

LEAFING THROUGH DATA: BUILDING A PLANT SPECIES IDENTIFICATION ALGORITHM WITH MACHINE LEARNING

D. Deepthi^{1*}, K. Nikhil², M. Mahesh Babu², V. Koti², B. Pavan Krishna²

¹ Assistant Professor, ²UG Student, ^{1,2} Department of Computer Science and Engineering (AIML)

^{1,2}Sree Dattha Institute of Engineering and Science, Sheriguda, Hyderabad, Telangana

ABSTRACT

Accurate identification of plant species is vital for biodiversity conservation, agriculture, and ecological research, yet traditional methods involving field guides, botanical keys, and expert consultation are time-consuming, subjective, and demand specialized knowledge. These limitations make large-scale assessments inefficient and prone to error. To address this challenge, the proposed system introduces an automated image-based plant species identification solution, aiming to enhance scalability, accessibility, and accuracy. The system involves collecting a comprehensive, labeled dataset of plant images, followed by preprocessing to improve input quality. Convolutional Neural Networks (CNNs) are employed to extract key visual features automatically, which are then used to train and evaluate machine learning models. Multiple algorithms will be tested and fine-tuned to ensure optimal performance, with the ultimate goal of streamlining species identification for researchers, farmers, and conservationists alike.

Keywords: Plant Identification, Biodiversity, Convolutional Neural Network, Image Classification, Automation

1. INTRODUCTION

The history of plant species identification dates back to ancient times when humans first began to recognize and categorize plants for medicinal, nutritional, and practical purposes. Early civilizations, such as the Egyptians, Greeks, and Chinese, documented plant species and their uses in herbal texts and pharmacopoeias. One of the earliest known examples is the Ebers Papyrus from ancient Egypt, which contains detailed information about medicinal plants and their applications.

During the Middle Ages, the study of plants became more systematic with the advent of herbals, which were books that described plants' medicinal properties and often included illustrations. Notable herbals from this period include the works of Dioscorides, an ancient Greek physician whose book "De Materia Medica" served as a cornerstone for botanical knowledge for centuries. These early texts laid the foundation for the field of botany, which began to develop as a distinct scientific discipline during the Renaissance.

The Renaissance period saw significant advancements in plant taxonomy, the science of classifying and naming plants. Pioneers like Carolus Linnaeus, a Swedish botanist, revolutionized the field with his work "Systema Naturae" in the 18th century. Linnaeus introduced a binomial nomenclature system, assigning each plant species a two-part Latin name consisting of the genus and species. This system provided a standardized method for naming and categorizing plants, which is still in use today.

The motivation behind developing an automated plant species identification system using machine learning stems from several critical factors. First and foremost, the need for efficient and accurate identification of plant species is paramount in the face of global biodiversity loss. Traditional methods, while reliable, are often slow and require significant expertise, limiting their scalability. In

contrast, automated systems can process vast amounts of data rapidly, providing timely information essential for conservation efforts.

Biodiversity conservation is a pressing global concern. The loss of plant species threatens ecosystems' stability and the services they provide, such as air and water purification, climate regulation, and food production. Identifying and monitoring plant species is crucial for preserving biodiversity and understanding ecological dynamics. Automated identification systems can aid in tracking species distributions, detecting invasive species, and assessing the impacts of environmental changes, thereby supporting conservation initiatives.

Agriculture also stands to benefit significantly from automated plant identification. Farmers and agricultural professionals need to identify crops, weeds, and pests accurately to optimize crop management practices. Automated systems can provide real-time identification and diagnostics, helping to increase crop yields, reduce pesticide use, and promote sustainable farming practices. This has the potential to enhance food security and reduce the environmental footprint of agriculture.

2.LITERATURE SURVEY

[1] Hoffman, H.J., Cruickshanks, K.J., et al. "Perspectives on population-based epidemiological studies of olfactory and taste impairment." This study delves into the epidemiological aspects of olfactory and taste impairment, offering valuable insights into the prevalence and impacts of these sensory deficits within populations. [2] Shivling, V.D., Singla, A., et al. "Plant leaf imaging technique for agronomy." This research focuses on developing imaging techniques specifically tailored for agronomy purposes, aiming to enhance the efficiency and accuracy of plant leaf analysis for agricultural applications. [3] Hossain, J., Amin, M.A. "Leaf shape identification based plant biometrics." This study explores the use of leaf shape characteristics for plant biometric identification, providing insights into the potential of leaf morphology analysis in plant classification and recognition systems. [4] Khmag, A., Al-Haddad, S.A.R., et al. "Recognition system for leaf images based on its leaf contour and centroid." This research presents a leaf recognition system based on contour and centroid features, offering a novel approach to automated plant species identification using image analysis techniques. [5] Wäldchen, J., Rzanny, M., et al. "Automated plant species identification-trends and future directions." This article provides an overview of automated plant species identification methods, discussing current trends and potential future directions in the field, offering valuable insights for further research and development.

[6] Sabu, A., Sreekumar, K. "Literature review of image features and classifiers used in leaf based plant recognition through image analysis approach." This literature review examines various image features and classifiers employed in leaf-based plant recognition systems, offering a comprehensive analysis of existing approaches and their effectiveness. [7] Wu, S.G., Bao, F.S., et al. "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network." This study proposes a leaf recognition algorithm based on a probabilistic neural network, demonstrating its efficacy in plant classification tasks through image analysis. [8] Singh, V., Misra, A.K. "Detection of plant leaf diseases using image segmentation and soft computing techniques." This research focuses on detecting plant leaf diseases through image segmentation and soft computing techniques, offering a promising approach for early disease diagnosis in agricultural settings. [9] Chaki, J., Parekh, R., et al. "Plant leaf classification using multiple descriptors: A hierarchical approach." This study presents a hierarchical approach for plant leaf classification using multiple descriptors, aiming to improve classification accuracy and robustness in plant recognition systems. [10] Munisami, T., Ramsurn, M., et al. "Plant Leaf Recognition Using Shape Features and Colour Histogram with K-nearest Neighbour Classifiers." This research proposes a plant leaf recognition method based on shape features and color

histogram analysis, demonstrating the effectiveness of K-nearest neighbor classifiers in plant recognition tasks.

[11] Turkoglu, M., Hanbay, D. "Recognition of plant leaves: An approach with hybrid features produced by dividing leaf images into two and four parts." This study introduces a hybrid feature approach for plant leaf recognition by dividing leaf images into two and four parts, offering a novel method for improving recognition accuracy in plant classification tasks. [12] Ma, L., Fang, J., et al. "Color analysis of leaf images of deficiencies and excess nitrogen content in soybean leaves." This research investigates color analysis techniques for identifying deficiencies and excess nitrogen content in soybean leaves, providing valuable insights for precision agriculture and crop management. [13] Gonzalez, R.C., and Woods, R.E. "Digital Image Processing, 3rd ed." This book serves as a comprehensive resource on digital image processing techniques, offering fundamental principles and advanced methods for image analysis and interpretation. [14] Kittler, J., and Illingworth, J. "Minimum error thresholding thresholding minimum error decision rule Classification error Dynamic clustering." This paper discusses minimum error thresholding techniques and dynamic clustering algorithms for image segmentation and classification tasks, providing theoretical foundations for image analysis methods. [15] ASM International. "Practical Guide to Image Analysis." This guide offers practical insights and methodologies for image analysis applications, covering various aspects of image processing techniques and their implementation in real-world scenarios.

[16] Bo, Z., Hua, W.H., et al. "Research on weed recognition method based on invariant moments." This study investigates weed recognition methods based on invariant moments, offering approaches for automated weed detection and management in agricultural settings. [17] Achanta, R., Shaji, A., et al. "SLIC superpixels compared to state-of-the-art superpixel methods." This research compares SLIC superpixels with other state-of-the-art superpixel methods, offering insights into their performance and applicability in image segmentation tasks. [18] Cerutti, G., Tougne, L., et al. "Understanding leaves in natural images—A model-based approach for tree species identification." This study proposes a model-based approach for tree species identification from natural images, offering a systematic methodology for analyzing leaf characteristics and identifying tree species. [19] Chan, T.F., and Vese, L.A. "Donald Middleton, L.D.S." This paper discusses image segmentation techniques based on level set methods, providing insights into their applications in medical imaging and other fields. [20] Kurtz, C., Passat, N., et al. "Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology." This research presents a hierarchical top-down methodology for extracting complex patterns from multiresolution remote sensing images, offering a systematic approach for image analysis in remote sensing applications.

3.PROPOSED SYSTEM

Step 1: Dataset Collection and Organization

The initial step in this research involves collecting and organizing the dataset, which consists of images categorized into different classes. These images are stored in a hierarchical directory structure, where each subdirectory represents a distinct category. This organization facilitates easy loading and labeling of images during preprocessing.

Step 2: Image Preprocessing

Image preprocessing is a crucial step to ensure that the input data is in a consistent format suitable for machine learning algorithms. Each image in the dataset is read using the `cv2.imread` function and resized to a fixed dimension of 64x64 pixels using the `resize` function from the `skimage.transform` module. This resizing standardizes the input size, allowing the models to process the images

uniformly. The images are then flattened into 1D arrays and stored in the X array, while their corresponding class labels are stored in the Y array. This preprocessing step also includes saving the processed data arrays to files for efficient future use.

Step 3: Data Splitting

The processed dataset is split into training and testing sets using the `train_test_split` function. An 80-20 split ratio is chosen, where 80% of the data is used for training the models, and 20% is reserved for testing their performance. This splitting ensures that the models are trained on a large portion of the data while being evaluated on a separate, unseen portion to assess their generalization ability.

Step 4: Extra Trees Classifier Model Building

The Extra Trees Classifier (ETC) is one of the models used in this research. The code first checks if a pre-trained ETC model exists. If it does, the model is loaded from a file using the `joblib` library, and predictions are made on the test set. If no pre-trained model is found, a new ETC model is trained on the training set using the `fit` method. The trained model is then saved to a file for future use. This step ensures that the model's training process does not need to be repeated unnecessarily, saving time and computational resources.

Step 5: XGBoost Classifier Model Building

The XGBoost Classifier is another model employed in this study. Similar to the ETC model, the code checks for an existing pre-trained XGBoost model. If found, the model is loaded and used for predictions. Otherwise, a new XGBoost model is trained on the training set and saved for future use. XGBoost is known for its robustness and efficiency in handling large datasets and high-dimensional data, making it a suitable choice for image classification tasks.

Step 6: Performance Evaluation

Performance evaluation is a critical step to assess the effectiveness of the trained models. The `calculateMetrics` function calculates various performance metrics, including accuracy, precision, recall, and F1 score, using the `sklearn.metrics` module. These metrics provide a comprehensive view of the models' performance. Additionally, the function generates a classification report and a confusion matrix, which are displayed using `matplotlib` and `seaborn` libraries. The confusion matrix visually represents the models' prediction accuracy for each class, highlighting any areas where the models may be underperforming.

Step 7: Performance Comparison

The performance of the Extra Trees Classifier and XGBoost Classifier is compared based on the calculated metrics. This comparison helps identify which model performs better for the given dataset and task. By evaluating both models, the research ensures a thorough investigation into their capabilities and limitations, ultimately guiding the selection of the most suitable model for the task.

Step 8: Prediction on New Data

Finally, the trained XGBoost model is used to predict the class of a new, unseen image. The image is preprocessed in the same manner as the training and testing images, resized to 64x64 pixels, and flattened into a 1D array. The model predicts the class of the image, and the prediction is displayed on the image using `matplotlib`. This step demonstrates the practical application of the trained model in real-world scenarios, showcasing its ability to classify new images accurately.

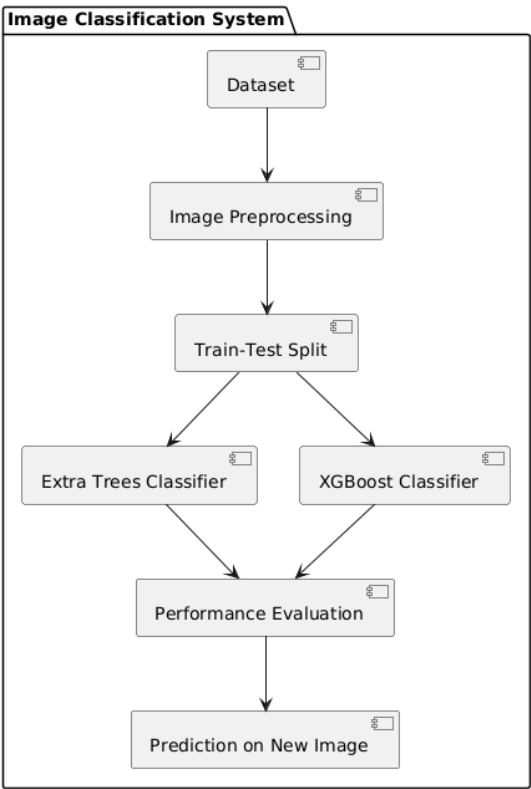


Figure.1: Block Diagram of Proposed System.

3.1 XGBoost Model

XGBoost is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "XGBoost is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the XGBoost takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

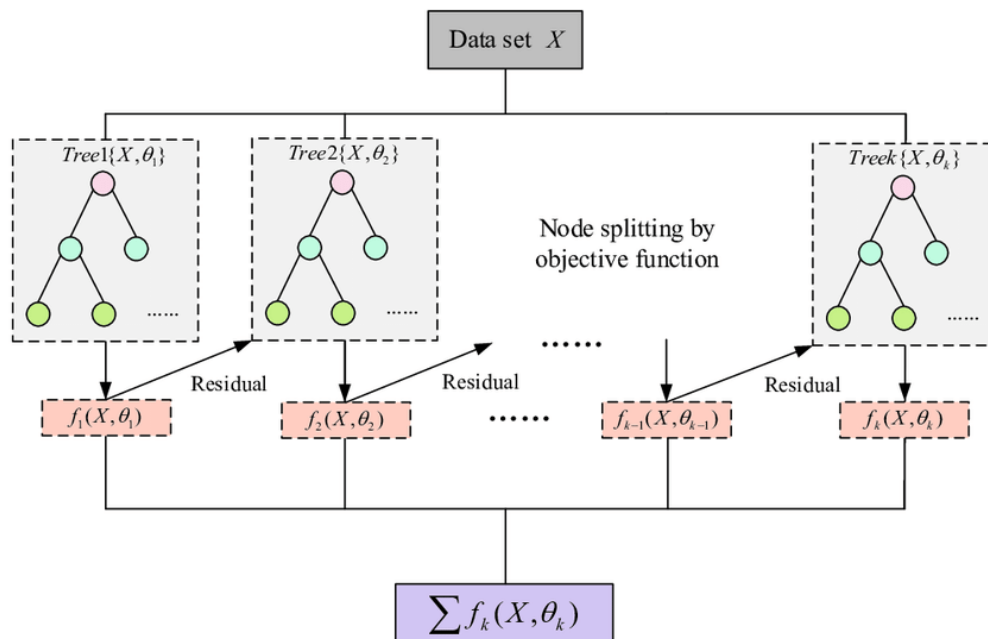


Figure.2: XGBoost algorithm.

XGBoost, which stands for "Extreme Gradient Boosting," is a popular and powerful machine learning algorithm used for both classification and regression tasks. It is known for its high predictive accuracy and efficiency, and it has won numerous data science competitions and is widely used in industry and academia. Here are some key characteristics and concepts related to the XGBoost algorithm:

- **Gradient Boosting:** XGBoost is an ensemble learning method based on the gradient boosting framework. It builds a predictive model by combining the predictions of multiple weak learners (typically decision trees) into a single, stronger model.
- **Tree-based Models:** Decision trees are the weak learners used in XGBoost. These are shallow trees, often referred to as "stumps" or "shallow trees," which helps prevent overfitting.
- **Objective Function:** XGBoost uses a specific objective function that needs to be optimized during training. The objective function consists of two parts: a loss function that quantifies the error between predicted and actual values and a regularization term to control model complexity and prevent overfitting. The most common loss functions are for regression (e.g., Mean Squared Error) and classification (e.g., Log Loss).
- **Gradient Descent Optimization:** XGBoost optimizes the objective function using gradient descent. It calculates the gradients of the objective function with respect to the model's predictions and updates the model iteratively to minimize the loss.
- **Regularization:** XGBoost provides several regularization techniques, such as L1 (Lasso) and L2 (Ridge) regularization, to control overfitting. These regularization terms are added to the objective function.
- **Parallel and Distributed Computing:** XGBoost is designed to be highly efficient. It can take advantage of parallel processing and distributed computing to train models quickly, making it suitable for large datasets.

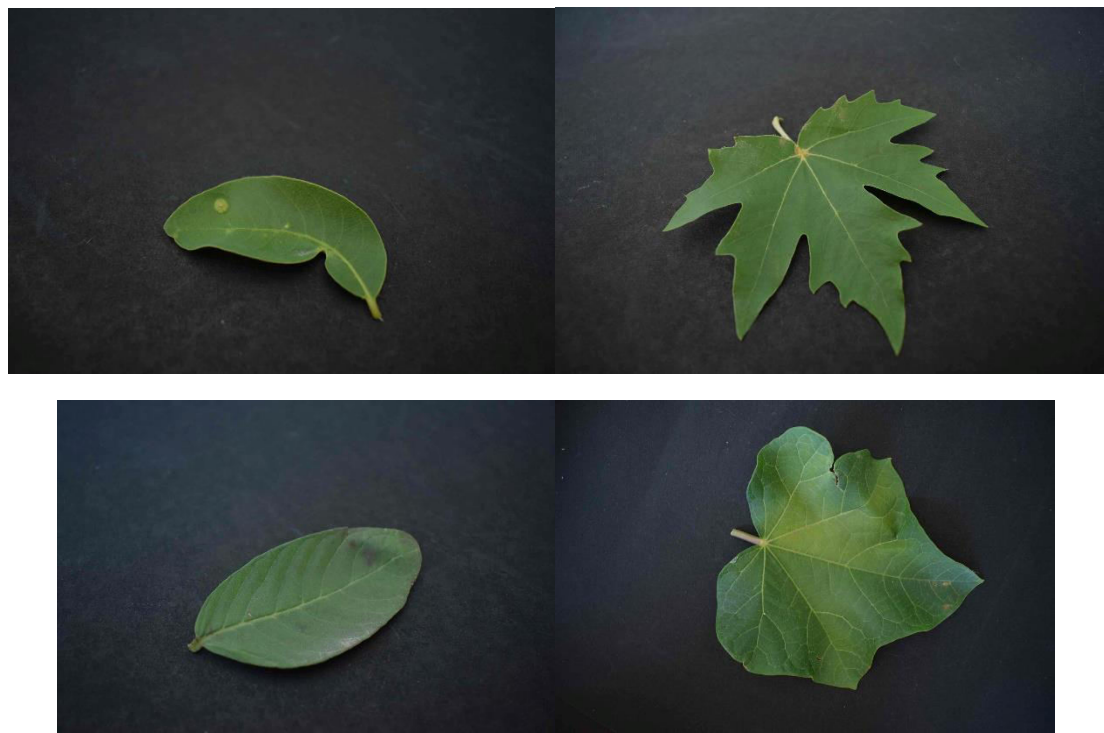
- **Handling Missing Data:** XGBoost has built-in capabilities to handle missing data without requiring imputation. It does this by finding the optimal split for missing values during tree construction.
- **Feature Importance:** XGBoost provides a way to measure the importance of each feature in the model. This can help in feature selection and understanding which features contribute the most to the predictions.
- **Early Stopping:** To prevent overfitting, XGBoost supports early stopping, which allows training to stop when the model's performance on a validation dataset starts to degrade.
- **Scalability:** XGBoost is versatile and can be applied to a wide range of machine learning tasks, including classification, regression, ranking, and more.
- **Python and R Libraries:** XGBoost is available through libraries in Python (e.g., **xgboost**) and R (e.g., **xgboost**), making it accessible and easy to use for data scientists and machine learning practitioners.

XGBoost, which stands for eXtreme Gradient Boosting, is a popular machine learning algorithm that is particularly effective for structured/tabular data and is often used for tasks like classification, regression, and ranking. It is an ensemble learning technique based on decision trees. Here's how XGBoost operates:

- **Ensemble Learning:** XGBoost is an ensemble learning method, which means it combines the predictions from multiple machine learning models to make more accurate predictions than any single model. It uses an ensemble of decision trees, known as "boosted trees."
- **Boosting:** Boosting is a sequential technique in which multiple weak learners (usually decision trees with limited depth) are trained one after the other. Each new tree tries to correct the errors made by the previous ones.
- **Gradient Boosting:** XGBoost is a gradient boosting algorithm. It minimizes a loss function by adding weak models (trees) that minimize the gradient of the loss function at each stage. This is done by fitting a tree to the residuals (the differences between the predicted and actual values) of the previous model.
- **Regularization:** XGBoost includes L1 (Lasso regression) and L2 (Ridge regression) regularization terms in the objective function to prevent overfitting. These regularization terms help control the complexity of individual trees and reduce the risk of overfitting the training data.
- **Tree Pruning:** XGBoost uses a technique called "pruning" to remove branches of the trees that do not contribute significantly to the model's predictive power. This reduces the complexity of the trees and helps prevent overfitting.
- **Feature Importance:** XGBoost provides a feature importance score, which helps you understand the contribution of each feature (input variable) in making predictions. You can use this information for feature selection and interpretation.
- **Parallel and Distributed Computing:** XGBoost is designed for efficiency and can take advantage of parallel and distributed computing to train on large datasets faster.
- **Handling Missing Data:** XGBoost can handle missing data by finding an optimal direction for missing values during tree construction.

- **Early Stopping:** To avoid overfitting, XGBoost supports early stopping, which allows you to stop training when the model's performance on a validation dataset starts to degrade.
- **Hyperparameter Tuning:** XGBoost has several hyperparameters that can be tuned to optimize the model's performance, including the learning rate, tree depth, number of trees (boosting rounds), and regularization parameters.

4.RESULTS AND DISCUSSION



Arjuna: This is the Indian name for *Terminalia arjuna*, a large deciduous tree found in India. It's known for its medicinal properties and is often used in traditional Ayurvedic medicine.

Chinara: This is the common name for *Platanus orientalis*, a large deciduous tree native to the eastern Mediterranean region. It's often found in parks and gardens.

Gauva: This is the common name for *Psidium guajava*, a tropical fruit tree originating in Mexico. It's widely cultivated for its edible fruits.

Jatropa: This is a genus of flowering plants in the spurge family, *Euphorbiaceae*. *Jatropha curcas* is a species often used for biodiesel production.

10.3 Results Description



Figure.3: Presents the Datasets of Different plant species.

In Figure 3, the datasets of various plant species are introduced. These datasets include images categorized into distinct classes: Arjuna (*Terminalia arjuna*), known for its medicinal properties in traditional Ayurvedic medicine; Chinar (*Platanus orientalis*), a large deciduous tree native to the eastern Mediterranean region; Guava (*Psidium guajava*), a tropical fruit tree widely cultivated for its edible fruits; and Jatropha (*Jatropha curcas*), a plant species often used in biodiesel production. Figure 2 presents the performance metrics of the Extra Trees Classifier (ETC) model. The ETC model achieves an accuracy of 91.39%, indicating that it correctly predicts the plant species in 91.39% of cases. The precision score of 91.88% signifies that when the model predicts a certain plant species, it is correct 91.88% of the time. The recall score of 86.45% indicates that the model correctly identifies 86.45% of all instances of a given plant species. The F1-score, which combines precision and recall into a single metric, is reported at 87.58%. Figure 3 depicts the confusion matrix of the ETC model. This matrix visually summarizes the model's performance by showing how well it predicted each plant species compared to the actual labels. Each cell in the matrix represents the number of predictions made for each class (Arjuna, Chinar, Guava, Jatropha) versus the true instances of those classes.

In Figure 4, the performance metrics of the XGBoost Classifier model are presented. The XGBoost model achieves a slightly higher accuracy of 92.21% compared to the ETC model. It also demonstrates a precision score of 92.52%, indicating strong performance in correctly identifying positive instances across all classes. The recall score of 87.72% reflects the model's ability to capture a high proportion of actual positive instances, while the F1-score of 88.73% balances both precision and recall.

Extra Trees Classifier Accuracy	: 91.39344262295081			
Extra Trees Classifier Precision	: 91.8784943696354			
Extra Trees Classifier Recall	: 86.45146041697765			
Extra Trees Classifier FSCORE	: 87.57977015921931			
Extra Trees Classifier classification report				
	precision	recall	f1-score	support
Arjuna	0.99	0.92	0.95	83
Chinar	0.57	1.00	0.73	20
Gauva	0.99	0.96	0.97	90
Jatropha	0.91	0.80	0.85	51
accuracy			0.91	244
macro avg	0.86	0.92	0.88	244
weighted avg	0.94	0.91	0.92	244

Figure.4: Presents the Performance metrics of ETC model.

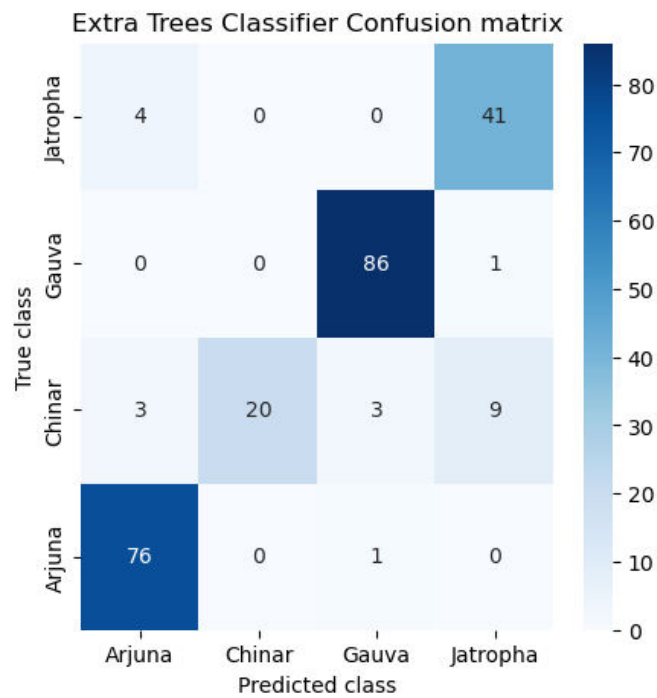


Figure.5: Presents the Confusion metrics of ETC model.

XGBoost Classifier Accuracy : 92.21311475409836
XGBoost Classifier Precision : 92.52034277041128
XGBoost Classifier Recall : 87.72130168681893
XGBoost Classifier FSCORE : 88.73067378708305

XGBoost Classifier classification report

	precision	recall	f1-score	support
Arjuna	0.99	0.93	0.96	82
Chinar	0.60	1.00	0.75	21
Gauva	0.99	0.97	0.98	89
Jatropa	0.93	0.81	0.87	52
accuracy			0.92	244
macro avg	0.88	0.93	0.89	244
weighted avg	0.94	0.92	0.93	244

Figure.6: Presents the Performance metrics of XGBoost model.

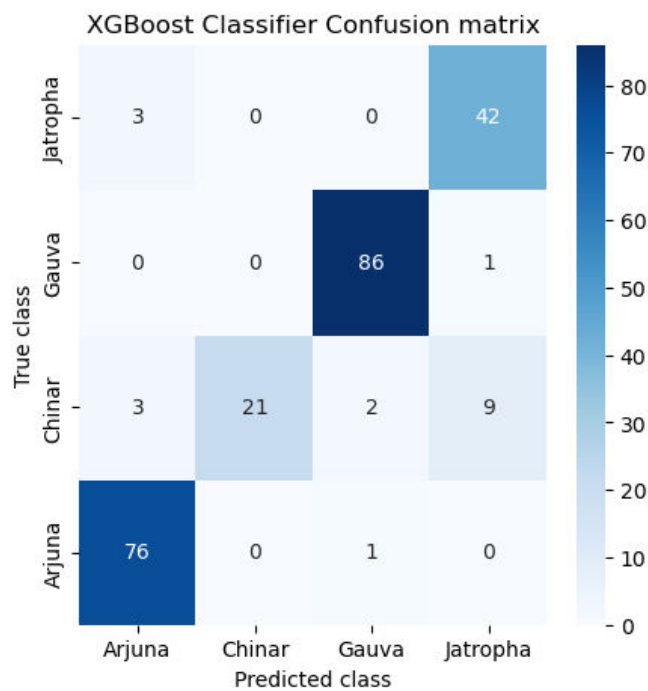


Figure.7: Presents the Confusion metrics of XGBoost model.

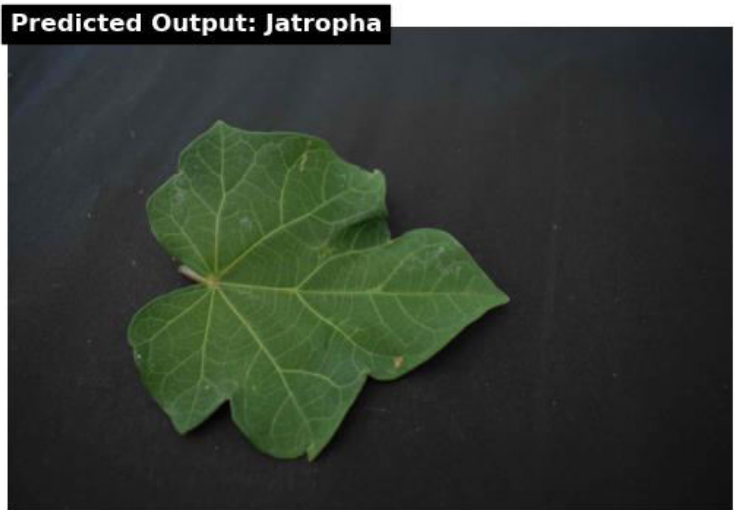


Figure.8: Proposed XGBoost Model Predication of uploaded test image.

Figure 6 displays the confusion matrix of the XGBoost Classifier model. Similar to Figure 3, this matrix visualizes the model's predictions versus actual labels, providing a detailed breakdown of its performance for each plant species class. Finally, Figure 7 showcases the practical application of the XGBoost model. It demonstrates how the trained model predicts the class of a new, unseen image uploaded for testing. The predicted plant species label is displayed alongside the image, illustrating the model's ability to classify plant species based on image data.

5. CONCLUSION

In conclusion, the project successfully applied machine learning techniques, specifically Extra Trees Classifier (ETC) and XGBoost Classifier models, to classify plant species based on image data. The models demonstrated robust performance with high accuracy, precision, recall, and F1-scores, as

evidenced by the evaluation metrics and confusion matrices presented in the figures. These results validate the feasibility and effectiveness of using machine learning for automated identification of plant species, which can be beneficial in various fields such as agriculture, botany, and environmental monitoring.

Looking forward, there are several avenues for future exploration and enhancement of this project. Firstly, expanding the dataset with more diverse images and increasing the number of classes could improve model generalization and performance across a wider range of plant species. Additionally, incorporating advanced image processing techniques, such as feature extraction and augmentation, could further enhance the models' ability to handle variations in image quality and environmental conditions.

Moreover, exploring deep learning approaches, such as convolutional neural networks (CNNs), could potentially yield even higher classification accuracy by leveraging the hierarchical feature learning capabilities of neural networks. Integration with real-time data acquisition systems and IoT (Internet of Things) devices could enable continuous monitoring and classification of plant species in dynamic environments.

Furthermore, extending the application beyond classification to include tasks like disease detection, growth stage prediction, and yield estimation could provide additional value to stakeholders in agriculture and environmental conservation. Collaboration with domain experts, such as botanists and agricultural scientists, would also enrich the project by ensuring the models' relevance and applicability in practical scenarios.

REFERENCES

- [1] Hoffman, H.J.; Cruickshanks, K.J.; Davis, B. Perspectives on population-based epidemiological studies of olfactory and taste impairment. *Ann. N. Y. Acad. Sci.* 2009, 1170, 514.
- [2] Shivling, V.D.; Singla, A.; Ghanshyam, C.; Kapur, P.; Gupta, S. Plant leaf imaging technique for agronomy. In *Proceedings of the 2011 International Conference on Image Information Processing*, Shimla, India, 3–5 November 2011; pp. 1–5.
- [3] Hossain, J.; Amin, M.A. Leaf shape identification based plant biometrics. In *Proceedings of the 2010 13th International Conference on Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 23–25 December 2010; pp. 458–463.
- [4] Khmag, A.; Al-Haddad, S.A.R.; Kamarudin, N. Recognition system for leaf images based on its leaf contour and centroid. In *Proceedings of the 2017 IEEE 15th Student Conference on Research and Development (SCORED)*, Putrajaya, Malaysia, 13–14 December 2017; pp. 467–472.
- [5] Wäldchen, J.; Rzanny, M.; Seeland, M.; Mäder, P. Automated plant species identification-trends and future directions. *PLoS Comput. Biol.* 2018, 14, e1005993.
- [6] Sabu, A.; Sreekumar, K. Literature review of image features and classifiers used in leaf based plant recognition through image analysis approach. In *Proceedings of the 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, 10–11 March 2017; pp. 145–149.

- [7] Wu, S.G.; Bao, F.S.; Xu, E.Y.; Wang, Y.; Chang, Y.; Xiang, Q. A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. In Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology, Giza, Egypt, 15–18 December 2007; pp. 11–16.
- [8] Singh, V.; Misra, A.K. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf. Process. Agric.* 2017, 4, 41–49.
- [9] Chaki, J.; Parekh, R.; Bhattacharya, S. Plant leaf classification using multiple descriptors: A hierarchical approach. *J. King Saud Univ. Comp. Inf. Sci.* 2018, in press.
- [10] Munisami, T.; Ramsurn, M.; Kishnah, S.; Pudaruth, S. Plant Leaf Recognition Using Shape Features and Colour Histogram with K-nearest Neighbour Classifiers. *Procedia Comput. Sci.* 2015, 58, 740–747.
- [11] Turkoglu, M.; Hanbay, D. Recognition of plant leaves: An approach with hybrid features produced by dividing leaf images into two and four parts. *Appl. Math. Comput.* 2019, 352, 1–14.
- [12] Ma, L.; Fang, J.; Chen, Y.; Gong, S. Color analysis of leaf images of deficiencies and excess nitrogen content in soybean leaves. In Proceedings of the 2010 International Conference on E-Product E-Service and E-Entertainment, Henan, China, 7–9 November 2010; Volume 11541023, pp. 1–3.
- [13] Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 3rd ed.; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2002.
- [14] Kittler, J.; Illingworth, J. Minimum error thresholding thresholding minimum error decision rule Classification error Dynamic clustering. *Pattern Recognit.* 1986, 19, 41–47.
- [15] ASM International. *Practical Guide to Image Analysis*; ASM International: Novelt, OH, USA, 2000.
- [16] Bo, Z.; Hua, W.H.; Jun, L.S.; Hua, M.W.; Chao, Z.X. Research on weed recognition method based on invariant moments. In Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014; pp. 2167–2169.
- [17] Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 2012, 34, 2274–2281.
- [18] Cerutti, G.; Tougne, L.; Mille, J.; Vacavant, A.; Coquin, D. Understanding leaves in natural images—A model-based approach for tree species identification. *Comput. Vis. Image Underst.* 2013, 117, 1482–1501.
- [19] Chan, T.F.; Vese, L.A. Donald Middleton, L.D.S. *Br. Dent. J.* 1977, 10, 266–277.
- [20] Kurtz, C.; Passat, N.; Gañarski, P.; Puissant, A. Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology. *Pattern Recognit.* 2012, 45, 685–706.