the final run, the model achieved 93.26% accuracy. This improvement suggests that transformations such as flipping, rotation, and brightness adjustment helped increase visual variability in the training data. However, several classes still obtained lower performance compared to other classes, such as Anthracnose with a recall of 0.70 and F1-score of 0.78, and Bird Eye Spot with an F1-score of 0.82. These results indicate that conventional augmentation improved model performance but did not fully address the class imbalance problem.

3.3.3. SMOTE

The SMOTE approach achieved an average test accuracy of 89.21% ± 1.52%, with a 95% confidence interval of [87.10%, 91.33%]. On the final run, the model achieved 87% accuracy. Several classes still showed relatively lower performance, including Anthracnose with an F1-score of 0.70, Bird Eye Spot with 0.76, and Gray Light with 0.76.

These results indicate that feature-space interpolation was able to improve class balance and overall classification performance. However, some synthetic feature representations may not have fully preserved fine-grained disease characteristics, causing overlap between visually similar classes and contributing to several remaining misclassifications.

3.4. Performance Comparison Between Methods

The performance comparison of all methods is shown in Fig. 7.

The graph is written on a decimal scale of 0–1. Based on the repeated experiments, DeepSMOTE achieved the highest average test accuracy of 95.73% ± 2.50%, followed by conventional augmentation with 93.48% ± 1.65%, no augmentation with 89.89% ± 2.56%, and SMOTE with 89.21% ± 1.52%. Compared to no augmentation, conventional augmentation, and SMOTE, DeepSMOTE improved the average test accuracy by approximately 5.84%, 2.25%, and 6.52%, respectively. The corresponding 95% confidence intervals were [92.26%, 99.20%] for DeepSMOTE, [91.19%, 95.78%] for conventional augmentation, [86.33%, 93.45%] for no augmentation, and [87.10%, 91.33%] for SMOTE. Although the confidence interval of DeepSMOTE was relatively

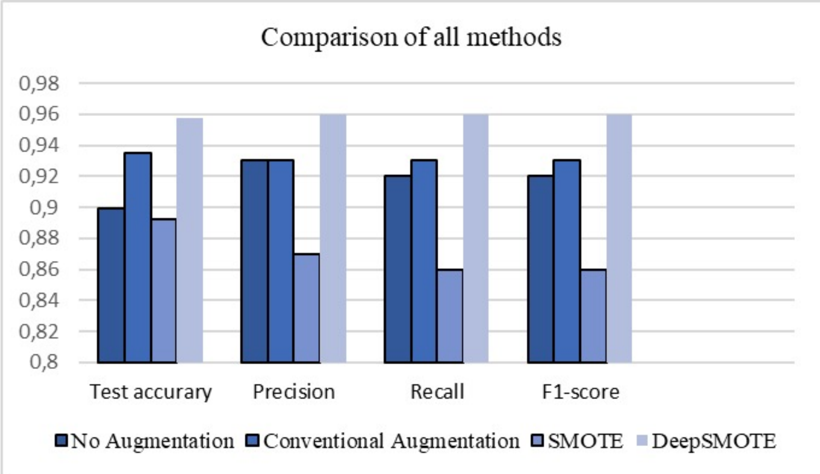


Fig. 7: Performance comparison of all methods

wider, overlap with conventional augmentation was still observed. These results suggest that DeepSMOTE generally maintained higher average performance across repeated experiments within the scope of this study.

The higher average performance observed for DeepSMOTE in this study may be associated with interpolation in the latent space of the autoencoder, which may help preserve important structural characteristics of tea leaf diseases while generating smoother and more semantically consistent variations among minority-class samples. In contrast, conventional augmentation only modifies existing images without generating entirely new representations for minority classes. Meanwhile, SMOTE generates synthetic samples through interpolation in the extracted feature representation space. Although this approach improved class balance, overlapping feature distributions may still have contributed to several misclassifications between visually similar diseases.

It should be noted that the comparison between DeepSMOTE and SMOTE in this study is informative but does not involve identical pipeline structures. DeepSMOTE generates reconstructed synthetic images through an encoder-decoder framework and performs classification on image data, whereas SMOTE was applied directly to DenseNet-121 feature representations followed by a classifier head. Therefore, the observed performance differences should be interpreted as reflecting two different interpolation-based balancing pipelines rather than strictly equivalent implementations. In addition, DeepSMOTE requires additional computational stages, including encoder-decoder and discriminator training prior to classification, resulting in higher computational complexity compared to conventional augmentation and SMOTE.

Although DeepSMOTE achieved the highest average performance in this study, the results should be interpreted with caution because the testing dataset was relatively limited and performance variations between repeated experiments were still observed, as reflected by the standard deviation of 2.50% and the corresponding 95% confidence interval. Therefore, the present findings suggest that DeepSMOTE may provide improved classification performance under the experimental settings considered in this study, although additional statistical significance tests and larger test sets would be beneficial to further confirm these observed differences.

4. Conclusions

This study evaluated different augmentation and imbalance handling approaches for tea leaf disease classification using DenseNet-121 transfer learning. Based on repeated experiments, DeepSMOTE achieved the highest average test accuracy of 95.73% $\pm$ 2.50%, with a 95% confidence interval of [92.26%, 99.20%]. Compared with conventional augmentation, no augmentation, and SMOTE, DeepSMOTE generally showed stronger performance across repeated experiments within the scope of this study. The results suggest that latent-space interpolation in DeepSMOTE may have contributed to improved minority-class representation and classification performance. These findings suggest that DeepSMOTE may be a promising approach for handling class imbalance in tea leaf disease classification. Since this study was conducted using a single dataset with a relatively limited test set, the results should be interpreted with caution and may not fully represent the generalization capability of the proposed approach to other datasets. Future research should evaluate DeepSMOTE on larger datasets and compare it with other imbalance handling approaches such as focal loss, class weighting, and GAN-based augmentation.

CRediT Authorship Contribution Statement

Nikita Putri Insani: Conceptualization, Methodology, Formal Analysis, Writing-Original Draft, Visualization. **Sugiyarto Surono:** Methodology, Validation, Writing-Review & Editing, Project Administration. **Aris Thobirin:** Validation, Writing-Review & Editing, Project Administration.

Declaration of Generative AI and AI-assisted technologies

ChatGPT was used solely for language refinement, including paraphrasing and grammar improvement. All content, analysis, and conclusions remain the responsibility of the authors.

Declaration of Competing Interest

The authors declare no competing interest.

Funding and Acknowledgments

This research received no external funding. The authors thank the Department of Mathematics, Universitas Ahmad Dahlan, for providing academic support and facilities.

Data and Code Availability

The dataset used in this study is publicly available on Kaggle under the title Identifying Disease in Tea Leaves. The source code used in this study is available from the corresponding author upon reasonable request.

References

[1] Sheikh M Saqib et al. "DenseHillNet: a lightweight CNN for accurate classification of natural images". In: *PeerJ Computer Science* 10 (2024), pp. 1–21. DOI: 10.7717/PEERJ-CS.1995.

[2] Kadek R Ariawan, Anak A G Ekayana, I PY Indrawan, et al. "Performance Comparasion of DenseNet-121 and MobileNetV2 for Cacao Fruit Disease Image Classification". In: *Indonesian Journal of Data Science* 6.1 (2025), pp. 30–38. DOI: 10.56705/ijodas.v6i1.233.

[3] Tao Zhou, XinYu Ye, HuiLing Lu, et al. "Dense Convolutional Network and Its Application in Medical Image Analysis". In: *Biomedical Research International* 2022.1 (2022). DOI: 10.1155/2022/2384830.

[4] Abayomi Bello, Sin-Chun Ng, and Man-Fai Leung. "Skin Cancer Classification Using Fine-Tuned Transfer Learning of DENSENET-121". In: *Applied Sciences* 14.17 (2024). DOI: 10.3390/app14177707.

[5] Adi D Saputra, Djarot Hindarto, and Handri Santoso. "Disease Classification on Rice Leaves using DenseNet121, DenseNet169, DenseNet201". In: *Sinkron* 7.1 (2023), pp. 48–55. DOI: 10.33395/sinkron.v8i1.11906.

[6] Cevher Ozden. "Corn Disease Detection Using Transfer Learning". In: *Black Sea Journal of Engineering and Science* 6.4 (2023), pp. 387–393. DOI: 10.34248/bsengineering.1322907.

[7] Mohammadreza Iman, Hamid R Arabnia, and Khaled Rasheed. "A Review of Deep Transfer Learning and Recent Advancements". In: *Technologies* 11.2 (2023), pp. 1–14. DOI: 10.3390/technologies11020040.

[8] Mehwish Rasheed, Muhammad A Jaffar, Arslan Akram, et al. "Improved brain tumor classification through DenseNet121 based transfer learning". In: *Discover Oncology* 16.1 (2025). DOI: 10.1007/s12672-025-03501-3.

[9] Waleed Albattah and Rehan U Khan. "Impact of imbalanced features on large datasets". In: *Frontiers in Big Data* 8 (2025). DOI: 10.3389/fdata.2025.1455442.

[10] Ying Li, Yali Yang, Peihua Song, Lian Duan, and Rui Ren. "An improved SMOTE algorithm for enhanced imbalanced data classification by expanding sample generation space". In: *Scientific Reports* 15.1 (2025), pp. 1–21. DOI: 10.1038/s41598-025-09506-w.

[11] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357. DOI: 10.1613/jair.953.

[12] Mudhafar Jalil Jassim Ghrabat, Arkan A. Ghaib, et al. "Effective SMOTE boost with deep learning for IDC identification in whole-slide images". In: *PLoS One* 20.9 (2025), pp. 1–34. DOI: 10.1371/journal.pone.0329078.

[13] Ekram Chamseddine, Nesrine Mansouri, Makram Soui, and Mourad Abed. "Handling class imbalance in COVID-19 chest X-ray images classification: Using SMOTE and weighted loss". In: *Applied Soft Computing* 129 (2022), p. 109588. DOI: 10.1016/j.asoc.2022.109588.

[14] Agus E Minarno, Ilham S Kantomo, Fauzi D S Sumadi, et al. "Classification of Brain Tumors on MRI Images Using DenseNet and Support Vector Machine". In: *International Journal of Informatics and Visualization* 6.2 (2022), pp. 404–410. DOI: 10.30630/joiv.6.2.991.

[15] Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, et al. "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications". In: *Journal of Big Data* (2023). DOI: 10.1186/s40537-023-00727-2.

[16] Arnab Kumar Mondal, Lakshya Singhal, Piyush Tiwary, Parag Singla, and Prathosh AP. "Minority Oversampling for Imbalanced Data via Class-Preserving Regularized Auto-Encoders". In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Vol. 206. Proceedings of Machine Learning Research. PMLR, 2023, pp. 3440–3465. https://proceedings.mlr.press/v206/mondal23a.html.

[17] Anburaj Jeyaraj, Tamilselvi Elango, Xuan Chen, et al. "Advances in understanding the mechanism of resistance to anthracnose and induced defence response in tea plants". In: *Molecular Plant Pathology* 24.10 (2023), pp. 1330–1346. DOI: 10.1111/mpp.13354.

[18] Intan Ratna Dewi Anjarsari et al. "Effect of Plant Growth Regulator on Growth, Yield and Catechin Content of Tea (Camellia sinensis (L.) O. Kuntze)". In: *Int. J. Adv. Sci. Eng. Inf. Technol* 12.1 (2022), pp. 403–410. DOI: 10.18517/ijaseit.12.1.11657.

[19] Hafijur Rahman, Iftekhar Ahmad, Parvej Hasan Jon, Abdus Salam, and Md. Forhad Rabbi. "Automated detection of selected tea leaf diseases in Bangladesh with convolutional neural network". In: *Scientific Reports* 14.1 (2024), pp. 1–12. DOI: 10.1038/s41598-024-62058-3.

[20] Narayan Sapkota, Khim Bahadur Khattri, and Divyashwori Aryal. "Modeling Precipitation: A Statistical and Machine Learning Approach". In: *InJET* 2.2 (2025), pp. 188–203. DOI: 10.3126/injet.v2i2.78616.

[21] Damien Dablain, Bartosz Krawczyk, and Nitesh V. Chawla. "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data". In: *IEEE Transactions on Neural Networks and Learning Systems* 34.9 (2023), pp. 6390–6404. DOI: 10.1109/TNNLS.2021.3136503.

[22] Moncef Garouani, Adeel Ahmad, Mourad Bouneffa, and Mohamed Hamlich. “Autoencoder-kNN meta-model based data characterization approach for an automated selection of AI algorithms”. In: *Journal of Big Data* 10.14 (2023). DOI: 10.1186/s40537-023-00687-7.

[23] Robert K. L. Kennedy, Flavio Villanustre, Taghi M. Khoshgoftaar, and Zahra Salekshahrezaee. “Synthesizing class labels for highly imbalanced credit card fraud detection data”. In: *Journal of Big Data* 11.38 (2024). DOI: 10.1186/s40537-024-00897-7.

[24] Juan Terven, DM Cordova-Esparza, JA Romero-Gonzalez, et al. “A comprehensive survey of loss functions and metrics in deep learning”. In: *Artificial Intelligence Review* 58.195 (2025). DOI: 10.1007/s10462-025-11198-7.

[25] Yi-Yang Shangguan, Shi-Shun Chen, and Xiao-Yang Li. “Minimum Enclosing Ball Synthetic Minority Oversampling Technique from a Geometric Perspective”. In: *IEEE Transactions on Industrial Informatics* (2024), pp. 1–8. DOI: 10.48550/arXiv.2408.03526.

[26] Annisa Salsabila Ahdyani, Irwan Budiman, Dwi Kartini, Andi Farmadi, et al. “Implementation of Chi-Square Feature Selection for Parkinson’s Disease Classification Using Light GBM”. In: *IJCCS* 19.3 (2025), pp. 327–336. DOI: 10.22146/ijccs.107881.

[27] Xiaoyu Wu et al. “GSDNet: A deep learning model for downscaling the significant wave height based on NAFNet”. In: *Journal of Sea Research* 198 (2024), p. 102482. DOI: 10.1016/j.seares.2024.102482.

[28] Gao Huang, Kilian Q. Weinberger, et al. “Densely Connected Convolutional Networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2017). DOI: 10.1109/CVPR.2017.243.

[29] Firas I AlZobi et al. “Optimized soft-voting CNN ensemble using particle swarm optimization for endometrial cancer histopathology classification”. In: *Comput. Methods Programs Biomed. Updat* 8 (2025), p. 100217. DOI: 10.1016/j.cmpbup.2025.100217.

[30] Khang Wen Goh, Sugiyarto Surono, et al. “Comparison of Activation Functions in Convolutional Neural Network for Poisson Noisy Image Classification”. In: *Emerging Science Journal* 8.2 (2024), pp. 592–602. DOI: 10.28991/ESJ-2024-08-02-014.