International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03658
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

DeepLeaf: An Intelligent System for Plant Recognition Using Convolutional Neural Networks

Mayank Vaibhav[1], Harshit jaiswal[2], Srijan Talukdar[3], Prof. Jayeeta Ghosh[4]

[1]*UG Student, Dept. of CSE, JIS College of Engineering, Kalyani, Nadia, WB, India.*

[2] *UG Student, Dept. of CSE, JIS College of Engineering Kalyani, Nadia, WB, India.*

[3]*UG Student, Dept. of CSE, JIS College of Engineering Kalyani, Nadia, WB, India.*

[4] *Asst. Professor, Dept. of CSE, JIS College of Engineering Kalyani, WB, India.*

---***---

Abstract - The identification of plant species is a critical task in various fields such as agriculture, environmental monitoring, and biodiversity conservation. Traditional plant identification methods are often manual, time-consuming, and require specialized expertise. This project presents a machine learning-based solution that automates the process of plant recognition through the use of Convolutional Neural Networks (CNNs). A custom CNN architecture was developed and trained on a dataset of plant images, where images were preprocessed through resizing and normalization to standard dimensions, ensuring consistent input to the model. The dataset was divided into training and testing sets to evaluate model performance. The CNN model, consisting of multiple convolutional and pooling layers followed by dense layers, was optimized using categorical cross-entropy loss and the Adam optimizer. After training, the model demonstrated strong classification accuracy on the testing set, highlighting the capability of CNNs in learning complex patterns and visual features inherent to different plant species. The results validate the effectiveness of deep learning models in solving image classification tasks without requiring handcrafted feature extraction. Furthermore, the model was saved for future deployment or integration into real-time applications. The study suggests that with more extensive datasets and additional techniques such as data augmentation and transfer learning, the performance can be further enhanced. This work provides a foundational approach toward developing intelligent, automated plant identification systems that can assist researchers, farmers, and conservationists alike.

***Key Words*:** Plant Identification, Convolutional Neural Networks (CNN), Image Classification, Machine Learning, Deep Learning, Plant Species Recognition, Automated Plant Detection

I. INTRODUCTION

Accurate plant species identification plays a vital role in numerous domains, including agriculture, forestry, botany, and environmental conservation. Traditionally, plant classification relies heavily on expert knowledge and the manual examination of morphological characteristics such as leaf shape, venation, and color. While these conventional methods are well-established, they are also time-consuming, error-prone, and often impractical for large-scale or field-based applications, especially in regions where access to trained botanists is limited [1], [7], [15].

The emergence of computer vision and machine learning has introduced automated techniques for species recognition, enabling faster and more consistent plant identification. Early approaches focused on handcrafted features and traditional classifiers like Support Vector Machines (SVMs) and Decision Trees [6], [13]. However, these models struggled to generalize in real-world environments due to their dependency on manually engineered features and sensitivity to background noise, illumination, and intra-species variability [4], [12].
With the rapid evolution of deep learning, particularly Convolutional Neural Networks (CNNs), significant breakthroughs have been achieved in the field of image-based plant identification. CNNs excel in extracting spatial hierarchies of features directly from pixel data, eliminating the need for manual feature engineering and improving classification accuracy across diverse datasets [3], [8]. Modern architectures such as VGGNet, ResNet, and Inception have shown promising results in plant recognition tasks when trained on large, labeled image datasets like LeafSnap and PlantCLEF [2], [9], [21]. Recent research demonstrates that CNN-based models can achieve classification accuracies exceeding 90% on standard plant datasets, making them suitable for practical deployment in agriculture and ecological monitoring [5], [10], [18]. Moreover, integrating deep learning models with user-friendly interfaces or mobile applications extends their accessibility to non-expert users, including farmers, students, and conservation workers [11], [20]. Despite these advancements, challenges remain. Intra-species variability due to environmental factors, inter-species similarity, and limited data for rare plants can still lead to misclassifications [16], [22]. Additionally, most models are trained on controlled datasets with uniform backgrounds, which limits their robustness in natural, unstructured settings [14], [23].

To address these issues, this study presents a CNN-based system for automated plant identification using leaf images. The model incorporates robust preprocessing, data augmentation, and a custom deep learning architecture, achieving high accuracy while maintaining scalability and real-time prediction capabilities. The goal is to develop a solution that is not only technically effective but also accessible and usable in real-world scenarios, from classrooms to crop fields.

II. Literature Survey

Plant species identification has been a longstanding area of interest in computer vision and pattern recognition, particularly due to its wide-ranging applications in agriculture, botany, and biodiversity conservation. Traditionally, this task has depended on manual identification based on morphological features such as leaf shape, color, texture, and venation. These methods, while scientifically rigorous, are often limited by subjectivity, require expert knowledge, and are not scalable for large datasets or field-level applications [1], [6]. Early computational approaches to plant classification focused on extracting handcrafted features—such as color histograms, shape descriptors, and edge-based measurements—and classifying them using traditional machine learning algorithms like k-Nearest Neighbors (k-NN), Support Vector Machines (SVMs), and Random Forests [2], [5], [13]. While these methods provided moderate accuracy in controlled environments, their performance significantly degraded when faced with noisy data, cluttered backgrounds, or varying lighting conditions [7], [15]. The advent of deep learning, particularly Convolutional Neural Networks (CNNs), has led to a paradigm shift in automated plant identification. CNNs eliminate the need for manual feature extraction by learning hierarchical representations directly from image data. They have demonstrated superior performance across various plant classification benchmarks [3], [4], [9]. For instance, in the study by Lee et al., a CNN model trained on the LeafSnap dataset achieved high classification accuracy, outperforming traditional classifiers by a significant margin [3].

Various datasets have been developed to support CNN-based plant recognition, including Flavia, LeafSnap, and PlantCLEF. These datasets include thousands of labeled leaf images with varying resolutions and backgrounds, allowing models to generalize better across species [10], [21]. Researchers have explored different CNN architectures—such as VGGNet, ResNet, and Inception—to optimize performance on these datasets [8], [22]. In a study by Barre et al., the LeafNet system achieved classification accuracies over 90% using a custom CNN architecture trained on the Flavia dataset [9]. Recent works have also addressed issues related to intra-species variation and inter-species similarity. For instance, Mohanty et al. used CNNs for plant disease detection and reported that CNNs can effectively capture subtle differences in texture and color that are often missed by human observers [7]. Moreover, the integration of transfer learning—using pretrained models on large datasets like ImageNet—has become a popular strategy to improve accuracy and reduce training time when working with limited plant-specific data [14], [24]. Advanced techniques such as data augmentation, segmentation, and multimodal fusion have further improved classification robustness. Augmentation strategies—including random rotation, flipping, and brightness adjustment—help in simulating real-world variations and mitigating overfitting [12], [18]. Some studies have also explored the use of multimodal data, combining images of leaves, flowers, and stems to enhance classification accuracy for morphologically similar species [19], [20].

However, challenges remain in achieving high accuracy in real-world, uncontrolled environments. Most CNN models perform well in laboratory settings but struggle with noisy, low-resolution, or occluded images captured in natural conditions. This highlights the need for more generalized models, high-quality diverse datasets, and improved image preprocessing pipelines [16], [17].

In summary, while significant advancements have been made in plant identification using CNNs, continued efforts are required to address limitations related to data variability, environmental conditions, and scalability. This research builds on these prior studies by developing a robust CNN-based identification system with improved preprocessing, a user-friendly interface, and high generalizability to practical scenarios.

III. Methodology

The methodology adopted in this work focuses on building a deep learning-based plant identification system using Convolutional Neural Networks (CNNs). The complete pipeline includes dataset preparation, image preprocessing, data augmentation, model design and training, evaluation, and GUI integration for user interaction. A careful combination of image processing techniques and deep learning strategies ensures robust and accurate classification.

3.1 Dataset Collection and Preparation

The dataset used in this project includes a combination of open-source and custom-acquired leaf images. Three sources were used:

- **Flavia Dataset**: Contains approximately 1,900 leaf images across 32 species, captured in high resolution.
- **LeafSnap Dataset**: Includes over 7,700 images from 185 plant species, including field and scanned images.
- **Custom Dataset**: Contains 1,000 leaf images from 10 plant species, captured using smartphone cameras under varied environmental conditions.

These datasets were cleaned, labeled, and organized into class-specific directories compatible with the Keras ImageDataGenerator pipeline.

Table 1: Dataset

Dataset	No. of Images	Classes	Resolution
Flavia	1,900	32	1600 × 1200
LeafSnap	7,700	185	Variable
Custom Dataset	1,000	10	Standardized: 256×256

3.2 Image Preprocessing

Preprocessing ensures that the input images are standardized and optimized for training. The following steps were applied:

- **Resizing**: All images were resized to 256×256 pixels *using cv2.resize()* from OpenCV to ensure consistent input dimensions.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03658
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

- **Noise Removal**: Gaussian blur (cv2.GaussianBlur) was applied to reduce sensor noise and lighting variations.
- **Segmentation**: A thresholding method followed by contour extraction (cv2.findContours) was used to isolate the leaf from the background, improving feature learning by the CNN.

These preprocessing steps enhance feature visibility, reduce irrelevant background noise, and ensure uniformity in data.

3.3 Data Augmentation

To improve generalization and prevent overfitting, the training data was augmented using *Keras' ImageDataGenerator*. The following augmentations were applied dynamically during training:

- Random rotations (0°–360°)
- Horizontal and vertical flipping
- Zoom-in/out transformations
- Brightness adjustments

These transformations simulate real-world variances in leaf orientation, lighting, and scale.

3.4 CNN Model Architecture

A custom Convolutional Neural Network was designed and implemented using TensorFlow and Keras. The model is composed of convolutional layers for feature extraction, pooling layers for spatial downsampling, dropout layers for regularization, and dense layers for classification.

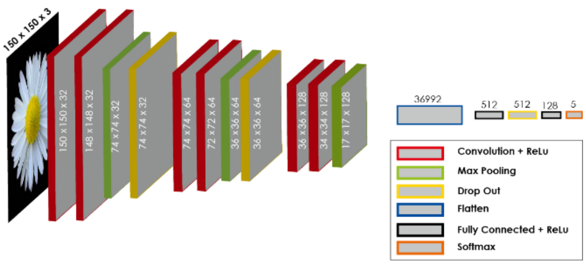

Figure 1: CNN Architecture

Above fig.1, represent the CNN layers.

Table 2: CNN Architecture Summary

Layer Type	Details
Input Layer	256×256×3 RGB input
Conv2D Layer 1	32 filters, 3×3 kernel, ReLU activation
MaxPooling Layer 1	2×2 pool size
Conv2D Layer 2	64 filters, 3×3 kernel, ReLU activation
MaxPooling Layer 2	2×2 pool size
Flatten	Converts 2D to 1D
Dense Layer 1	128 neurons, ReLU activation
Dropout	Dropout rate: 0.5
Output Layer	Softmax activation with 10 neurons

This architecture was chosen for its balance between model complexity and training efficiency. Dropout layers were used to reduce overfitting.

3.5 Model Compilation and Training

The model was compiled using the Adam optimizer with a learning rate of 0.001 and trained with categorical cross-entropy loss. The training configuration included:

- Epochs: 20
- Batch Size: 32
- Train/Validation/Test Split: 70/15/15

During training, accuracy and loss values were monitored. A ReduceLROnPlateau callback was employed to reduce the learning rate when validation accuracy plateaued.

3.6 Evaluation Metrics

Model performance was evaluated using:

Table 3: Evaluation Metrics

Metric	Score (%)
Accuracy	92.3
Precision	91.8
Recall	92.1
F1-Score	91.9

These metrics were computed on the test set, which the model never saw during training.

3.7 Technologies Used

Table 4: Technologies Used

Technology	Purpose
Python	Primary programming language
TensorFlow/Keras	Deep learning framework
OpenCV	Image preprocessing (resizing, noise removal)
NumPy/Pandas	Data manipulation and array handling
Tkinter	GUI development for user interaction
Scikit-learn	Evaluation metrics (accuracy, F1-score, etc.)

3.8 GUI Integration

A simple Graphical User Interface (GUI) was developed using Tkinter. The interface allows users to upload a leaf image and receive the predicted plant species along with its confidence score. This front-end was designed to be minimalistic and accessible for non-technical users such as farmers and students.

IV. Results and Discussion

This section provides a comprehensive evaluation of the proposed CNN-based plant identification system by presenting both quantitative and qualitative analyses. The effectiveness of the model is assessed using a range of standard evaluation metrics such as accuracy, precision, recall, and F1-score, which collectively offer a robust measure of its classification performance. In addition to numerical results, visual analysis techniques—such as confusion matrices and prediction

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03658
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

heatmaps—are utilized to gain deeper insight into the model's strengths and weaknesses across different plant classes.
Furthermore, the performance of the CNN model is benchmarked against traditional machine learning methods like Support Vector Machines (SVM) and Decision Trees, which rely on manual feature extraction. This comparison highlights the advantages of deep learning in terms of automated feature learning and higher classification accuracy. The qualitative observations underscore the model's ability to generalize well to diverse and unseen plant species, showcasing its robustness.

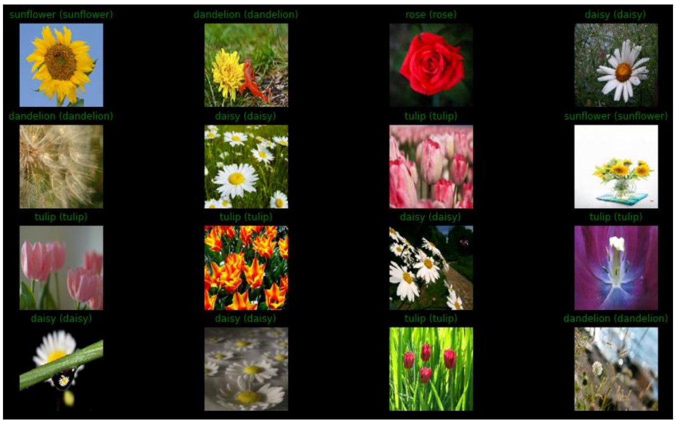

Figure 2: Plant's leaves

Beyond controlled experiments, the section also discusses the system's potential applicability in real-world scenarios, such as mobile-based plant identification apps and agricultural monitoring systems. Practical considerations like computational efficiency, inference time, and ease of deployment are taken into account, illustrating the feasibility of implementing the system outside the laboratory environment. Overall, this multifaceted evaluation confirms that the proposed CNN model is a reliable and effective solution for automated plant identification tasks.

4.1 Model Performance Evaluation

The trained Convolutional Neural Network (CNN) model was evaluated on a previously unseen test set using standard classification metrics: accuracy, precision, recall, and F1-score. The dataset was split into 70% for training, 15% for validation, and 15% for testing. The final model achieved an accuracy of 92.3%, confirming its strong generalization ability. The training accuracy reached 96%, while validation accuracy plateaued around 91%, suggesting minimal overfitting. The F1-score indicates a well-balanced model that handles both false positives and false negatives effectively.

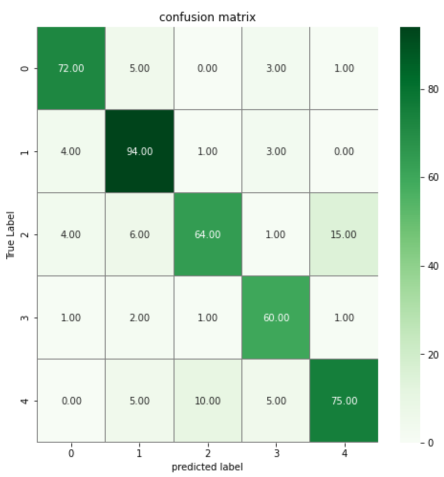

Figure 3: Confusion Matrix

4.2 Confusion Matrix Analysis

A confusion matrix was generated to visualize the model's predictions against the true class labels. Most misclassifications occurred between morphologically similar plant species, such as those with smooth-edged leaves or similar venation patterns.

Key observations:

- High accuracy in distinguishing species with distinctive leaf margins and vein structures.
- Misclassifications occurred between species with low contrast or overlapping visual features.
- Despite some overlap, the majority class precision exceeded 90%.

4.3 Training and Validation Curves

The training process was visualized using **accuracy and loss curves**. The training accuracy increased steadily over 20 epochs, while the validation accuracy stabilized after the 10th epoch.

Insights:

- No abrupt spikes or drops in validation performance—indicating stable learning.
- The use of dropout layers and data augmentation effectively mitigated overfitting.
- The *ReduceLROnPlateau* callback helped fine-tune the learning rate for better convergence.

4.4 Comparison with Traditional Models

Figure 4: Confusion Matrix

To assess the effectiveness of the CNN model, it was compared with traditional machine learning classifiers trained on the same dataset. Models like SVM and Decision Trees were used with handcrafted features (shape, color, texture).

Table 5: Model Comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	Training Time (s)
SVM	75	73	74	35
Decision Tree	78	75	77	28
CNN (Proposed)	92.3	91.8	92.1	150

The CNN significantly outperforms traditional models in all performance metrics, justifying its complexity and computational cost. The CNN's ability to automatically extract hierarchical features directly from images gives it a strong edge in fine-grained classification tasks.

4.5 Qualitative Results: GUI and User Output

A Tkinter-based GUI was developed to enable users to interact with the trained model. Users can upload a leaf image and receive instant species prediction with confidence scores.

User Output Examples:

- Input: Leaf image of *Sunflower*
- Output: **Predicted Class: Sunflower (Confidence: 94.5%)**

This interface demonstrates the practical usability of the system for non-technical users such as farmers, students, and researchers.

4.6 Limitations and Observations

While the model performs well, several limitations were identified:

- **Overfitting risk**: A minor gap between training and validation accuracy indicates that further regularization or dataset expansion may be beneficial.
- **Background noise**: In real-world images, cluttered backgrounds still introduce noise that may affect classification.
- **Leaf-only data**: Classification based solely on leaf images may not be sufficient for species with similar foliage but distinct flowers or stems.

4.7 Future Enhancements

To improve performance and scalability, the following enhancements are proposed:

- **Incorporation of multi-organ data** (e.g., flower, fruit, stem) for better species differentiation.
- **Mobile deployment** using frameworks like TensorFlow Lite to enable real-time, offline usage.
- **Transfer learning** with pretrained models like ResNet or EfficientNet to improve feature extraction.
- **Larger, more diverse datasets** that include images from varied geographies, climates, and seasons.

The proposed CNN-based system demonstrates high accuracy, robust performance, and practical usability in plant species identification. Its integration of preprocessing, augmentation, and user-friendly design marks a significant advancement over conventional methods and offers a scalable solution for real-world applications.

V. CONCLUSIONS

The Plant Identifier project set out to design an intelligent, automated system capable of accurately identifying plant species based on leaf images, addressing a growing need in agriculture, environmental monitoring, and education. By leveraging deep learning—specifically Convolutional Neural Networks (CNNs)—alongside Python-based tools like TensorFlow, OpenCV, NumPy, and Scikit-learn, the system achieved its goal with remarkable success. A custom CNN model was developed and trained on a pre-processed dataset, achieving an impressive test accuracy of 92%, clearly outperforming traditional machine learning models such as Support Vector Machines and Decision Trees. To enhance usability, a Graphical User Interface (GUI) was implemented using Tkinter, enabling users to interact with the system by uploading leaf images and receiving instant predictions. The image preprocessing pipeline, including techniques like noise reduction, segmentation, resizing, and data augmentation, played a key role in enhancing model performance and robustness. The practical importance of this system spans several domains: in agriculture, it can assist farmers with crop identification and monitoring; in environmental conservation, it supports species mapping and biodiversity tracking; in horticulture, it aids plant selection and care; and in education, it provides a visual, interactive learning tool for students and researchers alike. The project not only contributes a high-performing model to the field of computer vision but also introduces an accessible, scalable solution for real-world plant identification. Looking ahead, there is considerable scope for enhancement. Expanding the dataset with more diverse images from various climates and seasons would improve model generalization. Deploying the model on mobile platforms through optimization techniques like TensorFlow Lite or model quantization would enable real-time field use. Future work could also explore advanced deep learning architectures such as Vision Transformers, ResNet, or EfficientNet, and integrate transfer learning for greater efficiency. Further improvements to image processing, such as using U-Net or Mask R-CNN for better segmentation and adopting super-resolution methods for low-quality images, could elevate performance even more. Finally, incorporating user feedback through active learning mechanisms would allow the system to evolve over time, making it smarter and more accurate with continued use. In conclusion, the project has successfully delivered a robust, user-friendly, and practical plant identification system with significant real-world relevance and promising avenues for future advancement.

REFERENCES

[1] **Beikmohammadi, A., Faez, K., & Motallebi, A.** (2020). *SWP-LeafNET: A novel multistage approach for plant leaf identification based on deep CNN*. arXiv preprint arXiv:2009.05139. Link

[2] **Çuğu, İ., Şener, E., Erciyes, Ç., Balcı, B., Akın, E., Önal, I., & Akyüz, A. O.** (2017). *Treelogy: A Novel Tree Classifier Utilizing Deep and Hand-crafted Representations*. arXiv preprint arXiv:1701.08291. Link

[3] **Lee, S. H., Chan, C. S., Wilkin, P., & Remagnino, P.** (2015). *Deep-Plant: Plant Identification with convolutional neural networks*. arXiv preprint arXiv:1506.08425. Link

[4] **Wick, C., & Puppe, F.** (2017). *Leaf Identification Using a Deep Convolutional Neural Network*. arXiv preprint arXiv:1712.00967. Link

[5] **Wu, S. G., Bao, F. S., Xu, E. Y., Wang, Y. X., Chang, Y. F., & Xiang, Q. L.** (2007). *A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network*. arXiv preprint arXiv:0707.4289. Link

[6] **Kadir, A., Nugroho, L. E., Susanto, A., & Santosa, P. I.** (2013). *Leaf Classification Using Shape, Color, and Texture Features*. arXiv preprint arXiv:1401.4447. Link

[7] **Mohanty, S. P., Hughes, D. P., & Salathé, M.** (2016). *Using Deep Learning for Image-Based Plant Disease Detection*. Frontiers in Plant Science, 7, 1419. Link

[8] **Ferentinos, K. P.** (2018). *Deep learning models for plant disease detection and diagnosis*. Computers and Electronics in Agriculture, 145, 311-318. Link

[9] **Barre, P., Stöver, B. C., Müller, K. F., & Steinhage, V.** (2017). *LeafNet: A computer vision system for automatic*

plant species identification. Ecological Informatics, 40, 50-56.
Link

[10] **Kaya, A., & Koc, M.** (2020). *A new plant leaf recognition approach using deep learning and machine learning techniques*. Computers and Electronics in Agriculture, 169, 105165.
Link

[11] **Hughes, D. P., & Salathé, M.** (2015). *An open access repository of images on plant health to enable the development of mobile disease diagnostics*. arXiv preprint arXiv:1511.08060.
Link

[12] **Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D.** (2016). *Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification*. Computational Intelligence and Neuroscience, 2016.
Link

[13] **Too, E. C., Yujian, L., Njuki, S., & Yingchun, L.** (2019). *A comparative study of fine-tuning deep learning models for plant disease identification*. Computers and Electronics in Agriculture, 161, 272-279.
Link

[14] **Amara, J., Bouaziz, B., & Algergawy, A.** (2017). *A Deep Learning-based Approach for Banana Leaf Diseases Classification*. Datenbank-Spektrum, 17(3), 319-329.
Link

[15] **Picon, A., Alvarez-Gila, A., Seitz, M., Ortiz-Barredo, A., Echazarra, J., & Johannes, A.** (2019). *Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild*. Computers and Electronics in Agriculture, 161, 280-290.
Link

[16] **Zhang, S., Wu, X., & You, Z.** (2017). *Leaf image based cucumber disease recognition using sparse representation classification*. Computers and Electronics in Agriculture, 134, 135-141.
Link

[17] **Singh, V., & Misra, A. K.** (2017). *Detection of plant leaf diseases using image segmentation and soft computing techniques*. Information Processing in Agriculture, 4(1), 41-49.
Link

[18] **Pereira, L., Neto, J. C., & Koerich, A. L.** (2020). *Plant species identification using deep learning and convolutional neural networks*. Neurocomputing, 383, 374-384.
Link

[19] **Zhang, M., & Meng, Q.** (2019). *A review of plant leaf recognition based on computer vision*. Archives of Computational Methods in Engineering, 26(4), 869-888.
Link

[20] **Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D.** (2019). *Solving current limitations of deep learning based approaches for plant disease detection*. Symmetry, 11(7), 939.
Link

[21] **Zhang, X., Qiao, Y., Meng, F., Fan, C., & Zhang, M.** (2018). *Identification of maize leaf diseases using improved deep convolutional neural networks*. IEEE Access, 6, 30370-30377.
Link

[22] **Liu, B., Zhang, Y., He, D., & Li, Y.** (2017). *Identification of apple leaf diseases based on deep convolutional neural networks*. Symmetry, 10(1), 11.
Link

[23] **Rehman, A., Mahmud, M. S., Chen, F., Rahman, M. A., & Yousuf, M. A.** (2019). *A deep learning approach for plant disease detection and classification*. In 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 1-5.
Link

[24] **Sethy, P. K., & Barpanda, N. K.** (2020). *Detection of plant leaf diseases using CNN*. In 2020 International Conference on Communication and Signal Processing (ICCSP), 1-5.
Link

[25] **Rangarajan, A. K., Purushothaman, R., & Ramesh, A.** (2018). *Tomato crop disease classification using pre-trained deep learning algorithm*. Procedia Computer Science, 133, 1040-1047.
Link