Hybrid CNN-SVM with Borderline SMOTE for Imbalance Class Cabbage Plants

Nabila Ayunda Sovia[1*], Ni Wayan Surya Wardhani[1], and Eni Sumarminingsih[1]
[1]Department of Statistics, Brawijaya University, Malang City, Indonesia
*Corresponding author: nabilaayunda003@student.ub.ac.id

Received: 31 May 2024
Revised: 9 September 2024
Accepted: 30 September 2024

ABSTRACT – Cabbage farming is highly vulnerable to diseases and pests, leading to substantial yield losses if not properly managed. Traditional diagnostic methods, reliant on manual assessment, are often time-consuming and inaccurate. This study introduces a hybrid approach combining Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) to address these challenges, specifically focusing on improving classification accuracy in imbalanced cabbage image datasets. CNNs are leveraged for their powerful feature extraction, while SVM, optimized using a One-vs-All strategy, enhances multi-class classification. To handle data imbalance, Borderline SMOTE (Synthetic Minority Over-sampling Technique) is applied, generating synthetic samples to balance underrepresented classes. The SqueezeNet architecture is employed for feature extraction, with SVM hyperparameters fine-tuned via grid search. Results demonstrate that the integration of CNN, SVM, and Borderline SMOTE significantly improves classification performance, particularly for minority classes, achieving an accuracy of 99%. This approach offers a more reliable and efficient tool for early detection of cabbage diseases and pests, contributing to better agricultural management and reduced crop losses.

Keywords– Cabbage Plants, CNN, Imbalance Data, SVM.

I. INTRODUCTION

Brassica oleracea var. capita or cabbage, is a widely cultivated crop used as food, and easy to attack by disease and pest. Improper management of these issues can lead to crop failure and significant losses for farmers [1] Approaches like Convolutional Neural Networks (CNNs) are often used to address this. CNNs can automatically detect diseases with high accuracy and without requiring much time. The use of this method will assist farmers in early disease detection, ultimately improving crop management and yield [2]. However, real-world data is often imbalanced, as in the case of cabbages, where more cabbages are affected by pests than by diseases [3]. Although CNN is a reliable method for image classification, when faced with imbalanced data, CNN models tend to overfit to the majority class and struggle to capture the necessary features from the minority class, leading to poor generalization [4].

One way to solve this problem is by using data preprocessing techniques, such as resampling methods [5]. Studies have applied Borderline-SMOTE, an oversampling technique, to address imbalanced data in CNN-based brain image classification. This method generates synthetic samples near the decision boundary for minority classes, helping to balance the data before splitting it into training and testing sets. This framework helps CNN train on more balanced data and capture the necessary diversity in features for the minority class [6]. However, while Borderline-SMOTE can improve class representation, it has its limitations. In multiclass settings, the creation of synthetic samples can lead to class overlap, complicating the learning process, and increasing ambiguity in class boundaries [7].

To solve this, more advanced methods like hybrid approaches are proposed. Hybrids combine several powerful methods into one [8]. For example, a recent study demonstrated that hybrid CNN and SVM, show higher accuracy in multiclass classification [9]. SVM using One vs All method to improve model performance by treating them as separate binary problems, which particularly useful in imbalanced data and multiclass scenarios [10]. However, the effectiveness of the combined approach heavily relies on the quality and quantity of the training data. If the data is not representative of real-world scenarios, the performance of both CNN and SVM may suffer. This is where the role of Borderline-SMOTE comes in, as in hybrid models, this technique can help split the data more fairly, allowing SVM to create a more defined hyperplane, which can lead to better classification outcomes [11].

In this study, we aim to enhance the performance of CNN on imbalanced cabbage image datasets by utilizing two complementary approaches: data preprocessing and algorithmic solutions. Borderline-SMOTE will be applied to balance the dataset, while SVM will be integrated with CNN to improve multi-class classification performance. This approach is expected to assist farmers by providing a more reliable system for identifying diseases and pests in cabbages, ultimately improving agricultural practices in the future.

II. LITERATURE REVIEW

A. Base Model

CNN is a deep neural network architecture modeled after the functioning of the human brain. CNNs operate similarly to other feedforward neural network models, with the distinction that they incorporate convolutional layers to eliminate the important characteristics from images [12]. These features are subsequently employed for classification purposes or can be extracted and utilized in alternative methods [13]. A CNN's architecture comprises multiple layers, including the input layer, hidden layers, and the output layer. When integrated, these layers form a Multi-Layer Perceptron (MLP). The following equation (1) shows a single layer model [14].

$$y = f(z) = f\left(b + \sum_{k=1}^{K} w_k h_k(\boldsymbol{x}_i)\right) \quad (1)$$

where h is a hidden layer, z is a pre-activation function, and $\boldsymbol{x}$ is vector input. Parameter used: $\boldsymbol{w}$ is vector weight, and b is bias. The output is y, obtained by the non-linear function of input $x \in R^p$ parameterized by a synaptic vector of weights $(w_1, \ldots, w_p) \in R^p$ and a threshold $w_0 \in R$. For a neuron to be non-linear, it will be specified by an activation function $f(.)$. The activation function used is Leaky ReLu for feature learning and Softmax function for classification. The following equation (2) shows Leaky ReLu [15].

$$y = \begin{cases} 0{,}01, & if\ x < 0 \\ x, & if\ x > 0 \end{cases} \quad (2)$$

For Softmax function is shown in the following equation (3).

$$f(x) = \frac{e^{z_N}}{\sum_{l=0}^{N} e^{z_l}} \quad (3)$$

where N are amounts of pair (x_N, y_N) training data. There are many variants of architecture CNN, one of which is SqueezeNet. This architecture owing to the limited dataset size and the minimal parameter count, has approximately 1.2 million parameters, smaller than another architecture model. Given these advantages, this architecture is well-suited for image classification for imbalanced data [16].

B. Hybrid Model

SVM is a technique that functions by discovering the most effective separating line (hyperplane) to divide the data into positive and negative values [10]. The model for the hyperplane is shown in equation (4) above.

$$f(\boldsymbol{x}_i) = \boldsymbol{x}_i^T \boldsymbol{w} + b \quad (4)$$

while weight ($\boldsymbol{w}$) and bias (b) work as a parameter. To maximize the separate line, the weight vector expands the distance between the separate line and the nearest data called the margin. In non-linear data (image data) to maximize classification it used kernels. Radial Basis Function (RBF) is a common kernel used for high-dimension data. The following equation (5) shows the kernel RBF model.

$$K\langle \boldsymbol{x}_i, \boldsymbol{x}_{i'} \rangle = exp\left(-\gamma \sum_{j=1}^{p} \left(\boldsymbol{x}_{ij}, \boldsymbol{x}_{i'j}\right)^2\right) \quad (5)$$

where gamma (γ) is a positive constant to control the shape of the hyperplane [10].

Gird search is a method for optimizing hyperparameter, by utilizing the Cross-Validation technique as an evaluation metric with the aim of identifying effective hyperparameter. The wider the range of parameters available, the greater the potential for the Grid Search method to discover the best parameter combination [15].

C. Borderline-SMOTE

Borderline SMOTE is a modification of the SMOTE algorithm aimed at rectifying class imbalance in classification contexts. It concentrates on producing artificial samples for the minority class in close proximity to the decision boundary or boundary line separating the minority and majority classes. B-SMOTE identifies minority class samples proximate to this boundary, selects their nearest neighbors within the minority class, and subsequently generates synthetic instances. The following equation (6) shows the Borderline-SMOTE model.

$$x_{synthetic} = x_{borderline} + \lambda(x_{neighbor} - x_{borderline}) \quad (6)$$

where λ represents a random number within the range of 0 to 1 [17].

D. Evaluation Metrics

To evaluate the multi-class classification model, the Generalized Confusion Matrix is presented in **Table 1** displays the depiction of the confusion matrix for multiclass classification, reflecting the performance of the classification model. The diagonal elements, True Positives (TP), are used to calculate the correct predictions for each class. The off-diagonal elements represent the number of incorrect predictions, which are divided into two: the lower diagonal for False Positives (FP) and the upper diagonal for False Negatives (FN) [18].

Table 1 Confusion Matrix

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

Table 2 This section illustrates performance metrics derived from the values within the confusion matrix. Accuracy offers a broad evaluation of the model's overall performance. Recall evaluates the model's capability to accurately detect cabbage plants affected by pests. Precision assesses the model's accuracy in identifying cabbage plants unaffected by pests. Due to the dataset's imbalanced nature, the F1 score is calculated to offer a balanced evaluation of the model's performance, and Balanced accuracy is used to evaluate more fairly with K is representing number of classes in the model [19].

Table 2 Performance Measures

Name	Representation	
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	(7)
Loss	$1 - Accuracy$	(8)
Recall	$\frac{TP}{TP+FN}$	(9)
Precision	$\frac{TP}{TP+FP}$	(10)
F1-score	$2\ x\ \frac{Precision*Recall}{Precision+Recall}$	(11)
Balanced Accuracy	$\frac{1}{K}\sum_{k=1}^{K} Recall$	(12)

III. METHODOLOGY

A. Data

The data for this study were primarily obtained, and conducted in Poncokusumo, Malang, East Java in July 2023. The dataset consists of images of cabbage plants affected by diseases and pests, with an imbalance in the number of instances where more cabbage plants were affected by pests than by diseases. The images were captured using a Canon Mirrorless M100 camera on a 1000-square-meter plot. Cabbage plants typically have an average lifespan of 70 to 80 days, nearing the harvest period. The pests and diseases were identified based on visible external characteristics of the plants. External indicators such as odor or insects, which are symptomatic of disease and pest infestations in cabbage plants, cannot be identified from the images. A total of 242 images were collected, divided into 5 classes based on the visible characteristics of pests and diseases on the cabbages. Information regarding the diseases and the number of data instances in this study is presented in the following **Table 3**.

Table 3 Data Information

Type of Pests	Code	Data
Pest Infestation	P3	170
Foliar Disease	P6	35
Brassica rot disease	P7	12
Bacterial Soft Rot	P8	15
Plasmodiophoromycete Disease	P9	10

The images presented in **Figure 1** exhibit different diseases and pests commonly affecting cabbage plants. . In **Figure 1** (a) Pest Infestation is evident; although the plant's overall appearance may seem healthy, noticeable holes are present in the outer leaves. **Figure 1** (b) Illustrates cabbage impacted by Foliar Disease, displaying signs such as yellowish-brown discoloration and wilting of the outer leaves. **Figure 1** (c) illustrates symptoms of black or brown rot, manifested by brown spots on the cabbage head and subsequent wilting, eventually resulting in the plant's demise. **Figure 1** (d) portrays cabbage affected by Bacterial Soft Rot, identifiable by the darkened and deteriorating cabbage head. Lastly, **Figure 1** (e) depicts cabbage affected by Plasmodiophoromycete Disease, recognized by the lack of a well-formed head and stunted growth.

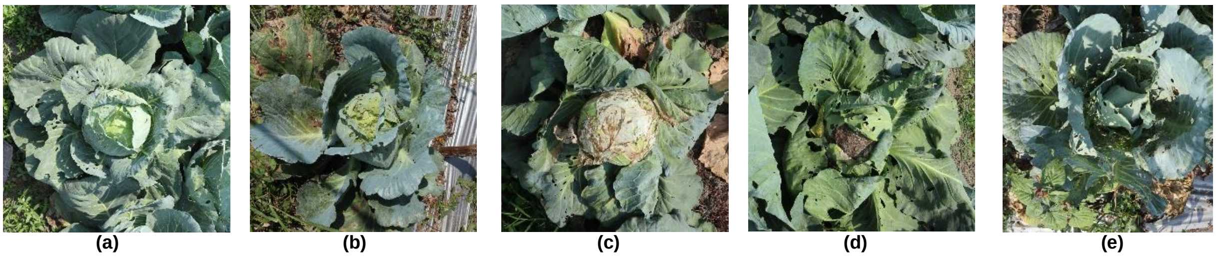

Figure 1 Infected Cabbage Plants (a) Pests Infestation, (b) Foliar Disease, (c) Brassica Rot, (d) Bacterial Soft Rot, (e) Plasmodiophoromycete Disease.

B. Proposed Methodology

This study utilizes Python software with the assistance of Google Collab. The proposed methodology is Hybrid CNN-SVM with Borderline-SMOTE for the classification of unhealthy cabbage plants images. The pre-classification steps are shown below.

(1) Preprocessing the image data, which includes:
  (a) removing the background using Photoshop to highlight the cabbage plant
  (b) resizing images from the original 1600 x 1600 pixels to 256 x 256 pixels to reduce filtering time
  (c) enhancing images post-resizing to retain essential features.
  (d) segmenting images.
(2) Labeling the data into five types of cabbage pests and diseases, to be used as the dependent variable.
(3) Dividing the data into training, testing, and validation sets using an 80:10:10 ratio.

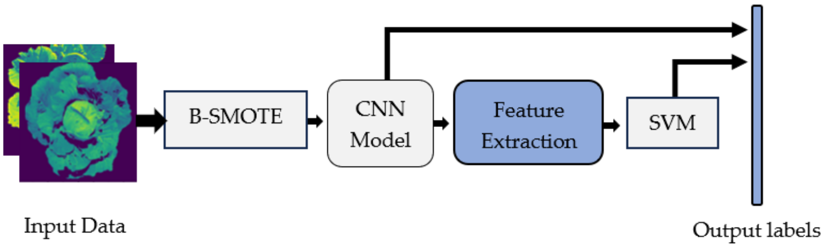

Figure 2 Proposed Classification Model

Figure 2 illustrates the classification process using CNN-SVM with Borderline SMOTE. After splitting the data, Borderline SMOTE is applied to balance the classes. Feature extraction is performed with the SqueezeNet architecture, and hyperparameters—such as epochs, batch size, and optimizer—are optimized to improve model performance. The CNN model incorporates hidden layers with max pooling, Leaky ReLU activation, and fully connected layers with softmax for classification. Fine-tuning the learning rate enhances training, testing, and accuracy measurement. CNN-extracted features are then used with an SVM classifier, optimized using Grid Search, and the results are compared to evaluate the approach's effectiveness.

Before classifying the data, we need to adjust the hyperparameter to optimize the result. Table 4 shows Epochs, Batch Size, and optimizer used in the model.

Table 4 Hyperparameter for CNN

Hyperparameter	Input
Epochs	1000
Batch Size	12
Optimizer	Adam

Table 5 shows the value of each hyperparameter using in SVM and being optimized by Grid search determined by the researchers.

Table 5 Hyperparameter for SVM

Hyperparameter	Input
C	[10; 100; 1000]
Kernels	[rbf]
Gamma	[0,001; 0,0001]

IV. RESULTS AND DISCUSSIONS

A. Dataset pre-processing

Effective preprocessing of images is essential to ensure optimal input data for classification tasks. The depicted image preprocessing procedure results is shown in Figure 2 with the following explanation.

(1) Removing the background: As seen in Figure 2 (b) after vanishing unnecessary parts of the cabbage plants, resulting in improved visibility of the cabbage shape against a black background.

(2) Image Enhancement and Dimension Adjustment: Given the large size of the original images (1600x1600 pixels), resizing them to 256x256 pixels with 3 channels accelerates the modeling process while maintaining essential details.

(3) Segmentation: Segmentation is performed using a k-means-based method to enhance detail and convert colors into RGB format, facilitating more precise segmentation outcomes.

These preprocessing steps elevate the quality of the input data, guaranteeing that the ensuing classification model can effectively extract meaningful features and produce accurate predictions.

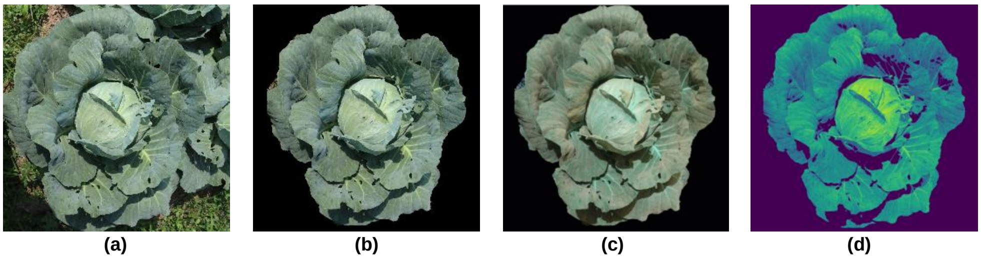

Figure 2 (a) Cabbage Image before Background Removal, (b) Cabbage Image after Background Removal, (c) Cabbage Image after Sharpening and Resizing, (d) Segmented Cabbage Image.

B. Borderline SMOTE

After data is preprocessed, the data is ready to be extracted and split into training and testing. Using equation (6) the B-SMOTE method will build more balanced data to solve the imbalance class data problem. Figure 3 portrays the distribution of data after its division into training and testing subsets. As we can see, the data distribution became more balanced in the training set after applying B-SMOTE. However, the test set remains imbalanced even after applying B-SMOTE. This adjusted training data is then fed into the CNN for further processing and model training.

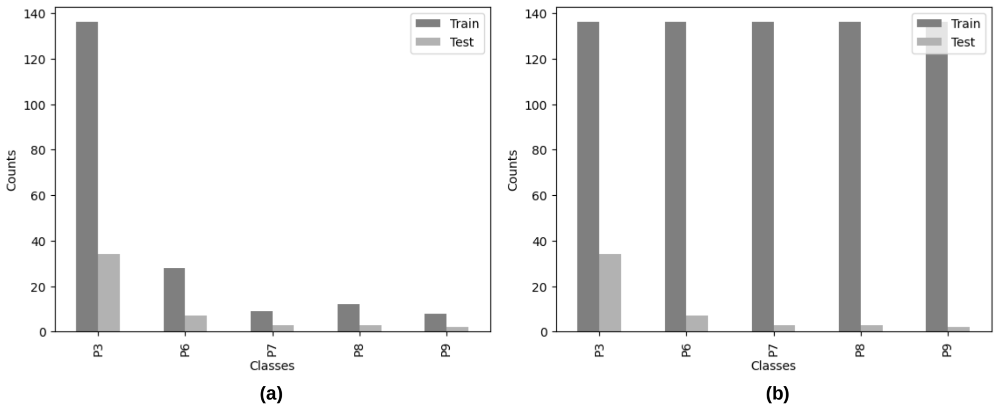

Figure 3 (a) Original dataset, (b) B-SMOTE dataset

C. Results and Observation

The best model is evaluated by comparing between using B-SMOTE and without B-SMOTE to acknowledge improvement of the proposed model. Table 6 presents a comparison of loss between the CNN using B-SMOTE and without B-SMOTE.

Table 6 Comparison Loss

Method	Loss
CNN without B-SMOTE	0,716
CNN with B-SMOTE	0,841

Based on the data in Table 6, it's evident that the CNN without B-SMOTE exhibits a lower loss value of 0,716 in comparison to the CNN without B-SMOTE, which has a loss value of 0,841. With the higher loss value, the CNN with B-SMOTE may have gained benefits such as improved robustness to class imbalances, enhanced generalization to unseen data, or better resilience against overfitting.

To assess the generalization of the model, the accuracies of both models will be compared based on their training and validation test results. **Table 7** shows the comparison of accuracy levels between the training and validation datasets.

Table 7 Comparison Accuracy in Training and Validation

	Method	Training	Validation
without B-SMOTE	CNN	0,80	0,79
	CNN + SVM	0,88	0,77
with B-SMOTE	CNN	0,77	0,78
	CNN + SVM	0,99	0,87

The results show that the hybrid models effectively categorized nearly all classes during training. The CNN + SVM model with B-SMOTE achieved the highest training accuracy at 99%, with a corresponding validation accuracy of 87%. In contrast, the CNN+SVM model without B-SMOTE had the lowest accuracy, with 88% during training, with a validation accuracy of 77%. Notably, both the CNN models with and without B-SMOTE displayed comparable accuracy between the training and validation phases. This suggests that the small gap between training and validation indicates good generalization, while a larger gap suggests potential overfitting. Models trained with B-SMOTE perform better overall, showing its importance in handling imbalanced data with higher accuracy.

To evaluate the misclassification in each class, here shown in **Table 8,** is the confusion matrix of the CNN-SVM model with B-SMOTE resampling and without resampling.

Table 8 Confusion Matrix of The Model

Model	Without B-SMOTE					With B-SMOTE				
	P3	P6	P7	P8	P9	P3	P6	P7	P8	P9
CNN	**165**	5	0	0	0	**163**	4	0	0	0
	3	**30**	0	1	1	6	**29**	0	0	0
	0	0	**12**	0	2	0	0	**12**	0	0
	0	0	0	**15**	0	0	0	0	**15**	0
	0	0	0	0	**10**	2	0	0	0	**8**
CNN+ SVM	**170**	0	0	0	0	**169**	1	0	0	0
	17	**18**	0	0	0	5	**30**	0	0	0
	6	0	**6**	0	0	0	0	**12**	0	0
	4	0	0	**11**	0	0	0	0	**15**	0
	4	0	0	0	**6**	0	0	0	0	**8**

The confusion matrix compares the CNN-SVM model's performance with and without B-SMOTE resampling across five classes. Without B-SMOTE, class P3 has 165 and 170 correct predictions in both model but several misclassifications into other classes. With B-SMOTE, class P3 and class P6 improve, showing fewer errors and more correct classifications, particularly for the underrepresented classes like P8, which goes from 6 correct predictions to 8. Misclassifications in other classes still occur but at lower frequencies. The frequent misclassifications in the original model suggest that certain classes are more prone to confusion, likely due to imbalance, while the rarer misclassifications, especially after resampling, indicate that B-SMOTE helps balance the dataset, improving the model's ability to correctly classify minority classes.

For further evaluation, the performance is assessed using the proposed method outlined in **Table 8** below.

Table 9 Comparison of F1 Score and Balanced Accuracy

	Method	F1	Balanced Accuracy
without B-SMOTE	CNN	0,78	0,94
	CNN+SVM	0,88	0,93
with B-SMOTE	CNN	0,94	0,97
	CNN + SVM	0,95	0,96

The evaluation in **Table 9** shows that models using B-SMOTE perform better than those without it, especially in handling imbalanced data. Without B-SMOTE, the CNN+SVM model still does well with an F1 score of 88% and balanced accuracy of 93%, but the CNN model alone struggles more, with a lower F1 score of 78%. This suggests that CNN has trouble with class imbalance when B-SMOTE isn't used. When B-SMOTE is applied, both models improve significantly. The CNN+SVM model with B-SMOTE achieves the highest performance with a 95% F1 score and 96% balanced accuracy. The CNN model also sees a big boost, reaching a 94% F1 score and 97% balanced accuracy. This shows that B-SMOTE helps both models better handle imbalanced data, especially in balancing the detection of affected and unaffected plants. In summary, B-SMOTE improves the classification ability of CNN and CNN+SVM models, making them much more effective at handling imbalanced data, as seen in their higher F1 scores and balanced accuracies.

V. CONCLUSIONS AND SUGGESTIONS

Overall, the integration of Borderline-SMOTE with CNN and SVM hybrid models significantly improves the classification performance on imbalanced cabbage image datasets. While models with B-SMOTE may exhibit slightly higher loss values, they consistently achieve better accuracy, F1 scores, and enhanced sensitivity, particularly in distinguishing minority classes such as pest-affected cabbages. The application of B-SMOTE helps mitigates misclassifications, leading to more reliable and balanced classification outcomes. This confirms the effectiveness of the proposed hybrid approach in handling real-world data challenges, aligning with the research aim of developing a more accurate system for identifying diseases and pests in cabbages. Future research could explore additional techniques such as data augmentation or alternative imbalance correction methods to further optimize performance and improve sensitivity in models without B-SMOTE.

REFERENCES

[1] M. F. 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.

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

[3] W. Feng, W. Huang, and J. Ren, 'Class imbalance ensemble learning based on the margin theory', *Appl. Sci.*, vol. 8, no. 5, 2018.

[4] D. Dablain, K. N. Jacobson, C. Bellinger, M. Roberts, and N. V Chawla, 'Understanding CNN Fragility When Learning With Imbalanced Data', *Mach. Learn.*, pp. 1–26, 2023, [Online]. Available: https://doi.org/10.1007/s10994-023-06326-9

[5] L. Yu and N. Zhou, 'Survey of Imbalanced Data Methodologies', *arXiv Prepr.*, 2021.

[6] P. U. Neetha *et al.*, 'Borderline-DEMNET for Multi-Class Alzheimer's Disease Classification', *5th IEEE Int. Conf. Cybern. Cogn. Mach. Learn. Appl. ICCCMLA 2023*, no. 979, pp. 192–197, 2023, doi: 10.1109/ICCCMLA58983.2023.10346937.

[7] J. N. Lee and J. Y. Lee, 'An Efficient SMOTE-Based Deep Learning Model for Voice Pathology Detection', *Appl. Sci.*, vol. 13, no. 6, 2023, doi: 10.3390/app13063571.

[8] N. H. A. Malek, W. F. W. Yaacob, Y. B. Wah, S. A. Md Nasir, N. Shaadan, and S. W. Indratno, 'Comparison of ensemble hybrid sampling with bagging and boosting machine learning approach for imbalanced data', *Indones. J. Electr. Eng. Comput. Sci.*, vol. 29, no. 1, pp. 598–608, 2023.

[9] A. Bansal, R. Sharma, A. K. Jain, V. Sharma, and V. Kukreja, 'Enhancing Fashion Cloth Image Classification through Hybrid CNN-SVM Modeling:A Multi-Class Study', in *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, 2023, pp. 484–489.

[10] S. K. Baranwal, K. Jaiswal, K. Vaibhav, A. Kumar, and R. Srikantaswamy, 'Performance analysis of Brain Tumour Image Classification using CNN and SVM', in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020, pp. 537–542.

[11] T. Lee, M. Kim, and S. P. Kim, 'Data augmentation effects using borderline-SMOTE on classification of a P300-based BCI', *8th Int. Winter Conf. Brain-Computer Interface, BCI 2020*, pp. 1–4, 2020, doi: 10.1109/BCI48061.2020.9061656.

[12] A. N. Myna, M. K., M. K., R. H. S., and Y. D. Jain, 'Classification and Detection of Cabbage Leaf Diseases from Images Using Deep Learning Methods', *Autom. Control Intell. Syst.*, vol. 11, no. 1, pp. 1–7, 2023, doi: 10.11648/j.acis.20231101.11.

[13] A. Bansal, R. Sharma, A. K. Jain, V. Sharma, and V. Kukreja, 'Enhancing Fashion Cloth Image Classification through Hybrid CNN-SVM Modeling:A Multi-Class Study', *2023 Int. Conf. Sustain. Comput. Smart Syst.*, pp. 484–489, 2023.

[14] F. Fleuret, *The Little Book of Deep Learning*, vol. 1. Carouge, Switzerland: University of Geneva, 2023. [Online]. Available: https://fleuret.org/public/lbdl.pdf

[15] 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, p. 100069, 2023.

[16] J. Wang, M. A. Khan, S. Wang, and Y. Zhang, 'SNSVM: SqueezeNet-Guided SVM for Breast Cancer Diagnosis', *Comput. Mater. Contin.*, vol. 76, no. 2, pp. 2201–2216, 2023.

[17] H. Al Majzoub, I. Elgedawy, Ö. Akaydın, and M. Köse Ulukök, 'HCAB-SMOTE: A Hybrid Clustered Affinitive Borderline SMOTE Approach for Imbalanced Data Binary Classification', *Arab. J. Sci. Eng.*, vol. 45, no. 4, pp. 3205–3222, 2020.

[18] M. Grandini, E. Bagli, and G. Visani, 'Metrics for Multi-Class Classification: an Overview', pp. 1–17, 2020, [Online]. Available: http://arxiv.org/abs/2008.05756

[19] H. Byeon, 'Comparing the Balanced Accuracy of Deep Neural Network and Machine Learning for Predicting the Depressive Disorder of Multicultural Youth', *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 584–588, 2021, doi: 10.14569/IJACSA.2021.0120668.

© 2024 by the authors. This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (http://creativecommons.org/licenses/by-sa/4.0/).