



Kerbala Journal for Engineering Science
<https://kjes.uokerbala.edu.iq>



Deep CNN-Based Grape Image Classification with Enhanced Feature Representation

Mohammed Abdallazez Mohammed *¹

¹ Department of Computer Science, College of Computer Science and Information Technology, University of Karbala.

*Corresponding author, E-mail: mohammed.abdallazez@uokerbala.edu.iq

Received: 12 March 2026; Revised: 12 April 2026; Accepted: 29 April 2026; Published: 30 September 2026.

Abstract

Classifying grape varieties based on images of their leaves is a delicate and challenging task in image classification due to the significant similarity in shape, texture, and color among different varieties. This research proposes a hybrid framework combining machine learning and deep learning to achieve high classification performance in real-world environments. The task is formulated as a problem of multiclass classification, including eight grape classes, using a balanced dataset of 8,000 images collected from real-world agricultural environments. Preprocessing steps were implemented to improve the quality of features, including removing the background to separate the leaf from unrelated elements and edge detection to highlight the details of the structure, such as contours and veins. Deep features are then extracted by multiple pre-trained CNN models, like InceptionV3, ResNet50, VGG16, and VGG19, allowing the model to capture balancing visual representations. Principal Component Analysis is used to remove redundancy and decrease feature dimensionality, then resulting features are fused into a unified representation. Some machine learning classifiers were evaluated, including K-Nearest Neighbors, SVM, Logistic Regression, Random Forest, Decision Tree, and Naïve Bayes. The SVM model demonstrates the finest performance (71.6% accuracy) with balanced precision, F1-score, and recall. Although this accuracy is less than the accuracy described on controlled datasets, it reflects real-world data complexity. Generally, the proposed approach explained that combining deep feature extraction, dimensionality reduction, and feature fusion can efficiently enhance the performance of classification in real-world agricultural scenarios.

Keywords: Convolutional Neural Networks; Feature Extraction; Deep Learning; Computer Vision, Image Classification; Grape Leaf Classification.

1. Introduction

Grapes are the most significant horticultural crops supporting the pharmaceutical and food industries and the global agricultural economy [1]. Amplified consumer demand and enhanced living values have contributed to the expansion of grape cultivation. Grapes are rich and nutritious, containing important nutrients, making them a valuable source in the pharmaceutical industry, particularly for treating hypertension [2]. Also in the plant classification field, classifying grape varieties depends on the leaves, which present an important challenge [3]. Effective grape management necessitates identifying varieties before planting due to their diversity and varying importance and value. Therefore, classifying grape varieties based on their leaves is crucial for breeding programs and cultivation strategies [4]. The wide variety of grape varieties necessitates more precise and effective methods and techniques for accurately classifying grape leaves. [5]. The classification stages include image acquisition, preprocessing, feature extraction, and then training the machine learning model to obtain a model capable of classifying test data with high accuracy and generalizing to future data, noting that each step of the classification process greatly affects the accuracy of the final prediction [6].

The most important step in classifying plant images is feature extraction, due to its role in representing the visual information used to train machine learning models [6]. Traditional approaches rely on handcrafted feature extraction techniques such as color histograms, texture descriptors (e.g., Gray-Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP)), shape descriptors, and edge-based features to represent plant images [6], [7]. It is noteworthy that the effectiveness of these methods depends heavily on operator expertise, as they are limited when dealing with highly variable datasets [6], [8]. With the advent of deep learning, high efficiency in extracting features from images has emerged, particularly with convolutional neural networks (CNNs) [9] [10]. Unlike classical techniques, CNNs learn hierarchical feature representations in which low-level features, such as edges and textures, are progressively combined into more abstract and discriminative patterns [9], [8]. The model will improve in terms of robustness because of this hierarchical learning ability under varying environmental conditions like illumination, viewpoint changes, and background complexity [8], [7].

Thus, CNNs have outperformed traditional machine learning in terms of accuracy and performance [9], [7]. This led to the dominance of convolutional neural networks in the task of feature extraction and plant image recognition [6]. Because of their ability to learn from raw data, hierarchical feature representations automatically learned by CNNs are widely integrated into machine learning approaches, reducing the need for manual feature engineering and improving model performance, especially in image-related tasks [10]. The hierarchical feature representations are extracted by CNNs through a sequence of convolutional and pooling operations, where initial layers detect low-level patterns like edges and corners, while deeper layers learn more complex features like textures and object parts. These learned representations are progressively refined during training via backpropagation. Subsequently, fully connected layers perform high-level operations by mapping the extracted features to output targets, enabling tasks such as classification or regression. This hierarchical learning capability allows CNNs to achieve high performance without the need for manual feature engineering, whereas the final classification layers are often disregarded [10], [11], [12]. Using CNN in feature extraction will solve the challenges arising from the similarity between visually related species and the large differences in leaf shape, size, texture, growth stage, and climatic conditions, which make feature extraction from leaf

images difficult [13]. When dealing with partial agricultural images, the pre-trained convolutional neural network models improve feature robustness and reduce over-allocation [14, 15]. This research addresses the problem of classifying eight grape varieties based on leaf images. Deep learning models were employed as feature extractors, where pre-trained architectures, including VGG16, VGG19, InceptionV3, and ResNet50, were utilized to extract discriminative deep features from grape leaf images. These extracted features were then used as input to machine learning algorithms for classification. The aim of this research is to evaluate the effectiveness of deep learning-based feature extraction in generating robust representations that improve classification accuracy. This research contributes through three main aspects:

1. **First Classification on A Novel Dataset.**

This study is the first to apply machine learning and deep learning classification to the grape leaf dataset [18], as no prior research has explored classification on this dataset. This dataset has real-world grape images, balanced grape varieties, and a fine-grained grapevine dataset.

2. **Hybrid deep learning–machine learning framework**

Pre-trained CNN architectures, including VGG16, VGG19, InceptionV3, and ResNet50, were used for deep feature extraction, and these features were then employed with multiple machine learning classifiers for effective grape leaf classification.

3. **Evaluation of robust deep features for fine-grained tasks**

The research investigates the effectiveness of extracted deep features in improving classification accuracy, including fine-grained discrimination among grape varieties.

The rest of this research is ordered as tracks: Section 2 Related Work, Section 3 The Proposed Framework, Section 4 Results and Discussion, Section 5 Limitations and Future Work, and Section 6 conclude the article.

2. Related Work

Unlike disease classification, grape variety classification is more challenging due to subtle differences in color, texture, and shape between grape types. After conducting a survey to identify previous works related to grape variety classification in the dataset used in this research, no previous work classifying this dataset was found up to the date of submission. A limited number of studies are focusing specifically on grape variety classification, whereas only two previous works were found on other datasets, as in Table 1. Carneiro et al. (2025) proposed a self-supervised learning approach using Masked Autoencoders (MAE) for grapevine variety classification. Their method yielded accuracy between 95% and 98%, especially in cases with limited labeled data [16]. multi-model approaches and ensembles are applied to tasks of fruit classification, including grape varieties. Alrashdi et al. (2026) use multiple architectures, like DenseNet, ResNet, and Xception, to achieve high classification accuracy and enhanced generalization across datasets [17].

A review of previous studies reveals that the task of classifying grape leaves is still emerging and represents a challenging research task compared to the classification of plant diseases. One of the important limitations is the moderate size of datasets, which limits the model's generalization capability. While earlier work by Carneiro et al. [16] stated accuracy (95–98%) in the classification of grape variety, their model was evaluated on controlled datasets or benefited from an advanced self-supervised model.

Additionally, there are other limitations, such as unbalanced classes. The leaves are both unhealthy and healthy, and they are at a distance of between 20 and 40 cm from the plant. While, the dataset in this research is real-world images of grapes with real-world conditions, such as differences in lighting, background, and image quality. Also, samples were captured from different areas, as mentioned in [18], and only healthy leaves were captured. A balance between varieties was also considered; these factors make the classification task more challenging and closer to real-world scenarios, and leading to more realistic classification results.

Table 1: Grape Variety Classification Studies.

Ref	Study	Task Type	Balancing	Dataset Details	No. of Images	Classes	Methods	Notes
[16]	Carneiro et al. (2025)	Grape Variety Classification	Unbalanced dataset	Field images collected in vineyards, preprocessed and augmented Dataset 1: 7,939 training, 1,508 validation, 4,032 test images. Dataset 2: 4,574 training, 1,060 validation, 1,142 test images.	Dataset 1: 13,479 Dataset 2: 6,776	43 grape varieties	self-supervised learning approach using Masked Autoencoders (MAE)	Focused only on grape varieties; real-world images; fine-grained classification
[17]	Alrashdi et al. (2026)	Multi-fruit Classification (including grapes)	Unbalanced dataset	Public multi-fruit dataset (controlled + some real images)	~10,000+	10–30+ fruits (grape subset included)	DenseNet, ResNet, and Xception	Broader dataset; grape is only a subset; easier problem; includes multiple fruit types

The limitation of the dataset used in [17] consists of manipulated images; therefore, it does not reflect the realistic, real-world environment, which includes variability such as lighting changes, occlusions, and background noise. Another limitation is that distinguishing between different types of fruit is

inherently easier than distinguishing among alike classes in the same category, like grapes. These gaps emphasize the necessity of real-world grape images, a balanced dataset, a fine-grained grapevine dataset, and specialized methods capable of generalizing across varieties, seasons, and imaging conditions. This research addresses all these gaps.

3. Proposed Framework

The proposed framework consists of six phases: preprocessing, feature extraction, dimensional reduction, feature fusion, training classifiers, and evaluation with evaluation metrics. Figure 1 and Algorithm 1 illustrate the proposed framework and each phase of data processing up to the display of the evaluation of grape leaf image classification. The methodology consists of image preprocessing and deep feature extraction using CNN architectures (VGG16, VGG19, ResNet50, Inception V3) and then classified by machine learning methods (SVM (RBF), Logistic Regression, Random Forest, KNN, Naive Bayes, and Decision Tree). In the final step, the test phase, the model’s performance is evaluated using appropriate metrics and plotting.

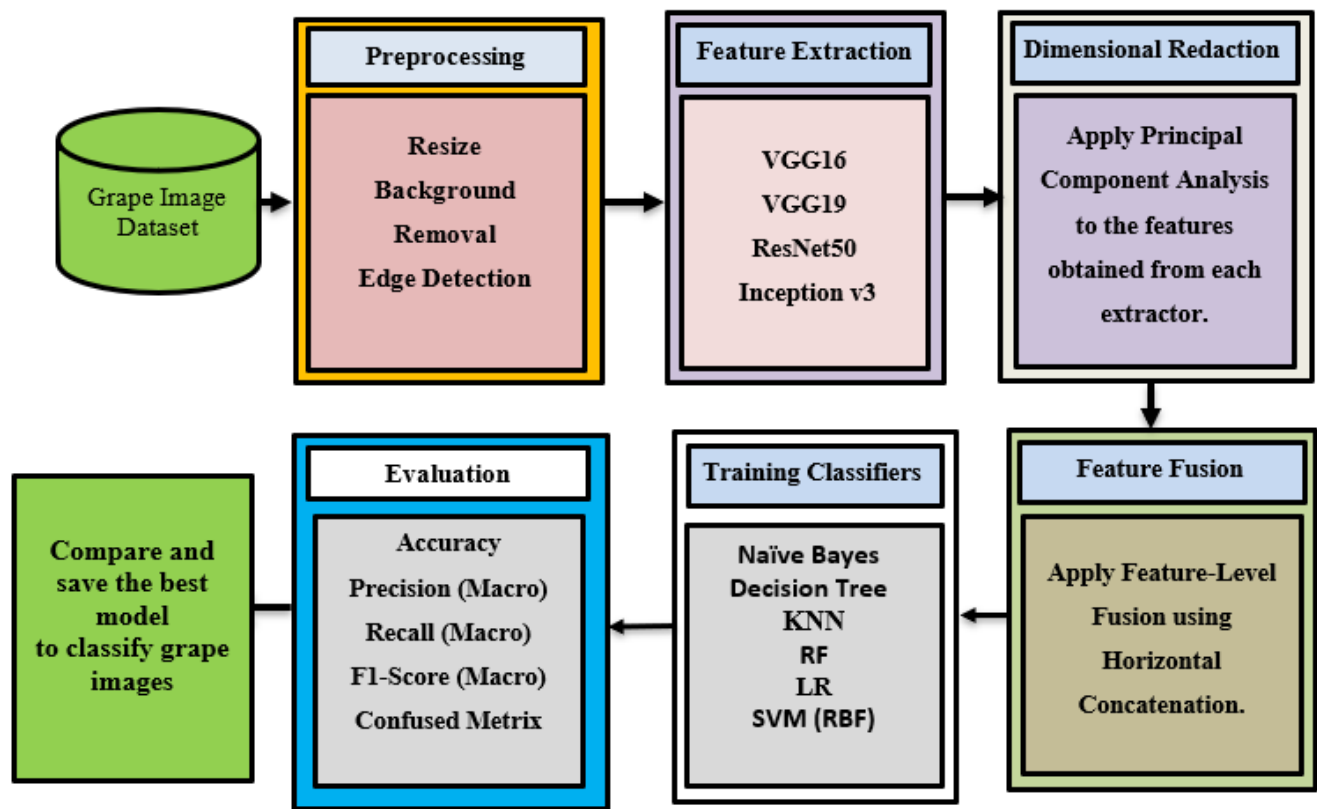


Figure (1): The flow of the proposed framework.

Algorithm 1. Pseudocode for CNN-Based Feature Extraction.**Algorithm 1:** Pseudocode for CNN-Based Feature Extraction**Input:** Grape leaf image dataset**Output:** Feature vectors and classification results

1. Load the grape leaf image dataset
2. For each image in the dataset:
 - a. Resize image to $299 \times 299 \times 3$
 - b. Normalize pixel values (optional)
3. Load pre-trained CNN model (VGG16 / VGG19 / Inception V3/ ResNet50)
4. Remove fully connected layers from the model
5. Pass pre-processed images through CNN
6. Extract features from the final convolutional layer
7. Flatten extracted feature maps into feature vectors
8. Store feature vectors with corresponding labels
9. Apply PCA with fusion to dimensional reduction
10. Train classifier using fused PCA features.
11. Evaluate model performance using accuracy, precision, recall, and F1-score

3.1 Dataset Overview

The dataset [18] used in this research is suitable for grape variety classification, as it represents real-world agricultural conditions and includes economically relevant classes, enabling the development of practical machine learning models for early plant identification. Also, it is fully balanced, with 1000 images per class, ensuring that no class dominates the learning process and reducing bias during model training. Images were captured in multiple agricultural regions in Iraq during the fruiting season, ensuring high visual discriminability between grape varieties and reflecting realistic environmental variability (lighting, background, leaf orientation). The eight grape varieties are Deas Al-Annz, Kamali, Halawani, Thompson Seedless, Aswud Balad, Riasi, Frinsi, and Shdah. The grape leaves were photographed, as shown in Figure. 2. For model development, the dataset was split using a stratified approach into training and testing sets to ensure fair and reliable evaluation.

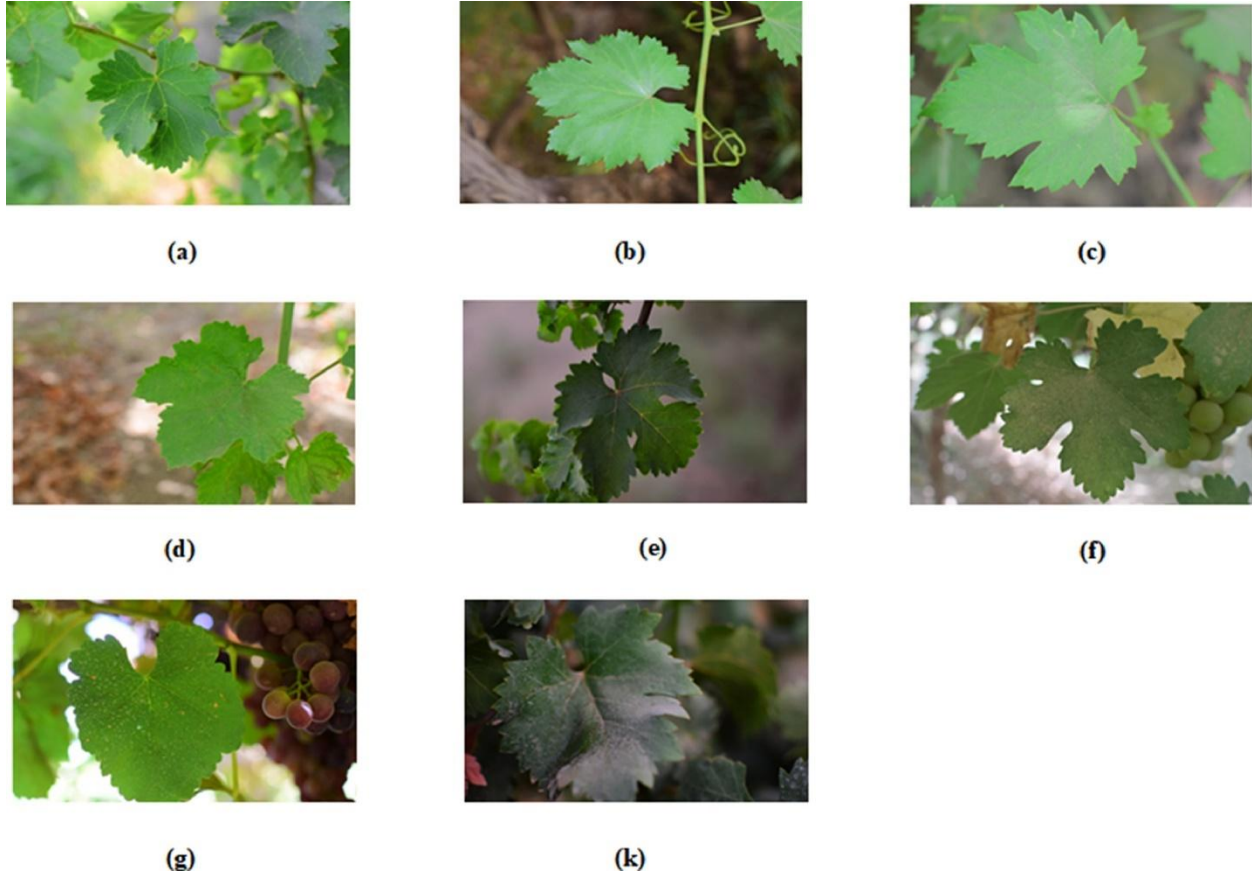


Figure 2. Examples of each of the included grape classes as follows: (a) Deas al-annz's leaf class, (b) Kamali's leaf class, (c) Halawani's leaf class, (d) Thompson seedless's leaf class, (e) Aswud Balad's leaf class, (f) Riasi's leaf class, (g) Frinsi's leaf class and (k) Shdah's leaf class.

3.2 Image Preprocessing

Let the raw image be represented as $I \in \mathbb{R}^{H \times W \times 3}$, where H and W are the height and width of the image. Preprocessing includes resizing to the input dimensions of the CNN (H', W'): Where the resizing to (299, 299) RGB.

• Background Removal

Background removal isolates the leaf from irrelevant pixels. Let $M \in \{0,1\}^{H \times W}$ be a binary mask obtained from image segmentation:

$$M(x, y) = \begin{cases} 1, & \text{if pixel } (x, y) \text{ belongs to leaf} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The leaf-only image I_{leaf} is computed as:

$$I_{\text{leaf}} = I_{\text{norm}} \odot M \quad (2)$$

where $\odot$ denotes element-wise multiplication. This ensures that the CNN focuses solely on leaf features.

• Edge Detection

The patterns of structural leaf can be highlighted by edge detection, such as margins and veins. With a generic edge detection operator $\mathcal{E}$ (e.g., Sobel or Canny), the edge map E is given by:

$$E = \mathcal{E}(I_{\text{leaf}}) \quad (3)$$

For example, in Sobel edge detection:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I_{\text{leaf}}, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I_{\text{leaf}} \quad (4)$$

$$E = \sqrt{G_x^2 + G_y^2}$$

where $*$ denotes convolution. The edge map E emphasizes leaf contours and vein structures.

3.3 Feature Extraction

Pre-trained CNN models such as VGG16, VGG19, ResNet50, and Inception V3 are widely used as feature extractors through transfer learning. Features extracted from these models can be fed into classifiers such as SVM and k-NN. Feature extraction was performed using multiple pre-trained convolutional neural network (CNN) architectures, including VGG16, VGG19, ResNet50, and InceptionV3, all of which were originally trained on the same dataset.

For each model, the top classification layers were removed to enable feature extraction. A global average pooling layer (pooling='avg') was applied to convert the convolutional feature maps into fixed-length feature vectors. This resulted in feature representations of 512 dimensions for VGG16, VGG19, and 2048 dimensions for ResNet50 and InceptionV3. The networks were used as fixed feature extractors without fine-tuning, meaning the pre-trained weights were not updated during the training. This approach reduces computational complexity and mitigates overfitting, particularly for moderate-sized datasets. All input images were resized to match the required input size of each architecture (224×224 for VGG16, VGG19, and ResNet50, and 299×299 for InceptionV3) and preprocessed using their respective normalization functions. The vectors of extracted feature are flattened and stored for subsequent classification. Using multiple CNN architectures allows capturing complementary feature representations, as VGG models focus on sequential feature hierarchies, while ResNet and Inception architectures capture deeper and multi-scale representations.

The choice of separating feature extraction and classification was motivated by several factors. First, the limited size of the dataset makes training an end-to-end deep learning model prone to overfitting. Second, pretrained CNNs are highly effective in extracting robust and discriminative features. Finally, traditional classifiers such as SVM can provide better generalization and require less computational cost. Therefore, this hybrid approach offers a practical and efficient alternative to fully end-to-end deep learning models. The novelty lies in combining deep features extracted from multiple heterogeneous CNN architectures rather than relying on a single model. Furthermore, unlike standard pipelines, PCA is applied individually to each feature set before fusion to reduce redundancy and improve discriminative power.

Let the processed image I_{proc} (after removing background and edge detection) be input to a CNN model f_{θ} parameterized by θ . The CNN computes hierarchical feature representations F :

$$F = f_{\theta}(I_{\text{proc}}) \quad (5)$$

For pre-trained models such as VGG16, VGG19, ResNet50, and InceptionV3, the feature vector $F \in \mathbb{R}^d$ is extracted from the output of a convolutional or fully connected layer before classification.

- VGG16/VGG19: VGGNet uses small convolution filters and deep architectures, making it effective for extracting detailed features. Feature maps are obtained after convolutional and pooling layers:

$$F_{\text{VGG}} = \text{Flatten}(\text{ConvPool}(I_{\text{proc}})) \quad (6)$$

- ResNet50: Residual Networks introduce skip connections that address the vanishing gradient problem and allow very deep networks to be trained. Uses residual blocks to learn deeper features:

$$F_{\text{ResNet}} = \text{ResBlock}_n(I_{\text{proc}}) + F_{\text{ResBlock}_{n-1}} \quad (7)$$

- InceptionV3: Inception models use parallel convolution operations with different kernel sizes to capture multi-scale features. Extracts multi-scale features with different convolution filter sizes:

$$F_{\text{Inception}} = \parallel_k \text{Conv}_k(I_{\text{proc}}) \quad (8)$$

where $\parallel$ denotes concatenation of feature maps from multiple filter sizes k .

3.4 Dimensional Reduction

Principal Component Analysis (PCA) was applied after feature extraction to reduce the dimensionality of deep features while retaining most of the variance. The number of principal components for each deep model was empirically determined based on preliminary experiments. Different values were tested, and the chosen numbers struck a beneficial balance between reducing dimensionality and improving

classification performance. Where (PCA) was practical after feature standardization using StandardScaler (with `with_mean=True`, `with_std=True`). PCA was implemented using *scikit-learn* with parameters `n_components = 150` for VGG16/VGG19 and `n_components = 300` for ResNet50/InceptionV3, `svd_solver = 'auto'`, and `whiten = False`. This setup reduces dimensionality while preserving the most significant variance, as verified by the cumulative explained variance ratio. These values were also verified to retain a high proportion of the total variance in the data, ensuring minimal information loss. Also, the truncation in our approach was applied to the feature dimension rather than the number of samples. Specifically, all samples were preserved, and only the feature vectors were adjusted to ensure consistent dimensionality across different models. Because of this approach, the data examples were not discarded, and the loss of examples did not occur.

3.5 Feature Fusion

Feature fusion used to enhance the discriminative ability of the extracted features by integrating complementary information obtained from multiple deep CNNs (VGG16, VGG19, ResNet50, and InceptionV3). Each of these architectures is known to capture distinct visual features due to differences in network depth, architectural design, and receptive field strategies. By assembling these heterogeneous features, the proposed fusion strategy aims to construct a more comprehensive and informative feature space, thereby improving the model's ability to distinguish between classes.

In this research, feature-level fusion is performed using horizontal concatenation (feature stacking). This approach preserves the full set of features extracted from each model without information loss, unlike averaging-based methods, which may suppress important discriminative signals. The concatenated feature vector enables the subsequent classifier to learn more robust decision boundaries by exploiting the diversity and complementarity of the individual feature sets.

To ensure proper alignment across feature matrices, all extracted feature sets are adjusted to have a consistent number of samples before fusion. Specifically, each feature matrix is truncated to match the minimum sample size among the models, guaranteeing compatibility during concatenation.

Since each type of CNN extracts features at different depths and domains, combining these features leads to strong class differentiation. To obtain dimensions suitable for machine learning algorithms and to eliminate feature redundancy, principal component analysis (PCA) was applied to the features extracted from each model. After applying PCA, all feature matrices were aligned by truncating them to the minimum number of samples across models. The aligned and truncated feature vectors were then horizontally linked using vector merging, resulting in a single feature matrix. This process links the features extracted from all models at the feature level, as expressed by:

$$\mathbf{X}_{fused} = [\mathbf{X}_{VGG16} \mid \mathbf{X}_{VGG19} \mid \mathbf{X}_{ResNet50} \mid \mathbf{X}_{InceptionV3}] \quad (11)$$

where $\mathbf{X}_{fused}$ represents the ending fused feature matrix, and $(\mathbf{X}_*)$ denotes the PCA-reduced features from CNN models. The fused feature vector was obtained, providing a clear representation of the input images and conveying distinct semantic and spatial information. This leads to satisfactory results in classification and overall performance compared to using features from a single model. The extracted deep feature vectors F , after applying PCA and fused PCA features, can then be used for the classification of grape leaf characteristics.

3.6 Training Classifiers

The dataset is split into training and testing sets using an 80/20 stratified split with a random seed of 42 to ensure reproducibility. We trained six machine learning classifiers: SVM with RBF kernel ($C=10$, $\gamma=\text{'scale'}$), Random Forest (200 trees, $\text{random_state}=42$), Logistic Regression ($\text{max_iter}=1000$), K-Nearest Neighbors ($k=5$), Gaussian Naive Bayes, and Decision Tree ($\text{random_state}=42$). Features were aligned with labels and preprocessed to ensure one-to-one correspondence. We evaluated models using accuracy, precision, recall, and F1-score (macro-averaged). All experiments were conducted using Python 3.10, scikit-learn 1.2.2, pandas 1.5.3, NumPy 1.25.0, and matplotlib 3.7.1.

3.7 Evolution Metrics

Assessing how well a model works involves using appropriate evaluation criteria supporting the particular problem. The dataset's properties and the type of analysis being done determine these criteria. The fundamental evaluation metrics are presented in Table 2 for models evaluated and investigated in this research. The above measures are based on four principal ideas:

- **True Positive (TP):** When the model correctly predicts the class of a grape sample (i.e., the predicted class matches the actual class among the eight grape classes).
- **True Negative (TN):** When the model correctly identifies that a grape sample does not belong to a specific class.
- **False Positive (FP):** When the model incorrectly predicts that a grape sample belongs to a certain class while it actually belongs to another class.
- **False Negative (FN):** When the model fails to identify the correct class of a grape sample and assigns it to a different class.

Table (2): Summarizes the Performance Metrics.

Metric	Equation
Accuracy	$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$
Precision	$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$
Recall (Sensitivity)	$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$
F1 Score	$\text{F1} = 2 * (\text{precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$
ROC-AUC	N/A (Area under the ROC curve)

4. Results and Discussion

This section presents preprocessing results (as shown in Figure 3 and Figure 4) and performance metrics—including precision, accuracy, recall, F-score, ROC-AUC, and others—for several machine learning models evaluated on Fused_PCA_Features_with_labels. After that, we compare the models with the empirical research that has utilized deferent machine learning models for the prediction of grape

classes. Using the VGG19 convolutional neural network to extract deep features from grape leaf images produced a 512-dimensional vector for each image. The sample size was 1000 for each of the eight grapes varieties. The same features were extracted from grape leaf images using ResNet50 (2048 features), Inception V3 (2048 features), and VGG16 (512 features). Statistical analysis has shown that convolutional neural networks produce deep features that accurately represent grape leaves and can be used to classify these leaves in support of agricultural artificial intelligence systems.

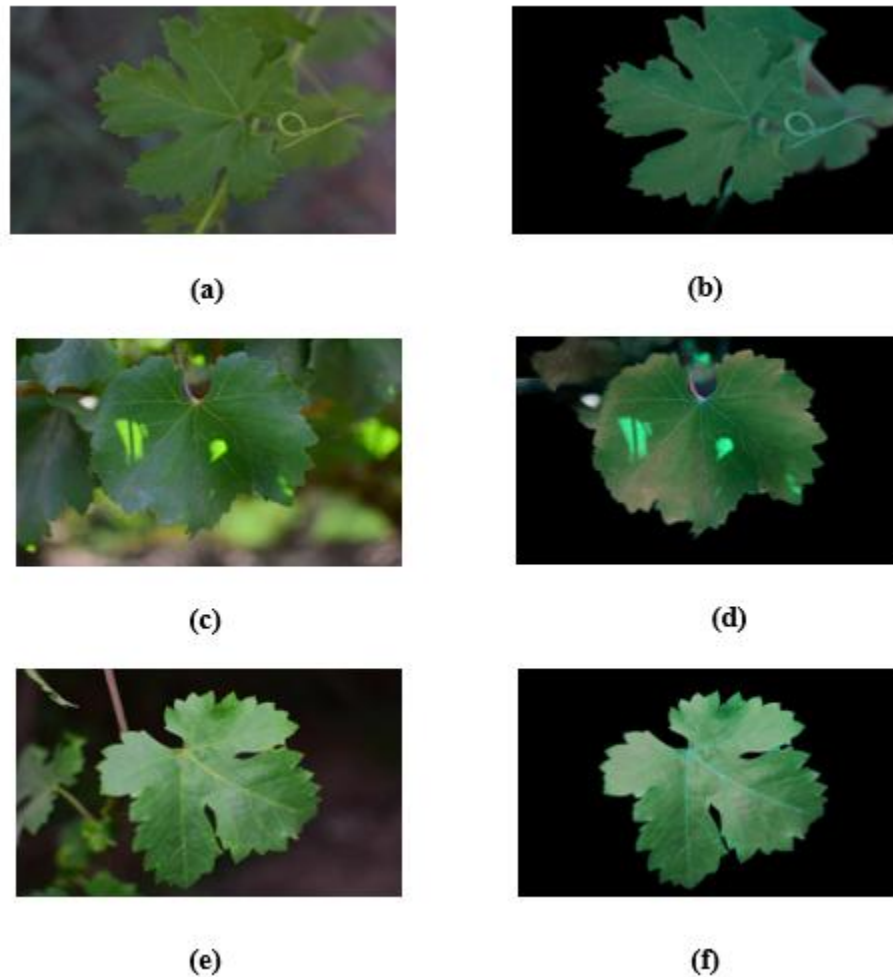


Figure 3. Examples of each of the remove background as follows: (a) Thompson seedless with background, (b) Thompson seedless without background, (c) Shdah with background, (d) Shdah without background, (e) Aswud Balad with background (f) aswud balad without background.

As illustrated in Figure 3, segmentation effectively removes complex backgrounds while preserving the structural integrity of the leaf. Background removal was applied to isolate the leaf region from irrelevant environmental elements such as soil, branches, and varying illumination conditions. In raw images, these background components introduce noise and may lead the model to learn non-discriminative features unrelated to grape variety classification. By performing segmentation, the model is encouraged to focus

on intrinsic leaf characteristics such as shape, venation, and texture, which are critical for accurate classification. Also, to reduce feature variation caused by external factors and improve generalizability and the model's robustness, the preprocessing is implemented, which is very useful when working with real-world agricultural images.

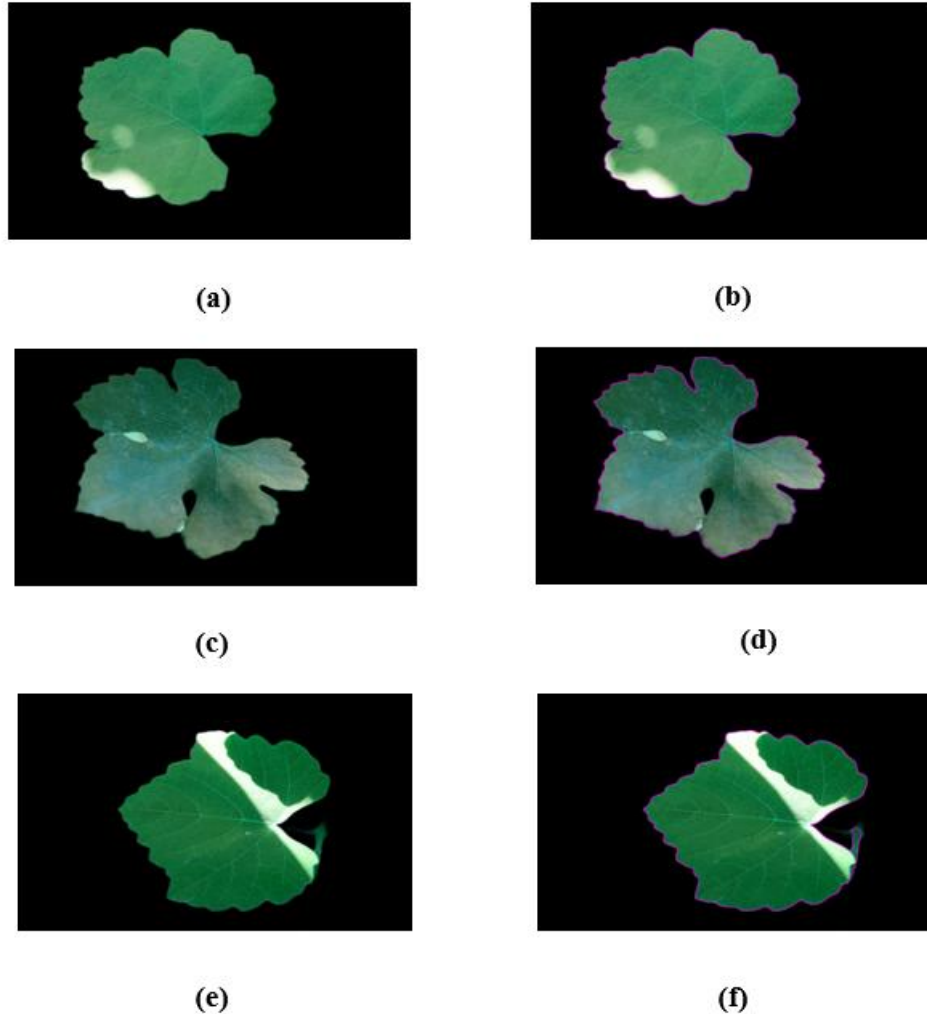


Figure 4. Examples of edge detection as follows: (a, c and e) Shdah without background and (b, d and f) Shdah with edge detection.

4.1 Dimensional reduction and feature fusion

Table 3 shows the features extracted using the convolutional neural network models employed in this study. Four pre-trained convolutional neural network models (VGG16, VGG19, ResNet50, and Inception V3) were used. The PCA was also used to decrease the dimensionality of the features. PCA was applied to each feature set extracted from the deep model for the eight grape varieties in the dataset described above. The feature vectors were then summed to produce the dataset used for classifying the grape images.

Table (3): Deep Feature Extracttion by Pretrained CNN Models.

Deep CNN Model	Feature Extraction Method	Dimensionality Reduction	Number of Features	Description
VGG16	Deep convolutional feature extraction	PCA	150	Captures hierarchical visual patterns from grape images
VGG19	Deep convolutional feature extraction	PCA	150	Provides deeper feature representation for image characterization
ResNet	Residual deep feature extraction	PCA	300	Enhances discriminative features using residual learning
Inception	Multi-scale convolutional feature extraction	PCA	300	Captures complex visual patterns at multiple scales
Label	Image annotation	—	1	Indicates the class/category of each grape image

4.2 Training Classifiers

After constructing the fused dataset, which includes feature vectors and corresponding class labels representing eight grape varieties, the data were split into 80% for training and 20% for testing using stratified sampling to preserve class distribution. Subsequently, several machine learning algorithms (Support Vector Machine (SVM) with an RBF kernel, Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), Simple Bayes, and Decision Tree) were trained, and the performance of each model was measured using the following metrics: accuracy, recall, and F1 score. Confusion matrices were also used for detailed error analysis. Figure 6 shows the macro-averaged precision–recall (PR) curves for all classifiers. It is noteworthy that SVM and logistic regression performed best, maintaining high accuracy across varying call levels. Random forest, on the other hand, performed moderately, with a noticeable decrease in accuracy at higher call levels, indicating an increase in false positives. While KNN and Naïve Bayes performed poorly, they were still included in the analysis.

Consistent with these observations, the average precision (AP) values further highlight the effectiveness of SVM (0.77) and logistic regression (0.76), followed by random forest (0.58). KNN (0.35), Naïve Bayes (0.30), and decision tree (0.17) achieved comparatively lower scores.

Figure 7 illustrates the macro-averaged receiver operating characteristic (ROC) curves. Logistic regression achieved the highest area under the curve (AUC) value of 0.94, closely followed by SVM with 0.93, indicating strong discriminative capability between classes. Random forest also performed well with an AUC of 0.87, whereas KNN (0.75), Naïve Bayes (0.67), and decision tree (0.59) showed weaker classification performance.

Overall, SVM and logistic regression consistently outperform the other models across both PR and ROC analyses, demonstrating their suitability for this multi-class classification task. Random forest provides acceptable performance but shows limitations at higher recall levels. The relatively lower performance of KNN, Naïve Bayes, and decision tree can be attributed to sensitivity to feature scaling, strong independence assumptions, and overfitting or poor generalization, respectively.

These results emphasize the importance of selecting classification models that balance precision, recall, and generalization capability, particularly in applications requiring reliable discrimination between multiple classes.

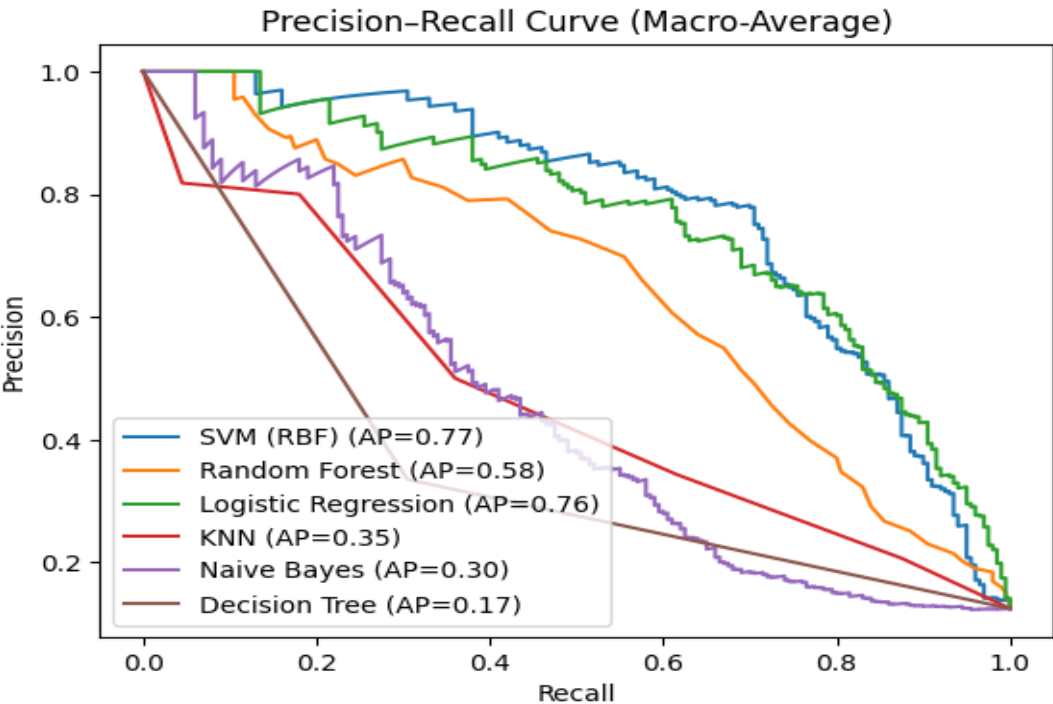


Figure 6 shows the macro-averaged Precision–Recall (PR) curves.

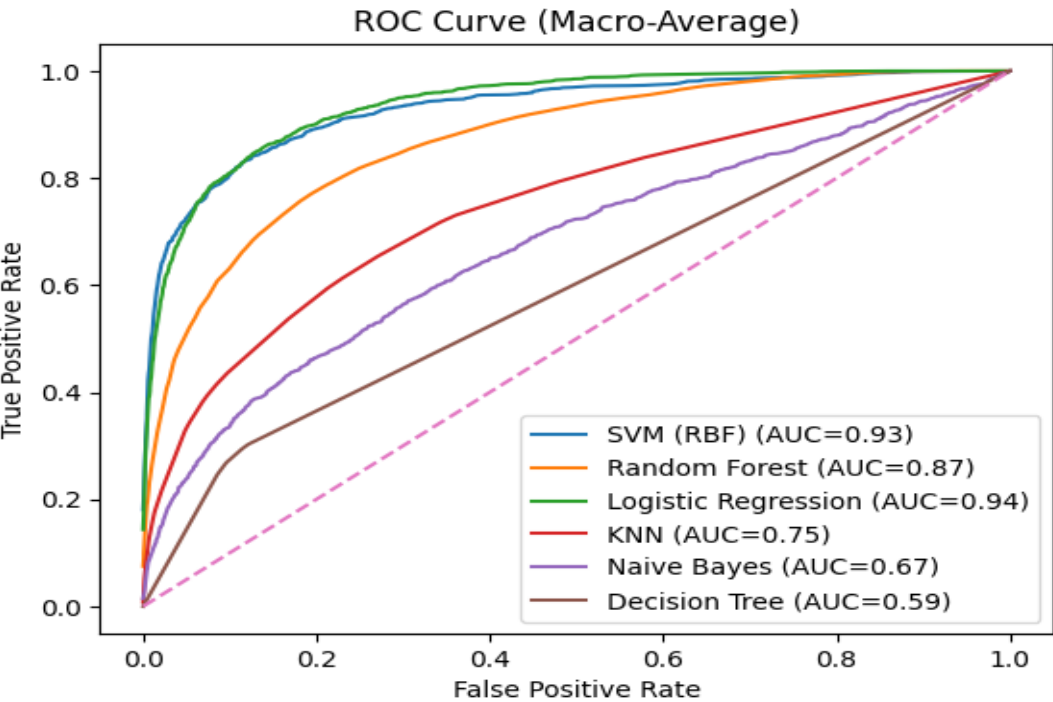


Figure 7 shows the macro-averaged (ROC) curves.

4.3 Evolution

In this section, we will discuss the test results, where accuracy, average accuracy, and F1 score calculated using macro analysis were used, along with recall. The SVM model with the RBF core outperformed the other models, achieving excellent results, as shown in Table 4, recording an accuracy of 0.716, along with similar values for overall accuracy (0.717), recall (0.716), and F1 score (0.716). This makes it the best among the other models. In second place, the logistic regression model reached an accuracy of 0.694, with similar results for accuracy, recall, and F1 score, indicating its efficiency in processing classification data. Third place came in with an average performance compared to the previous models, achieving an accuracy of 0.553 and an F1 score of 0.551. In contrast, the k-nearest neighbors (KNN) model performed worse than the earlier one, achieving an accuracy of only 0.386 and a low F1 score. The Naive Bayes and decision tree models had the lowest scores among all the models. The Naive Bayes model accomplished an accuracy of 0.310 and an F1 score of 0.298, while the Decision Tree model came in last place, accomplishing the lowest scores with an accuracy of 0.287 and an F1 score of 0.288.

Table 4 Reviews the Performance of the Evaluated Classifiers.

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
SVM (RBF)	0.716427	0.717189	0.716381	0.716117
Logistic Regression	0.693941	0.694707	0.693918	0.694075
Random Forest	0.553404	0.551842	0.553346	0.55126
KNN	0.386009	0.430997	0.386017	0.382352
Naive Bayes	0.309806	0.386109	0.309888	0.298305
Decision Tree	0.28732	0.289283	0.287301	0.287593

It is clear from examining the results that SVM (RBF) and logistic regression outperform the other models in classifying all categories, and the reason for SVM's superiority is its ability to distinguish nonlinear relationships between categories. This leads us to conclude that the SVM model is the most suitable among all models for classifying grape leaves. It is clear from Table 5 that the confusion matrix resulted from using the SVM classifier. The eight grape varieties are Dias Al-Anz, Kamali, Aswad Balad, Halwani, Frensi, Riasi, Shadah, and Thompson Seedless, where the rows denote the actual classes and the columns denote the predicted classes.

Table 5 The confusion matrix using the SVM classifier.

	Kamali	Aswud Balad	Deas Al-Annz	Frinsi	Halawani	Riasi	Shdah	Thompson Seedless
Kamali	151	7	2	8	6	3	11	12
Aswud Balad	6	139	10	5	4	17	13	6
Deas Al-Annz	3	8	145	15	9	11	6	3
Frinsi	17	6	12	115	6	10	13	21
Halawani	11	3	16	4	150	4	6	6
Riasi	4	10	5	8	4	159	8	3
Shdah	2	8	6	15	5	5	152	7
Thompson Seedless	15	2	7	24	2	3	11	136

We observe from the confusion matrix of the SVM classifier that it achieves acceptable classification accuracy for all eight grape varieties. The SVM achieved very high classification accuracy, yielding the following results: Riasi (159), Shaddah (152), Halwani (150), Dias Al-Anz (145), and Kamali (151), demonstrating the model's ability to learn clear decision boundaries for these classes. Aswud Balad class also achieved fine, reaching a classification accuracy of 139 correct samples, while there was some overlap with Shaddah (13) and Riasi (17). The errors in the matrix above show that these classes share similar morphological or visual features, which affects classification performance. The errors in the matrix above indicate that these classes share same morphological or visual characteristics, which affects the classification accuracy.

Table 5 displays the confusion matrix of the SVM for the eight grape classes. To give more insights, the per-class error rates have been computed as in Table 6.

$$\text{Error Rate} = 1 - \frac{\text{Correct Predictions}}{\text{Total Samples per Class}} \quad (12)$$

Table 6 The per-class error rates.

Class	Correct	Total	Error Rate
Kamali	151	197	23 / 197 $\approx$ 11.7%
Aswud Balad	139	200	61 / 200 $\approx$ 30.5%
Dias Al-Anz	145	200	55 / 200 $\approx$ 27.5%
Frinsi	115	200	85 / 200 $\approx$ 42.5%

Halwani	150	200	$50 / 200 \approx 25.0\%$
Riasi	159	200	$41 / 200 \approx 20.5\%$
Shadah	152	200	$48 / 200 \approx 24.0\%$
Thompson Seedless	136	200	$64 / 200 \approx 32.0\%$

Table 6 shows that some classes, like *Frensi* and *Aswud Balad*, are hard to tell apart, while *Kamali* and *Riasi* are easy to distinguish. Overall, the SVM demonstrates excellent performance across the eight grape varieties, providing the highest accuracy for classes such as *Riasi*, *Shadah*, *Halwani*, *Kamali*, and *Dias Al-Anz*, which exhibit some degree of differentiation in their features. Visually similar classes, such as the confusion between *Frensi* and *Thompson Seedless*, and between *Aswud Balad*, *Riasi* and *Shadah*, where misclassification cases are concentrated, indicate overlapping feature distributions and limited ability to distinguish subtle differences. Asymmetric error patterns suggest intra category variability and less clearly defined category boundaries. Overall, the model excels in clearly separable categories but emphasizes the need to extract more precise features to better differentiate highly similar grape varieties. Because no previous study on the same dataset exists, a logical comparison with existing work is impossible. Therefore, a qualitative comparison with existing work highlights the effectiveness of the proposed approach [16]. High classification accuracy (95–98%) has been reported using a self-learning framework. However, their experiments were conducted on controlled datasets and utilized advanced pretraining techniques.

Similarly, [17] addressed multi-fruit classification, where distinguishing between different fruit categories is inherently less complex than fine-grained classification within the same category, such as grape varieties. While the data used to evaluate the proposed method is a real-world dataset containing an equal number of images for each category, varying in backgrounds, image quality and lighting. With these challenging conditions, the proposed approach gives 71.6% accuracy by using SVM (RBF), which means high generalization capability. This accuracy is less than that reported in controlled settings, but the results are more representative of real-world agricultural scenarios, highlighting the strength and practical applicability of the proposed method. As a result, the achieved accuracy of 71% reflects the inherent difficulty of fine-grained grape variety classification rather than a limitation of the proposed approach. Furthermore, unlike previous works that rely on complex architectures or ensemble models, this study employs a simpler model, which may also contribute to the lower accuracy. However, the result highlights the need for more robust and generalizable models for real-world grape variety classification.

5. Conclusion

This research explored grape variety classification as a fine-grained image recognition problem under real-world conditions, where differences between classes are subtle and often difficult to capture. The SVM achieved the highest result, reaching an accuracy of 71.6%. While some previous studies achieved higher accuracy because they implemented easier classification tasks or used controlled datasets. These weaknesses actually make this paper preferable, as the data set used in this research is more realistic,

where this research implements harder classification problem because it uses real-world images with complex backgrounds and high similarity between classes.

The results explained that using CNN-based feature extraction with machine learning classifiers is advantageous, but not enough on its own to completely solve the challenges of fine-grained classification. Due to overlapping visual characteristics, some grape varieties remain difficult to distinguish even with feature fusion. These findings explained that the performance of the suggested framework reflects the difficulty of the dataset rather than a weakness in the approach.

Future enhancements could focus on implementing advanced models like transformer-based models or fine-tuning deep approaches using attention mechanisms. In general, this research give a realistic baseline and highlights the requirements for more specialized techniques in real-world agricultural image classification.

6. Limitations and Future Work

This research faced several limitations, including the fact that it worked on a single dataset despite containing 8000 samples for 8 classes. While the machine learning models performed well, their parameters still need improvement. Furthermore, some ambiguity persisted between similar grape varieties, potentially affecting the accuracy of differentiation. Therefore, the following research approaches can be suggested:

- Utilizing multiple datasets from diverse sources to enhance model robustness and generalizability.
- Expanding the proposed framework to include heterogeneous plant datasets.
- Performing more precise adjustments to the machine learning model parameters to improve performance.
- Employing interpretable artificial intelligence (XAI) techniques to analyze model decisions and deepen our understanding of their operation.

Declaration of generative AI and AI-assisted technologies in the writing process

-None

Disclosures

The author has no conflicts of interest to declare in relation to this manuscript.

References

[1] M. P. Mathew, "A comparative deep learning framework for grape leaf disease classification using EfficientNetB0, InceptionV3, and Xception," Discover Applied Sciences, vol. 7, p. 1222, 2025, doi: 10.1007/s42452-025-07457-5.

- [2] K. V. Prasad, H. Vaidya, C. Rajashekhar, K. S. Karekal, R. Sali, and K. S. Nisar, "Multiclass classification of diseased grape leaf identification using deep convolutional neural network (DCNN) classifier," *Scientific Reports*, vol. 14, p. 9002, 2024, doi: 10.1038/s41598-024-59562-x.
- [3] J. Huang, "Research on grape leaf classification based on optimized Densenet201 model," *PLOS ONE*, vol. 20, no. 10, p. e0334877, Oct. 2025, doi: 10.1371/journal.pone.0334877.
- [4] L. Deng, Z. Du, X. Liu, Z. Wu, X. Lin, and B. Wen, "Grape leaf cultivar identification in complex backgrounds with an improved MobileNetV3-Small model," *Plants*, vol. 14, no. 23, p. 3581, Nov. 2025, doi: 10.3390/plants14233581.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.
- [6] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
- [7] R. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
- [8] M. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019.
- [9] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 3289801, 2016.
- [10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2015.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
- [13] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, vol. 2016, Art. no. 3289801, 2016.

- [14] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2014, pp. 3320–3328.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
- [16] G.A. Carneiro, T. J. Aubry, A. Cunha, P. Radeva, and J. Sousa, "Advancing image-based grapevine variety classification with a new benchmark and evaluation of masked autoencoders," *arXiv preprint*, arXiv:2506.13335, 2025.
- [17] I. Alrashdi et al., "Utilizing deep learning models for fruit disease detection and classification," *Scientific Reports*, 2026.
- [18] L. R. Al-Khazraji, M. A. Mohammed, D. H. Abd, W. Khan, B. Khan, and A. J. Hussain, "Image dataset of important grape varieties in the commercial and consumer market," *Data in Brief*, vol. 46, p. 108906, 2023, doi: 10.1016/j.dib.2023.108906.

تصنيف صور العنب بالاعتماد على الشبكات العصبية الالتفافية العميقة وتحسين تمثيل السمات

الخلاصة: يُعد تصنيف أصناف العنب اعتمادًا على الصور الرقمية لأوراقها من المهام الصعبة في مجال تصنيف الصور، نظرًا للتشابه الكبير بين الأصناف المختلفة في الخصائص الشكلية والملبس واللون. في هذه الدراسة، نقترح إطارًا هجينًا يجمع بين تقنيات التعلم العميق والتعلم الآلي بهدف تحقيق أداء تصنيفي فعال في البيانات الزراعية الواقعية. وقد صيغت المشكلة باعتبارها مسألة تصنيف متعدد الفئات تشمل ثمانية أصناف من العنب، بالاعتماد على مجموعة بيانات متوازنة تتكون من 8,000 صورة لأوراق العنب جُمعت من بيانات زراعية واقعية.

ولتحسين جودة المعلومات البصرية وتعزيز قدرة النماذج على استخلاص السمات المميزة، خضعت الصور لعدد من مراحل المعالجة المسبقة، شملت إزالة الخلفية لعزل الورقة عن العناصر المحيطة غير ذات الصلة، بالإضافة إلى تطبيق تقنيات كشف الحواف لإبراز الخصائص البنوية المهمة للورقة، ولا سيما المحيط والعروق. بعد ذلك، استُخرجت السمات العميقة باستخدام مجموعة من نماذج الشبكات العصبية الالتفافية (CNN) المدربة مسبقًا، وهي InceptionV3 و ResNet50 و VGG16 و VGG19، بهدف الحصول على تمثيلات بصرية غنية ومتكاملة للصور. ولتقليل التكرار والتعقيد الحسابي في فضاء السمات، استُخدم تحليل المكونات الرئيسية (PCA) لتقليل أبعاد السمات، ثم دُمجت السمات المستخرجة من النماذج المختلفة في تمثيل موحد. ولتقييم فعالية الإطار المقترح، جرى اختبار مجموعة من خوارزميات التعلم الآلي، شملت خوارزمية أقرب الجيران (KNN)، وآلة المتجهات الداعمة (SVM)، والانحدار اللوجستي (Logistic Regression)، والغابة العشوائية (Random Forest)، وشجرة القرار (Decision Tree)، وخوارزمية نايف بايز (Naïve Bayes). أظهرت النتائج أن خوارزمية آلة المتجهات الداعمة (SVM) حققت أفضل أداء بين النماذج المُقيمة، مسجلة دقة تصنيف بلغت 71.6%.

مع أداء متوازن في مقاييس الدقة (Precision)، والاستدعاء (Recall)، ودرجة (F1 (F1-score). وعلى الرغم من أن مستوى الدقة المحقق كان أقل من المستويات المسجلة في الدراسات التي اعتمدت على مجموعات بيانات جُمعت في ظروف بيئية محكمة، فإن النتائج تعكس بصورة أكثر واقعية التحديات المرتبطة بتنوع الصور والظروف المعقدة للبيانات الزراعية الحقيقية. وبوجه عام، تشير النتائج إلى أن دمج استخراج السمات العميقة مع تقليل الأبعاد ودمج السمات يمكن أن يوفر إطارًا فعالًا لتصنيف أصناف العنب، وأن هذا النهج يمتلك إمكانات واعدة لتطبيقات التعرف الآلي على الأصناف النباتية في البيانات الزراعية الواقعية.

الكلمات المفتاحية: الشبكات العصبية الالتفافية؛ استخراج السمات؛ التعلم العميق؛ الرؤية الحاسوبية؛ تصنيف الصور؛ تصنيف أوراق العنب.