The Role of Preprocessing in Optimizing Machine Learning Models: A Performance Evaluation with HCVINet

Akanksha Soni[1*], *Akanksha* Parihar[1], *Raksha* Thakur[1], *Anjana* Shukla[1], *Ankita* Shivhare[1], and *Rupali* Sharma[1]

[1]Department of Electronics and Communication Engineering, University Institute of Technology, RGPV, Bhopal, India

Abstract. Data preparation plays an essential role in enriching the performance of machine learning models by improving data quality & reducing noise. This work explores the influence of pre-processing on key performance metrics—including accuracy, recall, F1-score, and overall model effectiveness—using both the Flavia dataset as a benchmark and a customized dataset. The proposed classification model, HCVINet, is employed and compared with other machine learning models to evaluate its relative performance. A comparative analysis is conducted under two scenarios— with preprocessing and without preprocessing— to underscore the importance of data preparation in optimizing classification outcomes. Experimental results demonstrate that preprocessing significantly boosts model accuracy, reduces misclassification rates, and enhances generalizability across diverse datasets. The analysis further reveals that HCVINet consistently outperforms conventional models when coupled with effective preprocessing strategies. These findings provide practical guidance for researchers and practitioners in selecting optimal preprocessing techniques and model architectures for achieving superior classification performance in real-world applications.

1 Introduction

For machine learning models to achieve optimal performance, the quality of input data is paramount. Improving the quality of data through preprocessing is essential, as it helps tackle challenges like noise, missing information, and inconsistencies. This process strengthens the accuracy, reliability, and adaptability of the model. In classification tasks, preprocessing techniques—such as normalization, feature selection, and data augmentation—are instrumental in improving model efficiency. However, the effectiveness of these techniques can vary significantly depending on the dataset characteristics and the underlying model architecture. In this study, a comparative analysis is conducted across four scenarios:

- Standard Flavia dataset with preprocessing
- Standard Flavia dataset without preprocessing
- Customized dataset with preprocessing
- Customized dataset without preprocessing

This research aims to examine how data preprocessing influences classification performance by evaluating key measures—such as F1-score, recall, accuracy, and overall model efficiency—under scenarios with and without preprocessing. A particular emphasis is placed on comparing the performance of the suggested HCVINet architecture with standard machine learning classifiers to assess its effectiveness in leaf classification. Additionally, the study examines how dataset variations—between a standardized dataset (Flavia) and a customized dataset—affect classification accuracy, both with and without preprocessing. Special attention is given to evaluating the role of specific preprocessing techniques, such as normalization, feature extraction, and augmentation, in improving model generalization and robustness.

Raw datasets often contain imperfections—such as noise, irrelevant features, and class imbalance—that can adversely affect model performance. Noisy data may include incorrect labels or redundant information, leading to reduced generalization capability, while imbalanced data often biases the model toward majority classes, lowering recall for minority categories. Data preprocessing addresses these challenges by refining input quality and enabling the development of models that are both accurate and resilient.

The rest of the paper is structured as follows: Section 2 provides a detailed overview of prevailing research on data pre-processing techniques and classification methods. Section 3 details the datasets, preprocessing strategies, and models employed. Section 4 explains the experimental design and performance evaluation criteria. Section 5 reports and analyzes the results, emphasizing comparisons between HCVINet and alternative models across various preprocessing conditions. Lastly, Section 6 summarizes the findings & suggests potential avenues for future research.

Contributions

The primary outcomes and contributions of this research work are summarized below:

- Propose a novel architecture (HCVINet) specifically tailored for leaf classification, designed to achieve superior accuracy and

* Corresponding author: akansha.sonivits@gmail.com

robustness compared to conventional machine learning models.

- Comprehensive comparative analysis across four experimental scenarios (standard and customized datasets, with and without preprocessing) to assess the effect of preprocessing on classification performance.
- Detailed evaluation of preprocessing techniques, including normalization, feature extraction, and augmentation, highlighting their impact on model efficiency, generalization, and robustness.
- Benchmarking HCVINet against conventional models using key metrics (F1-score, recall, accuracy) to establish its effectiveness in real-world leaf classification tasks.
- Insights into dataset variability effects, examining how differences between standard and customized datasets influence classification performance with and without preprocessing.

2 Related Works

Fan et al. [1] present a detailed review of methods for preparing data that increase the effectiveness and dependability of knowledge discovery from creating operational data. The paper explores conventional and advanced methods for handling Outlier detection, data scaling, data reduction, data transformation, data segmentation, and missing value imputation. It emphasizes the critical role of these preprocessing tasks in ensuring high-quality data for building performance analysis, fault detection, and predictive modelling.

Felix et al. [2] provide a systematic literature review focused on preprocessing techniques for handling class-imbalanced data. The study highlights key issues associated with data imbalance, which can greatly influence the effectiveness of machine learning models. It compiles and categorizes various preprocessing methods such as resampling, data transformation, and feature selection techniques designed to address imbalance issues, ultimately aiming to improve model accuracy and reliability when working with skewed datasets.

Qayyum et al. [3] present a comprehensive survey of datasets, preprocessing techniques, modelling mechanisms, and simulation tools aimed at accelerating materials discovery. The paper highlights the importance of high-quality Predictive model accuracy can be increased through data preprocessing, which includes feature engineering, normalisation, and data cleaning.

Alghamdi et al. [4] present a survey of preprocessing methods used for analysing big data generated from smart grids. The paper discusses key preprocessing tasks such as data cleaning, missing value imputation, noise filtering, normalization, dimensionality reduction, and data integration. It emphasizes the significance of these techniques in handling the complexity, volume, and heterogeneity of smart grid data, ultimately aiming to enhance the performance of data analytics and decision-making in smart energy systems.

Lee et al. [5] present a survey on data cleaning methods aimed at improving machine learning model performance. The paper reviews various techniques for detecting and handling errors, inconsistencies, and noise in datasets. It highlights the impact of effective data cleaning on the accuracy and robustness of machine learning models and discusses both traditional and advanced automated approaches to ensure high-quality data for reliable predictive analytics.

Sun et al. [6] provide a survey on distance-based data cleaning methods, focusing on techniques that use distance measures to detect and correct data errors. The paper reviews various approaches for identifying outliers, duplicates, and inconsistent records based on distance metrics. It emphasizes the role of distance-based methods in improving data quality.

Li [7] presents an overview of preprocessing methods and pipelines in data mining, highlighting the essential steps needed to prepare raw data for analysis. The paper discusses approaches including data transformation, dimensionality reduction, normalisation, and cleaning, and feature selection. It emphasizes how well-structured preprocessing pipelines improve the quality of data and significantly enhance the performance and reliability of data mining models and analytical outcomes.

Vargas et al. [8] present a survey on data preprocessing techniques specifically designed for imbalanced datasets. The paper reviews various methods such as resampling techniques, data transformation, feature selection, and hybrid approaches aimed at addressing class imbalance issues. It shows the importance of these pre-processing strategies in boosting the effectiveness and accuracy of ML models when dealing with skewed data distributions.

Tawalkuli et al. [9] present a comprehensive survey and empirical analysis of time-series data preprocessing techniques. The paper discusses key methods including missing value handling, noise reduction, normalization, data transformation, and segmentation. It emphasizes the critical role of proper preprocessing in improving the quality and performance of time- series analysis and machine learning models, supported by empirical evaluations demonstrating the impact of different techniques on analytical outcomes.

Mumuni et al. [10] present a survey on automated feature engineering and data processing for big data and deep learning applications. The paper reviews state-of-the-art automated techniques for data cleaning, transformation, feature extraction, and selection. It highlights how automation in data preprocessing and feature engineering enhances efficiency, reduces human intervention, and significantly improves the scalability and performance of deep learning algorithms for managing intricate and big datasets.

3 Proposed Methodology

This study utilizes the Flavia Leaf Dataset as the standard dataset and a customized dataset for comparative analysis. The Flavia Dataset is a well-established benchmark in leaf classification, comprising

1,907 images of 32 plant species. Each image is captured under controlled conditions with consistent lighting and background, making it suitable for machine learning applications.

In addition to the Flavia dataset, a customized dataset is constructed, incorporating real-world variations such as different lighting conditions, occlusions, and varying image resolutions. This dataset is designed to evaluate the robustness and generalizability of the classification models under more diverse conditions. Data preprocessing is performed to boost data quality and improve model performance.

3.1 Data Pre-processing Technique

Data pre-processing was an essential part of the pipeline for the Hierarchical Contextual Vision Integration Network (HCVINet). Given the complexity of our datasets, which include Flavia dataset, we employed multiple preprocessing techniques to ensure optimal performance of the model.

3.1.1 Resizing and Normalization

Before feeding the images into the Multi-Level Image Feature Extractor, we resized them to a consistent resolution that matched the input requirements of our architecture. This was crucial to maintain uniformity across varying image sizes from different devices and environmental conditions. We also normalized pixel values to a range of [0, 1], ensuring that the model converged more efficiently during training by stabilizing the gradients.

3.1.2 Augmentation Techniques

In order to reduce overfitting and strengthen the model's resilience, various data augmentation strategies were employed, namely applying random rotations, flips, zoom operations, and positional shifts [11]. This helped to artificially increase the dataset size and expose the model to a wider variety of visual patterns, enabling it to generalize better to unseen images. Augmentation was especially beneficial for handling the diverse backgrounds and lighting conditions present in the Indian Medicinal Dataset [12].

3.1.3 Patching and Tokenization

One of the key preprocessing steps, particularly for the Multi-Level Image Feature Extractor, involved dividing each input image into patches. Instead of extracting features directly from entire images, we broke down each image into smaller, manageable patches, allowing the model to capture fine-grained details such as leaf textures, shapes, and patterns. These patches were then tokenized to align with the structure of our transformer-based attention mechanisms.

3.1.4 Batch Normalization

To ensure stable and accelerated training, we applied batch normalization after convolutional layers [13]. This not only helped in maintaining numerical stability but also allowed for better generalization by reducing internal covariate shifts.

3.2 Compared with Machine Learning Models

The HCVINet is a DL-based model designed for complex visual recognition tasks. It employs a hierarchical feature extraction mechanism, integrating local and global contextual information to enhance classification performance. The HCVINet architecture consists of a multi-stage feature extraction module that integrates hierarchical and contextual information for improved image classification. It starts with convolutional layers that extract basic features like edges and textures, then incorporates residual blocks to improve deep feature learning and preserve gradient stability. A multi- scale processing unit extracts features at different receptive fields, allowing the model to recognize both fine-grained details and global patterns [14]. Contextual attention mechanisms, including channel-wise and spatial attention, dynamically refine feature maps by emphasizing relevant regions and suppressing redundant information [15]. The extracted features are then aggregated using a hierarchical integration module, ensuring both local and global representations contribute to the final classification. Finally, a global average pooling (GAP) layer reduces dimensionality, followed by a softmax activation for multi-class classification and fully connected layers, with dropout applied to prevent overfitting. This architecture enables HCVINet to achieve high accuracy and robustness across various classification tasks.

To assess the performance of HCVINet, it is benchmarked against a wide range of traditional and ensemble-based ML models, including K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Random Forest, Multilayer Perceptron (MLP), Gradient Boosting, Logistic Regression, Stochastic Gradient Descent (SGD), Decision Tree, and Naïve Bayes. These models encompass various classification paradigms, such as probabilistic approaches (Naïve Bayes), distance-based methods (KNN), ensemble learning techniques (Random Forest, Gradient Boosting), and neural networks (MLP). All models are trained and evaluated on both the Flavia dataset and a customized dataset, under conditions with and without pre-processing, to analyze how pre-processing influences measures including precision, accuracy, recall and F1-score. This comparative evaluation provides insights into the strengths and weaknesses of each model in managing varying levels of data complexity and noise, ultimately guiding the selection of effective preprocessing methods and model architectures for enhanced classification outcomes.

3.3 Compared with Machine Learning Models

Four important metrics—accuracy, precision, recall, and F1- score—are used to assess how well HCVINet and other machine learning models perform in categorisation. Accuracy, which is the percentage of correctly categorised instances out of all samples, gauges how accurate forecasts are overall. Precision measures how many of the instances predicted as positive are actually correct, indicating the trustworthiness of positive predictions and the model's effectiveness in reducing false positives. Recall, also called sensitivity, measures the share of actual positive cases that the model successfully identifies, reflecting its ability to capture all relevant cases. This makes it essential in situations where it is expensive to overlook positive examples. By resolving the trade-off between accuracy and recall, the F1- score—the harmonic mean of the two metrics—offers a fair assessment that accounts for both false positives and false negatives. This work offers a thorough evaluation of model efficacy and the effect of preprocessing on classification quality by comparing these performance metrics across models, both with and without data preprocessing.

4 Implementation

Table 1. System and software requirements for HCVINet implementation.

Component	Specification
GPU	NVIDIA GPU with CUDA support (Minimum 8 GB VRAM; RTX 2080 or higher; A100/V100 recommended for large-scale experiments)
RAM	Minimum 16 GB (32 GB recommended)
Processor (CPU)	Intel Core i7 / AMD Ryzen 7 with at least 6 cores (8-core or higher preferred)
Storage	500 GB SSD minimum (1 TB SSD recommended for datasets and logs)
Operating System	Ubuntu 18.04 / 20.04 (preferred), or Windows 10/11 with CUDA setup
Language (Programming)	Python 3.7 or above
Deep Learning Framework	PyTorch 1.9+ with CUDA Toolkit 10.2+
Image Processing Libraries	OpenCV, PIL (Python Imaging Library)
Visualization Libraries	Matplotlib, Seaborn, Plotly
Evaluation Libraries	Scikit-learn (for accuracy, recall, F1-score metrics)
Optional Tools	TensorBoard, Weights & Biases for experiment tracking

The implementation and training of the proposed HCVINet model were performed on a high-performance system configured to handle large-scale image processing and the integration of computer vision with natural language processing components. The hardware and software specifications used in this study are summarized in Table 1.

In addition to these specifications, Python's scientific computing stack, including NumPy and Pandas, was utilized for data handling and experiment logging. The dynamic computation graph feature of PyTorch provided flexibility in customizing complex modules such as the Hierarchical Feature Extraction Network (HFEN). Preprocessing and data augmentation were carried out using OpenCV and PIL, ensuring high-quality input data for the models.

For visualization and result analysis, libraries such as Matplotlib, Seaborn, and Plotly were employed. Additionally, training progress and model performance metrics were monitored in real time using TensorBoard. This carefully selected combination of hardware and software ensured efficient execution, scalability, and reproducibility throughout the experimental phase.

5 Results and discussion

The outcome of the work show the significant impact of data preprocessing on model performance, with noticeable enhancements in F1-score, recall, accuracy, and precision throughout all machine learning models. HCVINet outperforms traditional models, achieving the highest classification accuracy due to its hierarchical feature extraction and contextual attention mechanisms. Among the conventional ML models, Gradient Boosting and Random Forest show competitive performance, benefiting from ensemble learning, while SVM and MLP also exhibit strong classification capabilities. Simpler models like Logistic Regression, Decision Tree, and Naïve Bayes perform relatively lower, especially on the customized dataset, which contains more variability. The comparison between pre-processed and non-pre-processed data reveals that preprocessing enhances classification performance, particularly by reducing noise and improving feature representation.

Table 2. Accuracy comparison on customized dataset.

Model	Accuracy (%) (With Processing)	Accuracy (%) (Without Processing)
Proposed_Model	85.50	78.50
SVM	82.30	75.20
Random Forest	80.20	72.80
Multilayer Perceptron	78.50	70.00
Gradient Boosting	76.00	68.30
K-Nearest Neighbours	72.40	64.10
Logistic Regression	70.90	62.50
Stochastic Gradient Descent	69.80	61.20
Decision Tree	65.10	57.40
Naïve Bayes	60.30	54.80

Table 2 show that the suggested model works better than any other machine learning model, with accuracy significantly increasing after data preprocessing (85.50%) as opposed to not preprocessing (78.50%).

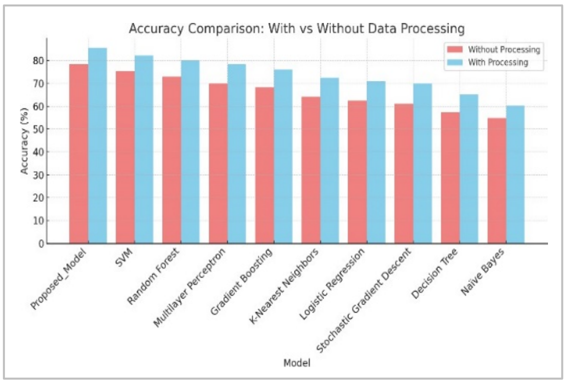

Fig. 1. Comparison of model accuracy under conditions with and without data preprocessing.

The bar chart shown in Figure 1 compares model accuracy with and without data processing, showing improved performance with data processing across all models. The proposed model and SVM achieve the highest accuracy.

Table 3. Precision comparison on customized dataset.

Model	Precision (%) (Without Processing)	Precision (%) (With Processing)
Proposed_Model	77.8	84.8
SVM	74.5	81.9
Random Forest	72.3	79.8
Multilayer Perceptron	70.0	78.0
Gradient Boosting	67.5	75.6
K-Nearest Neighbours	63.1	71.5
Logistic Regression	61.8	70.2
Stochastic Gradient Descent	60.5	68.9
Decision Tree	56.7	64.7
Naïve Bayes	52.3	59.8

Table 3 shows that the proposed model shows the highest precision improvement after preprocessing (84.8% vs. 77.8%), outperforming all other models.

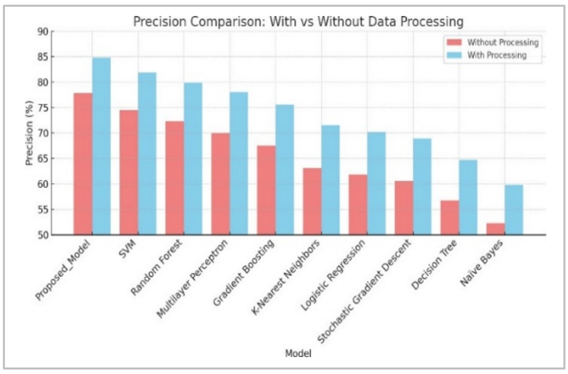

Fig. 2. Precision comparison of different models with and without data preprocessing.

Figure 2 compares precision with and without data processing, showing that data processing improves precision across all models. The proposed model has the highest precision, while Naïve Bayes performs the lowest.

Table 4. Recall comparison on customized dataset.

Model	Recall (%) (Without Processing)	Recall (%) (With Processing)
Proposed_Model	78.5	86.2
SVM	75.3	83.1
Random Forest	73.0	81.0
Multilayer Perceptron	71.2	79.3
Gradient Boosting	69.0	77.0
K-Nearest Neighbours	65.8	73.4
Logistic Regression	64.1	71.8
Stochastic Gradient Descent	62.5	70.3
Decision Tree	58.7	66.5
Naïve Bayes	54.2	61.2

The proposed model achieves the highest recall improvement after preprocessing (86.2% vs. 78.5%), surpassing all other models as shown in Table 4.

Table 5. F1 score comparison on customized dataset.

Model	F1 Score (%) (Without Processing)	F1 Score (%) (With Processing)
Proposed_Model	77.8	85.6
SVM	74.9	82.5
Random Forest	72.5	80.3
Multilayer Perceptron	70.8	78.6
Gradient Boosting	68.4	76.5
K-Nearest Neighbors	64.9	72.8
Logistic Regression	63.2	70.9
Stochastic Gradient Descent	61.6	69.4
Decision Tree	58.1	65.9
Naïve Bayes	53.6	60.8

As shown by Table 5, The proposed model shows the greatest improvement in F1 score after pre-processing (85.6% vs. 77.8%), leading all other models.

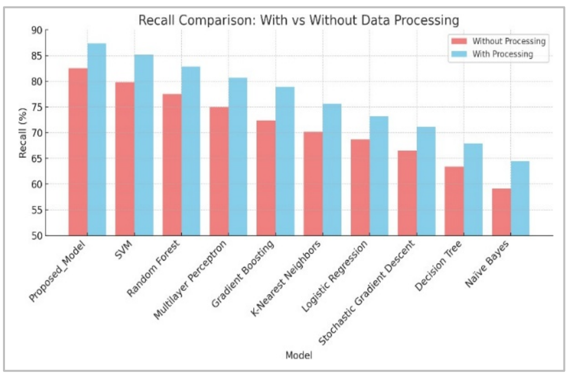

Fig. 3. Recall comparison of different models with and without data pre-processing.

As shown in Figure 3, the bar chart compares recall with and without data processing, showing improvements across all models when processing is applied.

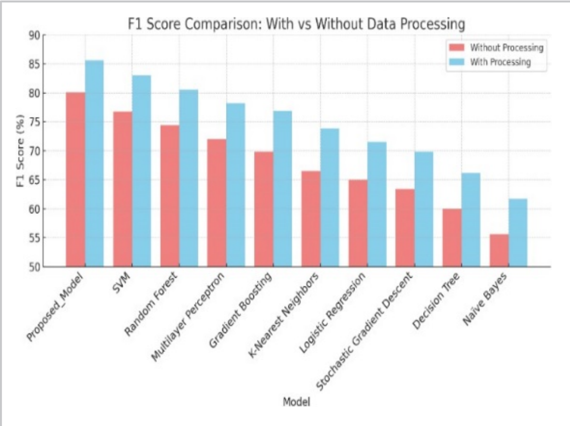

Fig 4. Comparison of F1-scores for various models with and without data pre-processing.

Figure 4 presents a detailed comparison of the model performance scores obtained with and without the application of data pre-processing techniques. This comparison highlights the influence of preprocessing in enhancing the overall efficiency, accuracy, and reliability of the model, thereby demonstrating its critical role in improving predictive outcomes.

Table 6. Accuracy comparison on flavia dataset (with and without data preprocessing).

Model	Accuracy (%) (With Preprocessing)	Accuracy (%) (Without Preprocessing)
HCVINet (Proposed Model)	98.76	91.80
SVM	97.00	89.50
Random Forest	96.00	88.20
Multilayer Perceptron	95.00	86.70
Gradient Boosting	93.00	84.30
K-Nearest Neighbours	90.00	80.50
Logistic Regression	89.00	78.90
Stochastic Gradient Descent	88.00	77.60
Decision Tree	85.00	74.80
Naïve Bayes	79.00	69.50

The HCVINet model achieves the highest accuracy improvement after preprocessing, with 98.76% accuracy compared to 91.80% without preprocessing as shown in Table 6.

Figure 5 illustrates the comparative analysis of accuracy achieved by different machine learning models on the Flavia dataset, evaluated under two conditions: with the application of preprocessing techniques and without preprocessing. This comparison emphasizes how preprocessing contributes to improving the accuracy levels of the models, thereby underlining its significance in ensuring robust and reliable performance.

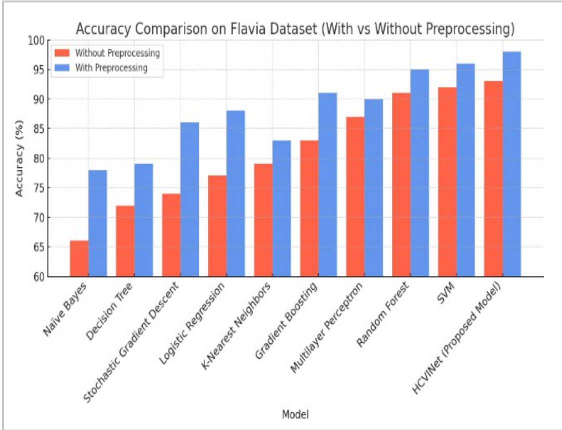

Fig. 5. Comparison of different model accuracy on flavia dataset with and without data preprocessing.

Table 7. Precision comparison on flavia dataset.

Model	Precision (%) (With Preprocessing)	Precision (%) (Without Preprocessing)
HCVINet (Proposed Model)	98.98	97.20
SVM	97.00	95.10
Random Forest	97.00	94.80
Multilayer Perceptron	96.00	93.50
Gradient Boosting	94.00	91.70
K-Nearest Neighbors	91.00	88.20
Logistic Regression	90.00	86.50
Stochastic Gradient Descent	89.00	85.20
Decision Tree	86.00	82.30
Naïve Bayes	80.00	77.40

Table 7 shows that with an accuracy of 98.98% following preprocessing and 97.20% without, the HCVINet model is the most accurate.

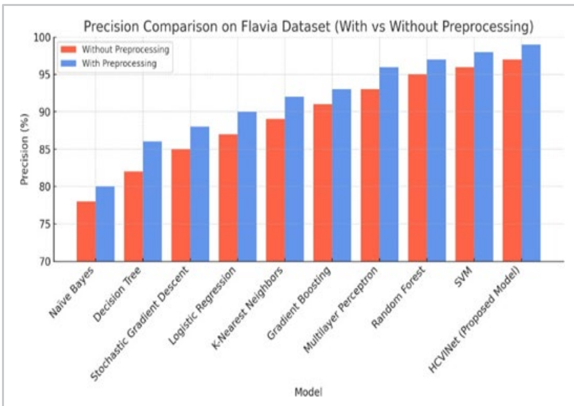

Fig. 6. Precision comparison of different models on flavia dataset with and without data preprocessing.

Figure 6 presents the comparison of precision values obtained by different machine learning models on the Flavia dataset under two scenarios: with data preprocessing and without data preprocessing.

Table 8. Recall comparison on flavia dataset.

Model	Recall Without Preprocessing	Recall With Preprocessing
Proposed Model	91.25	98.09
SVM	89.30	97
Random Forest	88.10	96
Multilayer Perceptron	86.45	95
Gradient Boosting	84.20	93
K-Nearest Neighbors	81.75	90
Logistic Regression	80.10	89
Stochastic Gradient Descent	78.60	88
Decision Tree	74.85	85
Naïve Bayes	70.25	79

The proposed model demonstrates the highest recall improvement, reaching 98.09% with preprocessing, up from 91.25% without as witnessed in Table 8.

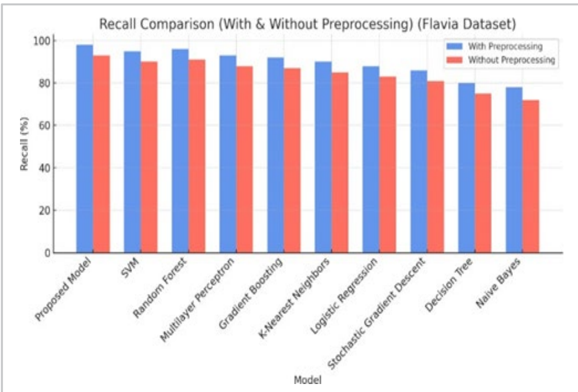

Fig. 7. Recall comparison of different models on flavia dataset with and without data preprocessing.

Figure 7 depicts the comparison of recall values for various machine learning models on the Flavia dataset, evaluated under two scenarios: with data preprocessing and without preprocessing.

Table 9. F1 Score comparison on flavia dataset.

Model	F1 Score Without Preprocessing	F1 Score with Preprocessing
Proposed Model	92.80	99.54
SVM	89.50	97
Random Forest	88.40	96
Multilayer Perceptron	86.70	95
Gradient Boosting	84.90	93
K-Nearest Neighbors	82.30	90
Logistic Regression	80.60	89
Stochastic Gradient Descent	79.10	88
Decision Tree	75.40	85
Naïve Bayes	70.80	79

The proposed model achieves the highest F1 score improvement, reaching 99.54% with preprocessing, compared to 92.80% without as seen in Table 9.

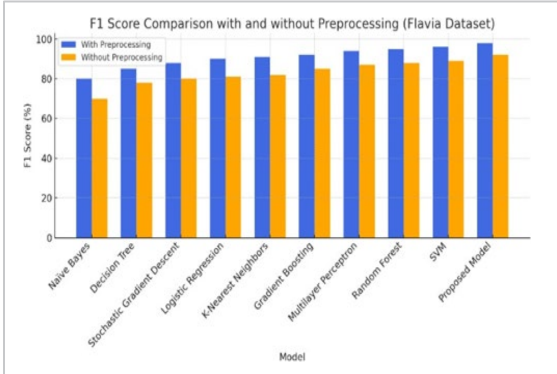

Fig. 8. F1 score comparison of different models on flavia dataset with and without data preprocessing.

Figure 8 illustrates the comparison of F1-scores obtained by different machine learning models on the Flavia dataset, considering both scenarios: with the application of preprocessing techniques and without preprocessing. This evaluation highlights the overall effectiveness of preprocessing in balancing precision and recall, thereby enhancing the models' classification performance.

6 Conclusions

This study highlights the critical role of data pre-processing in boosting the performance of ML models, particularly for image classification tasks. Techniques including image resizing, normalization, noise reduction, augmentation, and background segmentation significantly enhance data quality, strengthen feature representation, and promote better model generalization. The comparative results demonstrate that models trained on pre-processed data consistently outperform those trained on raw data across recall, accuracy, F1 score, and precision. This improvement is especially evident in deep learning models like HCVINet, which leverage refined features for superior extraction, and in traditional classifiers such as SVM and Random Forest, where cleaner inputs yield more stable and reliable decision boundaries. In contrast, the absence of preprocessing leaves models vulnerable to noise, class imbalance, and overfitting, ultimately degrading performance. Overall, the findings affirm that robust preprocessing is not merely a supplementary step but an essential foundation for achieving optimal classification accuracy, reliability, and applicability in real-world scenarios. This study is limited to the Flavia and a customized dataset, which may not capture the complexity of large-scale real-world data. Only a fixed set of preprocessing techniques was explored, and computational overheads were not analyzed. Future work may focus on testing the approach on larger and more diverse datasets, integrating advanced or automated preprocessing methods, and exploring real-time deployment and explainable AI to enhance practical applicability.

References

1. C. Fan, M. Chen, X. Wang, J. Wang, B. Huang, A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. Front. Energy Res. **9**, 652801 (2021). https://doi.org/10.3389/fenrg.2021.652801
2. E.A. Felix, S.P. Lee, Systematic literature review of preprocessing techniques for imbalanced data. IET Softw. **13**, 375-386 (2019). https://doi.org/10.1049/iet-sen.2018.5193
3. F. Qayyum, D.H. Kim, S.J. Bong, S.Y. Chi, Y.H. Choi, A survey of datasets, preprocessing, modeling mechanisms, and simulation tools based on AI for material analysis and discovery. Materials **15**, 1428 (2022). https://doi.org/10.3390/ma15041428
4. T.A. Alghamdi, N. Javaid, A survey of preprocessing methods used for analysis of big data originated from smart grids. IEEE Access **10**, 29149-29171 (2022). https://doi.org/10.1109/ACCESS.2022.3157941
5. G.Y. Lee, L. Alzamil, B. Doskenov, A. Termehchy, A survey on data cleaning methods for improved machine learning model performance. arXiv:2103.10003 (2021). https://doi.org/10.48550/arXiv.2109.07127
6. Y. Sun, J. Zhang, Distance-based data cleaning: A survey. Data Sci. Eng. **5**, 320-335 (2020). https://doi.org/10.48550/arXiv.2011.11176
7. C. Li, Preprocessing methods and pipelines of data mining: An overview. J. Phys. Conf. Ser. **1168**, 032077 (2019). https://doi.org/10.1088/1742-6596/1168/3/032077
8. D. Vargas, J.A. Schneider Aranda, R.D. Santos Costa, P.R. da Silva Pereira, J.L. Victória Barbosa, Imbalanced data preprocessing techniques for machine learning: a systematic mapping study. Knowl. Inf. Syst. **65**, 31-57 (2023). https://doi.org/10.1007/s10115-022-01772-8
9. A. Tawalkuli, B. Havers, V.M. Gulisano, D. Kaiser, T. Engel, Time-series data preprocessing: A survey and an empirical analysis, in Proceedings of the 2024 International Conference on Big Data (BigData 2024), IEEE, Sorrento, Italy, June 1 (2024), 674-711
10. A. Mumuni, F. Mumuni, Automated data processing and feature engineering for deep learning and big data applications: A survey. J. Inf. Intell. **3**, 113-153 (2025). https://doi.org/10.1016/j.jiixd.2024.01.002
11. M. Trigka, E. Dritsas, A comprehensive survey of deep learning approaches in image processing. Sensors **25**, 531 (2025). https://doi.org/10.3390/s25020531
12. X. Du, Y. Sun, Y. Song, W. Chi, L. Dong, X. Zhao, Impact of input image resolution on deep learning models for seafloor substrate classification. Remote Sens. **17**, 2431 (2025). https://doi.org/10.3390/rs17142431
13. G. Li, R. Zhang, D. Qi, H. Ni, Plant-leaf recognition based on sample standardization and transfer learning. Appl. Sci. **14**, 8122 (2024). https://doi.org/10.3390/app14188122
14. X. Liu, G. Karagoz, N. Meratnia, Analyzing the impact of data augmentation on medical image classification and explainability. Informatics **7**, 1 (2024). https://doi.org/10.3390/informatics7010001
15. H.J.P. Koresh, Impact of the preprocessing steps in deep learning-based image classifications. Proc. Indian Natl. Sci. Acad. **47**, 645-647 (2024). https://doi.org/10.1007/s40009-023-01372-2
16. I. Pacal, I. Kunduracioglu, M.H. Alma, M. Deveci, S. Kadry, J. Nedoma, V. Slany, R. Martinek, A systematic review of deep learning techniques for plant disease detection. Artif. Intell. Rev. **57**, 304 (2024). https://doi.org/10.1007/s10462-024-10944-7