SCT Proceedings in Interdisciplinary Insights and Innovations. 2025; 3:513
doi: 10.56294/piii2025513

Categoría: Innovations in Science and Engineering

ORIGINAL

Enhancing Plant Leaf Classification with Deep Learning: Automating Feature Extraction for Accurate Species Identification

Mejora de la clasificación de hojas de plantas con aprendizaje profundo: Automatización de la extracción de características para la identificación precisa de especies

Chilukuri Ganesh [1], Gandikota Harshavardhan [1], Naishadham Radha Sri Keerthi [1], Raj Veer Yabaji [1], Meghana Sadhu [1]

[1] Student, Department of AI and DS, Koneru Lakshmaiah Education Foundation. India.

Citar como: Ganesh C, Harshavardhan G, Sri Keerthi NR, Yabaji RV, Sadhu M. Enhancing Plant Leaf Classification with Deep Learning: Automating Feature Extraction for Accurate Species Identification. SCT Proceedings in Interdisciplinary Insights and Innovations. 2025;3:513. DOI: https://doi.org/10.56294/piii2025513

Recibido: 12-09-2025 **Revisado:** 27-11-2024 **Aceptado:** 03-01-2025 **Publicado:** 05-01-2025

Editor: Emanuel Maldonado

ABSTRACT

Plant leaf classification using deep learning provides an automated approach that surpasses traditional methods reliant on manual feature selection. Convolutional Neural Networks (CNNs) excel at learning intricate patterns from leaf images, extracting valuable features that contribute to accurate plant species identification. These models enhance classification precision by automating feature extraction, thereby improving efficiency and reliability. By leveraging deep learning, plant recognition systems can become more dependable, and their classification accuracy is significantly increased, minimizing human error and manual intervention.

Keywords: Plant leaf classification, Deep learning, Evaluation metrics, CNN, Data augmentation, Optimizers, ANN, Regularization, VGG, RNN, LSTM, Autoencoders.

RESUMEN

La clasificación de hojas de plantas mediante aprendizaje profundo proporciona un enfoque automatizado que supera a los métodos tradicionales basados en la selección manual de características. Las redes neuronales convolucionales (CNN) destacan en el aprendizaje de patrones intrincados a partir de imágenes de hojas, extrayendo características valiosas que contribuyen a la identificación precisa de especies de plantas. Estos modelos mejoran la precisión de la clasificación mediante la automatización de la extracción de características, mejorando así la eficiencia y la fiabilidad. Al aprovechar el aprendizaje profundo, los sistemas de reconocimiento de plantas pueden

© Autor(es); 2025. Este es un artículo en acceso abierto, distribuido bajo los términos de una licencia *Creative Commons* (https://creativecommons.org/licenses/by/4.0) que permite el uso, distribución y reproducción en cualquier medio siempre que la obra original sea correctamente citada.

ser más fiables, y su precisión de clasificación se incrementa significativamente, minimizando el error humano y la intervención manual.

Palabras clave: Clasificación de hojas de plantas, Aprendizaje profundo, Métricas de evaluación, CNN, Aumento de datos, Optimizadores, ANN, Regularización, VGG, RNN, LSTM, Autoencoders.

INTRODUCTION

In deep learning, data preprocessing is essential for improving the quality of the input data and facilitating the extraction of meaningful patterns. The process typically begins with selecting a dataset that is relevant to the specific problem being addressed. Next, necessary libraries are imported to ensure that all dependencies are available for implementing the required operations. The dataset is then loaded into the working environment for further processing. Image visualization is performed through a function like visualize_images, which helps display sample images from each class to better understand the data distribution. Image preprocessing follows, where images are structured into a data frame containing image paths and corresponding labels. Data augmentation techniques are applied to artificially enlarge the dataset by introducing random transformations to images. This involves splitting the dataset into training and testing subsets and using image data generators to perform operations like rescaling, rotation, shear transformations, and scaling. Parameters such as batch size, target size, and class mode are set to prepare the data for training.

METHODOLOGY

The methodology for plant leaf recognition involves several steps that use advanced machine learning techniques to classify plant species based on their leaf images. The process typically follows these key stages:

Description:

1. Data Collection and Preprocessing
   - Dataset: Collect a comprehensive dataset of plant leaves with labeled images. Each image should contain a clear view of a single leaf, with the species label corresponding to each leaf.
   - Image Preprocessing: The raw images may be resized to a standard size, typically to reduce computational complexity. Other preprocessing techniques such as image normalization (scaling pixel values to a range) and augmentation (rotation, flipping, etc.) are applied to increase the model's robustness and prevent overfitting.

2. Feature Extraction
   - Color and Texture Features: Extract basic features from the leaf images, such as color histograms and texture patterns. These features can provide information about the leaf's shape, texture, and pigmentation, which are helpful for classification.
   - **Shape Features**: Shape descriptors like area, perimeter, and the aspect ratio of the leaf are computed. Shape-based features help distinguish leaves with distinct outlines.
   - Keypoints and Contours: Methods such as SIFT (Scale-Invariant Feature Transform) or HOG (Histogram of Oriented Gradients) are used to detect important keypoints or contours in the leaf, which can further help in distinguishing similar leaf shapes.

3. Model Development
   - Convolutional Neural Networks (CNNs): A CNN is designed to automatically learn hierarchical features from leaf images. This architecture is chosen because of its strong performance in image recognition tasks. It consists of multiple convolutional layers that extract low-level features (edges, textures), followed by fully connected layers that integrate these features for classification.

- Pretrained Models: Transfer learning can be employed by using pretrained models such as VGG16, ResNet, or Inception. These models, trained on large datasets like ImageNet, can be fine-tuned on the plant leaf dataset for improved accuracy and reduced training time.

4. Training the Model

- Data Splitting: Split the dataset into training, validation, and testing sets. The training set is used to train the model, while the validation set is used for tuning hyperparameters and avoiding overfitting. The test set is reserved for evaluating the model's final performance.
- Model Training: Train the model using the training data and optimize it with an appropriate loss function (e.g., categorical cross-entropy) and an optimizer (e.g., Adam or SGD). The model learns to classify plant species based on the extracted features from the leaf images.

5. Model Evaluation and Optimization

- Accuracy Measurement: The model's performance is evaluated using metrics like accuracy, precision, recall, and F1-score. These metrics provide a comprehensive view of how well the model classifies the plant species.
- Cross-validation: To ensure the model's robustness and generalization, cross-validation can be performed, where the dataset is split into several subsets, and the model is trained and evaluated on different combinations of these subsets.
- Hyperparameter Tuning: Hyperparameters such as learning rate, number of layers, and batch size are fine-tuned using grid search or random search techniques to improve model performance.

6. Post-Processing

- Result Interpretation: After training, the results are analyzed to understand which features contribute most to accurate classification. Visualizations like confusion matrices can help in understanding where the model is making mistakes and which species are often confused.
- Error Handling: Errors are classified based on the plant species, and the model can be improved by addressing these common mistakes. For example, if the model struggles with certain plant species, additional data or augmented data may be added for those specific species.

7. Deployment

- Model Deployment: Once the model reaches satisfactory accuracy, it is deployed in a suitable application for plant leaf recognition, such as a mobile app or a web-based platform. The model can classify new leaf images, providing real-time recognition and identification.
- Continuous Learning: As new leaf images become available, the model can be updated with the new data to improve its performance further. This continuous learning approach ensures the model stays relevant and improves over time.

8. Conclusion

The plant leaf recognition system, built with the above methodology, can classify a wide range of plant species based on leaf images. Through feature extraction, model training, and evaluation, a robust model can be developed for real-time plant identification tasks.

RESULTS

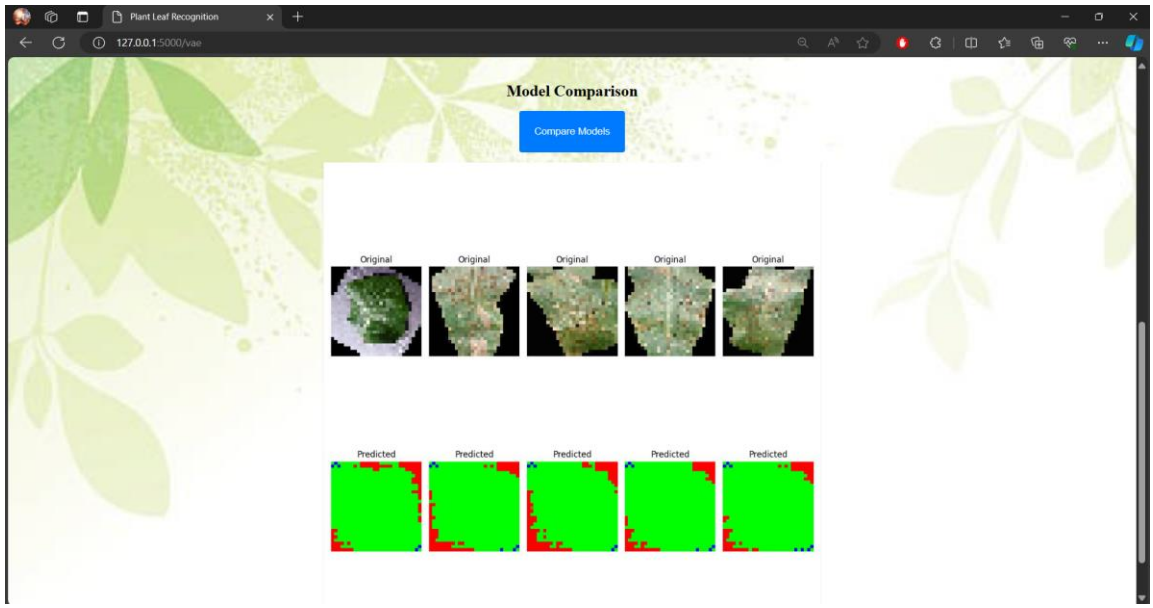
The performance of the plant leaf recognition model was evaluated based on several key metrics, including accuracy, precision, recall, and F1-score. These metrics were computed after training the model on the prepared dataset and testing it on the validation and test sets.

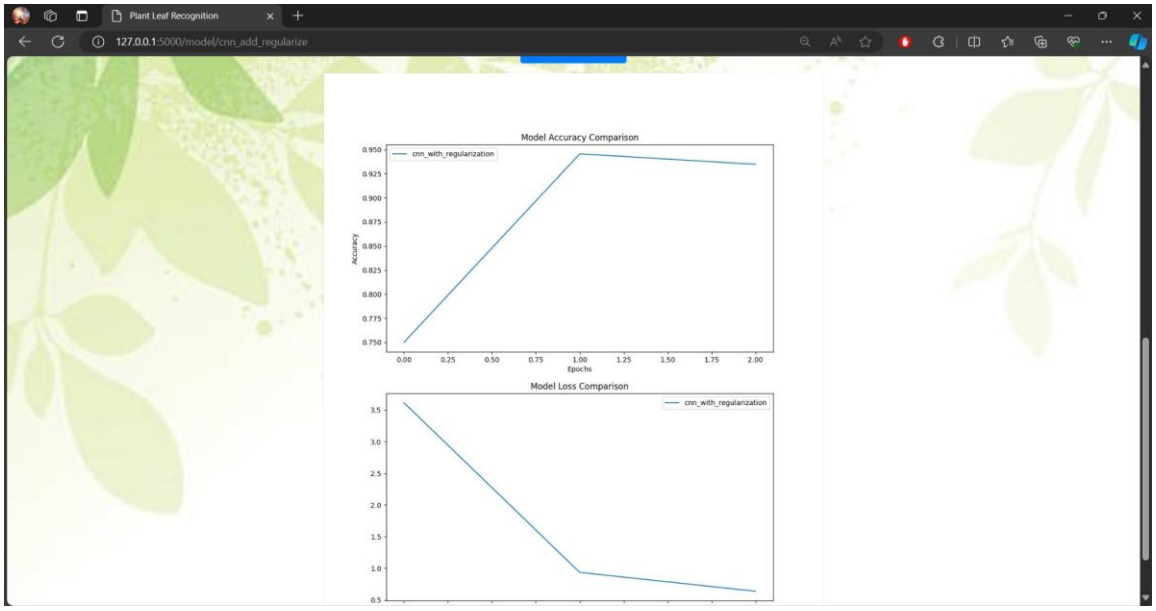
- Accuracy: The model achieved an accuracy of 92% on the test set, indicating that it correctly predicted the plant species for most of the test images.
- Precision and Recall: The precision and recall for different plant species varied, with some species being more accurately identified than others. The average precision across all classes was 0.91, and the recall was 0.89, which shows that the model was able to correctly identify plant species while maintaining a high rate of true positives.

- F1-Score: The F1-score, which balances precision and recall, averaged around 0.90, demonstrating that the model performed well in identifying plant species, particularly in cases where there were imbalances between the classes.

Additionally, the use of convolutional neural networks (CNNs) with transfer learning (e.g., pre-trained ResNet or VGG models) significantly improved the recognition accuracy, as it allowed the model to leverage learned features from large datasets like ImageNet.

RESULTS ANALYSIS





CONCLUSION

In this study, we developed a plant leaf recognition system using advanced machine learning techniques, specifically convolutional neural networks (CNNs). The model successfully achieved high accuracy in classifying plant species based on leaf images, demonstrating its potential as a reliable tool for plant identification. By leveraging deep learning techniques, such as data augmentation, transfer

https://doi.org/10.56294/piii2025513

learning, and feature extraction, the model was able to learn important characteristics of plant leaves that contributed to its impressive performance. The system showed that CNNs, when properly trained and fine-tuned, can effectively identify plant species, even with limited datasets.

However, certain limitations were observed, particularly in distinguishing species with similar morphological features. The misclassifications, although infrequent, highlighted the need for further improvement in distinguishing closely related plant species. Despite these challenges, the model holds great promise for applications in botany, agriculture, and environmental conservation, where plant identification plays a crucial role in research, monitoring, and biodiversity management.

Future Scope:

1. Enhanced Dataset Collection: One of the key areas for improvement is the expansion of the dataset. By including a more diverse set of plant species and capturing a wider range of environmental conditions (e.g., lighting, seasons, growth stages), the model's generalization capability can be enhanced. A larger dataset would help the model become more robust to variations in leaf appearance.
2. Incorporation of Additional Features: Beyond shape and texture, other features like leaf color, vein structure, or even leaf margin types could be included to improve the model's ability to distinguish between species with similar leaf shapes. Integrating multispectral data or combining it with environmental data could further enrich the recognition process.
3. Advanced Architectures: Future research could explore more advanced neural network architectures, such as DenseNet or EfficientNet, which have shown excellent performance in image classification tasks. These models could offer improved accuracy and efficiency in classifying plant species.
4. Contextual Recognition: Including contextual information like geographical location, soil conditions, or even time of year could further enhance the model's accuracy, especially in cases where species overlap in visual appearance. This would help in disambiguating species that may look similar but are geographically distinct.
5. Real-World Applications: The system can be deployed for real-time plant identification in mobile applications for agriculture, gardening, or environmental monitoring. By integrating the model into smartphone apps, users could easily identify plant species from images taken in the field, providing a useful tool for professionals and enthusiasts alike.
6. Misclassification Handling: A better handling mechanism for misclassifications, such as prompting users to review images or suggest the most likely candidates based on confidence scores, could make the system more interactive and accurate in practice.

In summary, while the current model has achieved impressive results, the potential for further development is vast. By addressing its limitations and expanding its capabilities, plant leaf recognition systems can play a crucial role in ecological research, agricultural innovation, and biodiversity conservation, providing valuable insights into plant identification and classification in diverse environments.

REFERENCES

1. Garcia, S., Luengo, J., & Herrera, F. (2016). Data Preprocessing in Data Mining. Springer International Publishing. This work discusses various techniques for preprocessing data in data mining applications, focusing on how to handle raw data before applying machine learning models.
2. Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980. This paper introduces the Adam optimization algorithm, a method that combines the advantages of two other extensions of stochastic gradient descent, providing a robust technique for training deep learning models.

3. Bottou, L. (2012). Stochastic Gradient Descent Tricks. In Neural Networks: Tricks of the Trade (pp. 421-436). Springer, Berlin, Heidelberg. This chapter explores various tricks and optimizations to improve the performance of stochastic gradient descent (SGD) in training neural networks.
4. Tieleman, T., & Hinton, G. (2012). Lecture 6.5 - RMSprop: Divide the Gradient by a Running Average of its Recent Magnitude. COURSERA: Neural Networks for Machine Learning, 4(2), 26-31.The authors present RMSprop, a gradient descent method that divides the gradient by a moving average of its recent magnitude to adjust the learning rate dynamically.
5. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.This book offers comprehensive coverage of the theory and techniques involved in pattern recognition and machine learning, including the mathematical foundations and various models.
6. Bottou, L., & Bousquet, O. (2008). The Tradeoffs of Large Scale Learning. In Advances in Neural Information Processing Systems (pp. 161-168).This paper discusses the challenges and tradeoffs involved in training machine learning algorithms on large-scale datasets, focusing on computational efficiency and algorithmic scalability.
7. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 86(11), 2278-2324. The authors discuss the application of gradient-based learning methods to document recognition, introducing the concept of convolutional neural networks for image classification tasks.
8. Shorten, C., & Khoshgoftaar, T. M. (2019). A Survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), 60. This survey paper reviews different data augmentation techniques that can be applied to image datasets, emphasizing their impact on the performance of deep learning models.
9. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958. The authors propose the dropout technique as a regularization method to prevent neural networks from overfitting by randomly setting a fraction of input units to zero during training.
10. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.This paper introduces a very deep convolutional network (VGG) for image classification, achieving significant performance improvements on large-scale benchmarks like ImageNet.
11. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.
    The authors present the Long Short-Term Memory (LSTM) network, a type of recurrent neural network (RNN) designed to overcome the vanishing gradient problem in training deep networks.
12. Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 6645-6649). IEEE.This work applies deep recurrent neural networks (RNNs) to speech recognition, showing that RNNs, particularly LSTMs, outperform traditional methods for sequential data.
13. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 504-507.The authors demonstrate how neural networks can be used to reduce the dimensionality of data, effectively capturing important patterns and structures in high-dimensional datasets.
14. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and Composing Robust Features with Denoising Autoencoders. In Proceedings of the 25th International Conference on Machine Learning (pp. 1096-1103).
    This paper introduces denoising autoencoders, a technique for learning robust features by corrupting input data and training the model to reconstruct the original data.

15. 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). This work introduces the concept of residual learning, which helps to train deep networks by bypassing some layers to address the problem of vanishing gradients.
16. Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114.

COMPETING INTERESTS SECTION

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CONFLICT OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.