Int. J. Sci. R. Tech., 2026 3(5)
A Multidisciplinary peer-reviewed Journal
www.ijsrtjournal.com [ISSN: 2394-7063]

Original Article

Grapevine Leaves Classification Using Advanced Data Augmentation Techniques

D. Banu Kranthi*, Surya Lakshmi Ramabhotla

Department of MSc. Data Science, Loyola Academy, Secunderabad-500010, India

ABSTRACT

This project presents an enhanced grapevine leaf classification by extending the CNN-SVM framework from the study "A CNN-SVM Study Based on Selected Deep Features for Grapevine Leaves Classification". While the original research used basic augmentation and feature fusion from custom CNN top blocks (MobileNetV2 and VGG19), our approach integrates advanced augmentation techniques- Mixup, CutMix, RandAugment, and Augmix- to increase data diversity and reduce overfitting. Features are extracted from MobileNetV2's intermediate convolutional layers using Global Average Pooling, followed by Chi-Square selection of the top 250 features for classification using a Cubic SVM.

The original dataset of 500 images was augmented to over 3,000 samples. Our method achieves a classification accuracy of 98.17% surpassing the original study's 97%. Both the previous work and present work included detailed metrics like Precision, Recall, F1-score, MCC, and confusion matrix analysis along with Accuracy as the primary report. This work demonstrates the impact of advanced augmentation in improving performance for fine-grained plant classification tasks and offers a practical, scalable pipeline for agricultural AI.

Keywords: Mixup, CutMix, RandAugmentation, MobileNetV2, CNN, SVM, Chi-Square Feature Selection.

INTRODUCTION

Grapevine cultivation plays a pivotal role in the agriculture economies of many countries, and accurate identification of grapevine leaf species is essential for proper vineyard management, pest control, and cultivar selection. Traditional methods of species classification often rely on expert knowledge and manual inspection, which can be both-consuming and error-prone. In recent years, the fusion of computer vision, and deep learning has opened new avenues for automating this task with high precision.

The advent of Convolution Neural Networks (CNNs) has revolutionized image-based classification tasks, providing robust feature representations for complex datasets. However, combining CNN's with traditional machine learning models such as Support Vector Machines (SVM's) has shown promising improvements in classification performance. In particular, the study by Kök and Yıldırım (2021) demonstrated that integrating CNN-based feature extraction with SVM classification using selected deep features significantly enhances grapevine leaf classification accuracy.

Motivated by this approach, our study implements an advanced hybrid framework that builds upon the foundational work presented in their research. Specifically, we leverage a pre-trained MobileNetV2 model to extract deep features from grapevine leaf images, followed by feature selection using the Chi-square (χ^2) statistical test. These selected features are then classified using SVMs with various kernel functions, including a polynomial kernel with degree three-referred to as the cubic kernel-known for its higher-order decision boundaries.

To further enhance the robustness and generalization capability of our model, we incorporate a range of modern data augmentation techniques such as RandAugment, AugMix, Mixup and CutMix. These augmentation strategies not only increase the diversity of the training data but also improve model resilience to real-world variations in image quality, lighting, and background conditions.

Our experiment pipeline is designed to evaluate the effectiveness of this CNN-SVM hybrid model using comprehensive evaluation metrics, including

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

accuracy, precision, recall, F1-score, ROC-AUC, and the Matthews Correlation Coefficient. The results validate the synergy between deep learning feature extraction and traditional SVM-classification, offering a scalable and efficient solution for grapevine leaf species identification.

LITERATURE REVIEW

1. **"Mixup: Beyond Empirical Risk Minimization"**

Authors: Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, David Lopez-Paz
Year:2017
Summary: Proposes a novel data augmentation method called *Mixup*, which creates new training samples by linearly interpolating pairs of examples and their labels. This simple yet powerful approach improves the generalization of neural networks, reduces memorization of noisy labels, and increases robustness to adversarial attacks. It serves as a strong regularizer by encouraging smoother decision boundaries.

2. **"AutoAugment: Learning Augmentation Policies from Data"**

Authors: Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, Quoc V. Le
Year: 2018
Summary: Introduces AutoAugment, a reinforcement learning-based method that automatically searches for optimal image augmentation policies directly from the training data. It discovers powerful combinations of transformations like rotation, color adjustment, and cropping, significantly boosting model performance across CIFAR, SVHN, and ImageNet without manual tuning.

3. **"CutMix: Regularization Strategy to Add Image-Level Context"**

Authors: Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe,YoungjoonYoo
Year:2019
Summary: Proposes *CutMix*, an augmentation technique that combines two images by cutting a patch from one and pasting it onto another while also mixing their labels proportionally. This method enriches the image context and object diversity, encouraging the model to pay attention to a broader set of visual cues. CutMix outperforms Mixup on multiple benchmarks and improves both classification and localization.

4. **"AugMix: A Simple Data Processing Method to Improve Robustness andUncertainty"**

Authors: Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer,BalajiLakshminarayanan
Year:2019
Summary: Introduces a powerful data augmentation strategy aimed at enhancing the robustness and reliability of deep learning models. AugMix works by mixing multiple augmentation operations—such as contrast adjustments, blurring, and geometric transformations—into a single composite image using a stochastic process. It significantly improves performance under image corruptions and offers better uncertainty estimation.

5. **"ReMixMatch: Semi-Supervised Learning with Augmentation Anchoring"**

Authors: David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, Colin Raffel
Year:2020
Summary: Extends MixMatch by introducing *ReMixMatch*, a semi-supervised learning method that integrates *augmentation anchoring*—a strategy where strongly augmented data must still predict consistently. It combines distribution alignment, entropy minimization, and a rotation-based auxiliary loss to improve generalization on unlabeled data, achieving state-of-the-art performance on multiple SSL benchmarks.

6. **"MixReg: Enhancing RL Generalization with Mixture Regularization"**

Authors: Tianjun Zhang, Xingyou Song, Yixin Lin, Yang Gao, Jacob Andreas, ChelseaFinn,DawnSong
Year:2020
Summary: Adapts mix-based augmentation concepts to reinforcement learning by introducing *MixReg*. It regularizes training by mixing states and returns, enabling better generalization to unseen tasks. This work broadens the application of data augmentation

beyond supervised learning and is especially impactful in environments with limited data or high variance.

7. **"RandAugment: Practical Automated Data Augmentation"**

Authors: Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, Quoc V. Le
Year:2020
Summary: Simplifies AutoAugment by introducing *RandAugment*, which removes the search phase and uses a fixed set of augmentations controlled by just two hyperparameters: the number of transformations and their magnitude. Despite its simplicity, RandAugment achieves performance close to AutoAugment on several benchmarks, making it practical for fast and scalable model training.

8. **"A Comprehensive Survey of Image Augmentation Techniques for Deep Learning"**

Authors: Yawen Xu, Jiayi Ma, and Xueyao Zhang
Year:2022
Summary: Offers a comprehensive taxonomy of image augmentation techniques across domains such as classification, detection, and segmentation. It categorizes augmentations into geometric, photometric, and adversarial types, while also covering automated and mix-based methods like Mixup, CutMix, RandAugment, and AugMix. The paper serves as a one-stop reference for selecting suitable augmentation methods for various tasks.

9. **"Image Data Augmentation for Deep Learning: A Survey"**

Authors: Wei Yang, Huan Wang, and Gang Pan
Year:2022
Summary: This survey provides an in-depth comparative analysis of augmentation techniques across multiple computer vision tasks. It explores both traditional (flips, crops) and advanced (GAN-based, learned policies) methods, emphasizing their impact on model generalization, robustness, and fairness. The paper also highlights open challenges and future directions in augmentation research.

10. **"A CNN-SVM study based on selected deep features for grapevine leaves classification"**

Authors: Köklü, Muhammed Fahri Ünlersen, Ilker Ali Özkan, Muhammet Fatih Aslan, andKadirSabanci
Year:2022
Summary: This study focuses on classifying grapevine leaves from five species using deep learning and traditional machine learning techniques. The authors expand a 500-image grapevine leaf dataset to 2,500 via augmentation, extract deep features with a fine-tuned MobileNetV2, select the top 250 features using Chi-square, and apply a cubic-kernel SVM to achieve highest accuracy across five leaf species.

METHODOLOGY

While traditional image classification techniques can provide baseline results, their performance often diminishes when faced with real-world challenges such as duplicate images, noise, blurriness, lighting variations, and limited datasets. To overcome these limitations, this work employs a hybrid approach that synergizes data quality filtering, advanced data augmentation, deep feature extraction using a pretrained CNN backbone, and feature selection techniques, culminating in an interpretable and high-performance classification pipeline.

1. Data Quality Analysis

The process begins with rigorous data validation to ensure the integrity and quality of input images. This involves several filtering steps:

- **Corruption Detection:** Images are checked for corruption using PIL's Image.verify() method, which detects unreadable or incomplete files.
- **Blurriness Filtering:** Blurry images are identified via Laplacian variance thresholding, with images below a certain variance (e.g., 50) being discarded, ensuring only sharp and clear samples are retained.
- **Duplicate Removal:** Duplicates are detected using perceptual hashing (pHash) through the imagehash library, which computes perceptual hashes; images with identical hashes are considered duplicates and are removed.

2. Data Augmentation

- To mitigate dataset limitations and improve the model's ability to generalize across various environmental conditions, multiple augmentation techniques are employed:
  - ✓ **Standard Augmentation:** Rotation, shift, zoom, and horizontal flip are applied via Keras' ImageDataGenerator.
  - ✓ **RandAugment:** Randomized augmentation policies are implemented to introduce stochastic variability, applied through Keras layers.
  - ✓ **AugMix:** Using the Albumentations library, multiple augmentation chains are combined probabilistically, including blurring, brightness, contrast, and rotation, to produce diverse augmented images.
  - ✓ **MixUp:** Pairs of images and their labels are blended linearly based on a Beta distribution, generating new training samples with mixed features.
  - ✓ **CutMix:** Regions of one image are replaced with corresponding regions from another image, and labels are mixed proportionally, encouraging the model to focus on multiple parts of the image.
- The augmented dataset, after normalization (resizing images to 224×224 pixels and scaling pixel values to [0,1]), is combined with original data to form a diverse training set aimed at improving generalization.

3. Dataset Preparation and Splitting

The validated and augmented dataset is split into training and testing subsets, typically using an 80/20 split with a fixed random seed for reproducibility. Labels are encoded as integers based on folder structure or metadata.

Deep Learning: MobileNetV2 for Feature Extraction

- A pretrained MobileNetV2 model, trained on ImageNet, is employed as a feature extractor:
  - ✓ The convolutional base of MobileNetV2 is used with the top classification layers.
  - ✓ The convolutional layers are unfrozen to learn deep features.
  - ✓ A custom classifier head comprising global average pooling, a dense layer with ReLU activation, dropout for regularization, and an output layer with softmax activation is added.
  - ✓ The combined model is trained for 50 epochs using the Adam optimizer with sparse categorical crossentropy loss, fine-tuning the network to adapt to grapevine leaf features.
- Once trained, the convolutional backbone of MobileNetV2 is utilized to extractdeep features from both training and testing images. These features, obtained from the last convolutional layer, are flattened into feature vectors for subsequent classical machine learning processing.

4. Feature Selection: Chi-square Method

Given the high dimensionality of deep features, a Chi-square feature selection method (via SelectKBest with chi2 scoring) is applied to retain only the top 250 most relevant features. This reduces computational complexity and mitigates overfitting, while preserving discriminative information critical for classification.

5. Classical Machine Learning: SVM Classifiers

The selected features serve as inputs to Support Vector Machine classifiers with various kernels (linear, RBF, polynomial). After training and tuning, the polynomial kernel (degree 3) was identified as optimal, providing the best trade-off between accuracy and generalization.

6. Model Evaluation

- The classification performance is evaluated on the test set using comprehensive metrics:
  - ✓ Accuracy
  - ✓ Precision
  - ✓ Recall (Sensitivity)
  - ✓ F1 Score
  - ✓ Matthews Correlation Coefficient (MCC)

- ✓ Confusion matrix visualization
- ✓ ROC-AUC score

- These metrics are visualized using heatmaps and ROC curves to assess model robustness and identify potential areas for improvement.

7. Deployment via Django Web Application

- The trained models—including the deep learning feature extractor, feature selector, and SVM classifier—are serialized and integrated into a Django-based web application. The system supports:
  - ✓ **Image Upload:** Users can upload grapevine leaf images through a web interface.
  - ✓ **Preprocessing:** Uploaded images undergo the same validation, resizing, normalization, and filtering steps as during training.
  - ✓ **Feature Extraction & Classification:** The backend loads the trained MobileNetV2 convolutional backbone to extract features, applies the Chi-square selector, and classifies the image with the trained SVM model.
  - ✓ **Results Presentation:** The predicted class, confidence scores, and relevant visualizations are displayed to the user.
- This deployment pipeline ensures real-time, interpretable, and accurate classification, making the system practical for field applications in vineyard management

Advantages:

- The combination of diverse augmentation techniques enriches the training data, leading to improved generalization.
- Data quality filtering ensures the training process is based on reliable samples, enhancing model confidence.
- Dimensionality reduction via Chi-square feature selection enhances classifier performance and reduces computational load.
- The hybrid CNN-SVM pipeline offers both high accuracy and interpretability, suitable for practical deployment.

8. MODULES

The following modules were created for this grapevine leaf classification project:

- **Data Quality Analysis** – Validates input images by removing corrupt, blurry, and duplicatesamples.
- **Data Augmentation** – Applies multiple transformations like RandAugment, MixUp, CutMix, and AugMix to increase dataset variety.
- **Dataset Preparation** – Loads the validated and augmented dataset, normalizes the images, and encodes the labels
- **Train-Test Split** – Separates the final dataset into training and testing sets for evaluation.
- **MobileNetV2 Training** – Uses a pretrained MobileNetV2 model for deep feature learning on grapevine leaf images.
- **Feature Extraction** – Extracts image features from MobileNetV2's last convolutional layer.
- **SVM Classification** – Applies SVM classifiers with various kernels to the extractedfeatures.
- **Chi-square Feature Selection** – Selects the most relevant features for optimized classification.
- **Model Evaluation** – Assesses classification results using accuracy, F1-score, precision, MCC, confusion matrix, and ROC AUC.
- **Model Saving & Export** – Exports trained models and feature maps for further integration or deployment.

IMPLEMENTATION

This section describes the practical implementation of the proposed grapevine leaf classification system, including dataset preparation, deep learning model training using MobileNetV2 with advanced data augmentations, feature extraction, feature selection

using Chi-square, and classification using a Support Vector Machine (SVM).

1. Dataset Acquisition

The Grapevine Leaves Image Dataset was obtained for images. It consists of images from five different grapevine varieties: Adakarası (Ak), Buzulu(Bz), Dimnit (Dm), Ala Idris, and Nazli (Nz). The dataset was downloaded and extracted into a local working directory.

2. Data Quality Preprocessing

Before model training, data quality checks were performed to remove:

- **Corrupted Images**: Detected using PIL's Image.verify() method.
- **Blurry Images**: Filtered using the Laplacian variance threshold (below 50 considered blurry).
- **Duplicates**: Detected using perceptual hashing (pHash) via the imagehash library.

This ensured that only high-quality and unique images were used for training.

3. Image Preprocessing

All valid images were resized to **224×224 pixels** and normalized by scaling pixel values to the range [0, 1]. Images were labeled using a folder-based labeling scheme and encoded using integer labels for further processing.

4. Advanced Data Augmentation

To address class imbalance and improve generalization, multiple advanced data augmentation techniques were applied:

- **Standard Keras Augmentation**: Rotation, shift, zoom, and horizontal flip via ImageDataGenerator.
- **RandAugment**: Applied using RandAugment layer from Keras for stochastic augmentation.
- **AugMix**: Implemented with Albumentations, combining blur, brightness, and rotation in a probabilistic chain.
- **MixUp**: Images and labels were mixed linearly with sampled lambda values from the Beta distribution.
- **CutMix**: Regions of one image were replaced with another's and labels were blended accordingly.

The final dataset (X_final, y_final) was formed by combining original and augmented images.

5. Dataset Splitting

The full dataset was split into **80% training** and **20% testing** using train_test_split with a fixed random seed for reproducibility.

6. Deep Learning Model: MobileNetV2

A pretrained **MobileNetV2** model was used as the base feature extractor, with its convolutional layers frozen. The classification head was added as follows:

- Global Average Pooling
- Dense layer (256 units, ReLU)
- Dropout layer (0.2 rate)
- Output Dense layer with 5 units (Softmax activation)

The model was compiled using the **Adam optimizer** and trained for **50 epochs** with **sparse categorical crossentropy** loss.

7. Feature Extraction

The convolutional backbone of MobileNetV2 was extracted as a separate feature extractor. It was used to convert both training and testing images into feature vectors by flattening the convolutional outputs. These vectors were further used for classical machine learning classification.

8. Feature Selection using Chi-Square

To reduce dimensionality and enhance SVM performance, **Chi-square feature selection** was

applied using SelectKBest with chi2 scoring. The top **250** most relevant features were retained.

9. SVM Classification

A **Support Vector Machine (SVM)** classifier was trained on the selected features. Multiple kernel options (linear, rbf, poly, cubic) were tested. The polynomial kernel (degree=3) gave the best results and was selected for the final model.

10. Model Evaluation

The SVM classifier was evaluated on the test set using various performance metrics:

- Accuracy
- Precision
- Recall (Sensitivity)
- F1-Score
- Matthews Correlation Coefficient (MCC)
- Confusion Matrix
- ROC-AUC Score

Visualizations such as heatmaps of confusion matrices were generated using seaborn.

11. Model Saving

All key components of the system were serialized and saved for integration into the Django web application:

- mobilenetv2_augmented.h5: Trained deep learning model.
- feature_extractor_advanced.h5: Feature extractor from MobileNetV2.
- svm_chi2_model_advanced.pkl: SVM classifier trained on Chi-square selected features.
- chi2_selector_advanced.pkl: Trained Chi-square feature selector.
- label_map_advanced.pkl: Class label-to-index mapping dictionary.

12. Django Deployment and System Integrartion

This section describes how the trained models and associated components are integrated into a Django-based web application for real-time inference and deployment. The deployment pipeline includes:

- **Model Loading:** The Django backend loads
  - ✓ the trained deep learning model (mobilenetv2_augmented.h5),
  - ✓ the feature extractor (feature_extractor_advanced.h5),
  - ✓ the Chi-square selector (chi2_selector_advanced.pkl), and
  - ✓ the trained SVM classifier (svm_chi2_model_advanced.pkl).
  - ✓ The label mapping dictionary (label_map_advanced.pkl) is also loaded to interpret predicted classes.
- **Image Upload & Preprocessing:** Users upload grapevine leaf images via a web interface. The server validates, resizes, and normalizes the images, ensuring consistency with the training preprocessing pipeline.
- **Feature Extraction & Classification:** The uploaded image is processed through the feature extractor to obtain deep features, then reduced via the Chi-square selector. The SVM classifier predicts the class label based on these features.
- **Result Presentation:** The predicted class, confidence scores, and relevant visualizations (e.g., confusion matrix, ROC curves) are generated and displayed on the web interface for user interpretation.
- **System Architecture & Workflow:** The deployment leverages Django's robust framework to facilitate scalable, real-time classification, enabling practical use in vineyard management and research.

EXPERIMENTAL RESULTS

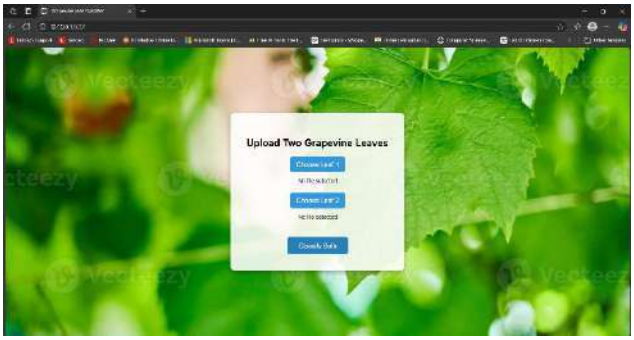

Fig 1: Graphical User Interface (GUI) Home Screen

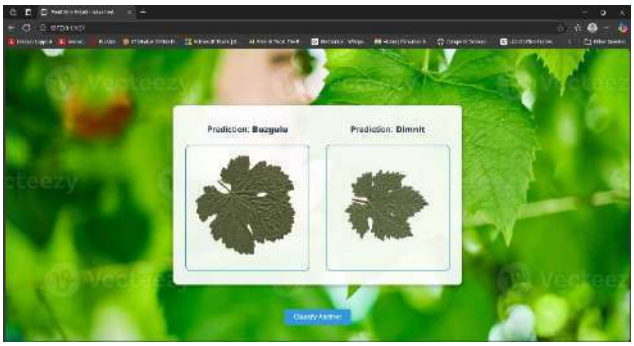

Fig 2: OUTPUT Screen

CONCLUSION

This project successfully developed an advanced grapevine leaf classification system by integrating deep learning and traditional machine learning techniques. Beginning with a rigorous data quality filtering process to eliminate corrupt, blurry, and duplicate images, the dataset was significantly refined to ensure high-quality input for model training. To further enhance robustness and generalization, a diverse set of augmentation techniques—including standard transformations, RandAugment, AugMix, Mixup, and CutMix—were applied, creating a significantly enriched training dataset.

A transfer learning approach using MobileNetV2 served as the foundation for feature extraction due to its efficiency and performance on image classification tasks. The trained CNN model achieved reliable results, and its intermediate features were extracted to serve as input for a classical Support Vector Machine (SVM) classifier. Among multiple kernel experiments, the polynomial kernel (degree=3), combined with Chi-square-based feature selection, yielded the best classification performance.

Comprehensive evaluation using metrics such as accuracy, precision, recall, F1 score, Matthews Correlation Coefficient (MCC), and ROC AUC confirmed the robustness and effectiveness of the final pipeline. The model achieved high classification performance across all five grapevine leaf classes, validated through confusion matrix analysis and ROC evaluation.

The finalized models—including the deep learning classifier, SVM with selected features, and the feature extractor—were saved for deployment or further research use. This modular and hybrid approach not only demonstrated improved classification accuracy but also provided flexibility for future enhancements, such as incremental learning or real-time deployment in agricultural field applications.

Overall, this project illustrates the potential of combining deep learning with feature selection and classical ML for solving complex visual classification problems in the agricultural domain.

REFERENCES

1. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
2. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
3. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 248-255).
4. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097-1105.
5. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
6. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on

Computer Vision and Pattern Recognition (pp. 770-778).
7. Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345-1359.
8. Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? Advances in Neural Information Processing Systems, 27, 3320-3328.
9. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.
10. Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2(3), 1-27.
11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

HOW TO CITE: D. Banu Kranthi*, Surya Lakshmi Ramabhotla, Grapevine Leaves Classification Using Advanced Data Augmentation Techniques, Int. J. Sci. R. Tech., 2026, 3 (5), 197-205. https://doi.org/10.5281/zenodo.20036123