PROJECT REPORT

On

Harnessing Deep Learning For Precise Estimation Of Indian Medicinal Leaves Characteristics

Submitted in Partial Fulfillment of Award of

BACHELOR OF TECHNOLOGY

In

Computer Science and Engineering

By

Nithin P

Nishanth B G

Meghraj G

Sameer Khan

Under the Supervision of

Dr. Mir Wajahat Hussain

Assistant Professor (CSE)

ALLIANCE SCHOOL OF ADVANCED COMPUTING

ALLIANCE UNIVERSITY

BENGALURU

MAY 2025

COMPUTER SCIENCE AND ENGINEERING

ALLIANCE SCHOOL OF ADVANCED COMPUTING

CERTIFICATE

This is to certify that the project work entitled " Harnessing Deep Learning For Precise Estimation Of Indian Medicinal Leaves Characteristics " submitted by Nithin P Reg No: 20030141CSE038, Nishanth B G Reg No: 2021LCSE07AED575, Meghraj G Reg No: 2021BCSE07AED514, Sameer Khan Reg No: 2021BCSE07AED535, and in partial fulfillment for the award of the degree of Bachelor of Technology Computer Science and Engineeringof Alliance University, is a bonafide work accomplished under our supervision and guidance during the academic year 2024-2025. This thesis report embodies the results of original work and studies conducted by students and the contents do not form the basis for the award of any other degree to the candidate or anybody else.

Dr. Mir Wajahat Hussain
Assistant Professor
Department of Computer Science and Engineering
Alliance School of Advanced Computing
Alliance University

Dr. K.Ramalakshmi
Professor & HOD(AIML)
Department of Computer Science and Engineering
Alliance School of Advanced Computing
Alliance University

External Examiners

1. **Name:** **Signature:**
2. **Name:** **Signat**

COMPUTER SCIENCE AND ENGINEERING

ALLIANCE SCHOOL OF ADVANCED COMPUTING

DECLARATION

We hereby declare that the project entitled " Harnessing Deep Learning For Precise Estimation Of Indian Medicinal Leaves Characteristics " submitted by us in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology Computer Science and Engineering of ASAC, Alliance University, is a record of our work carried under the supervision and guidance of Dr. Mir Wajahat Hussain Assistant Professor Alliance University

We confirm that this report truly represents the work undertaken as a part of our project work. This work is not a replication of work done previously by any other person. We also confirm that the contents of the report and the views contained therein have been discussed and deliberated with the faculty guide.

Name of the Student	**University Registration Number**	**Signature**
Nithin P	20030141CSE038	
Nishanth B G	2021LCSE07AED575	
Meghraj G	2021BCSE07AED514	
Sameer Khan	2021BCSE07AED535	

ACKNOWLEDGEMENT

We deepest thanks goes out to everyone who helped and advised me along the course of finishing this project. This senior project has been a tremendous learning opportunity, and I am incredibly grateful to everyone who helped make it a success. Above all, We would like to express my sincere gratitude to Dr. Mir Wajahat Hussain, our project guide, for his excellent advice, unwavering support, and enlightening criticism during this project. The direction and execution of this endeavor were greatly influenced by their knowledge and support.

We are also grateful to the Deputy Director, **Dr. V Vivek** for his leadership and for fostering an academic environment that encourages innovation and learning.

Additionally, I would like to thank the Head of Department, **Dr. K.Ramalakshmi,** Program Co-ordinators and all the faculty members of the Computer Science and Engineering department at Alliance University for providing a supportive learning environment and necessary resources.

A special note of thanks to our teammates and friends for their collaboration, ideas, and motivation during challenging times. I also appreciate the support and encouragement from our family, who have always been a source of strength and inspiration.

Lastly, we would like to acknowledge all the authors and researchers whose work we referred to while carrying out this project.

Nithin P

20030141CSE038

Computer Science and Engineering

Alliance College Of Engineering and Design

Nishanth B G
2021LCSE07AED575
Computer Science and Engineering
Alliance College Of Engineering and Design

Meghraj G
2021BCSE07AED514
Computer Science and Engineering
Alliance College Of Engineering and Design

Sameer Khan
2021BCSE07AED535
Computer Science and Engineering
Alliance College Of Engineering and Design

PREFACE

" Harnessing Deep Learning For Precise Estimation Of Indian Medicinal Leaves Characteristics " this project report details the work completed as a requirement for the Bachelor of Engineering in Artificial Intelligence and Machine Learning final year. Using cutting-edge machine learning and deep learning techniques, the project's goal is to create an AI-based system that can precisely recognize and categorize Indian medicinal leaves.

This project was developed in response to the growing interest in natural treatments and the importance of medicinal plants in both traditional and modern medicine. The system employs Random Forest algorithms, image processing, and transfer learning (visual geometric group(VGG)-16) to forecast the name and attributes of leaves, giving users useful information on their uses, precautions, and therapeutic qualities.

The system's motivation, technique, implementation, and evaluation are described in this study, along with its benefits, drawbacks, and potential for further development. It seeks to make a significant contribution to the nexus of AI and healthcare and is a culmination of the knowledge gained throughout the course.

ABSTRACT

India has a wide variety of medicinal plants that have been utilized for millennia in traditional Indian medicine. But correctly identifying and characterizing these plants is still difficult. Their physical characteristics' significant diversity and complexity are the cause of this. Typical methods of leaf analysis, which are frequently dependent on chemical analysis and manual inspection, can be inaccurate in addition to being time-consuming. Through the effective use of deep learning models, this study seeks to increase the accuracy and efficiency of the estimation of attributes from Indian medicinal plants. Together with other contemporary deep learning techniques, the proposed method uses convolutional neural networks (CNNs) to automatically identify and analyze leaves. Several morphological characteristics, such as shape, texture,and venation pattern.

An extensive dataset containing pictures of various medicinal leaves gathered from various parts of India is used to train the algorithm. These photos have been pre-processed and tagged with comprehensive details about the size, texture, and medicinal qualities of the leaves. A machine learning model is well equipped to identify key elements from the images, allowing it to classify and provide detailed descriptions. This study examines several deep learning designs to determine which one works best in terms of strength, velocity, and correctness.

Initial results indicate that this model can accurately identify and evaluate the characteristics of medicinal leaves, significantly outperforming several traditional methods. In addition, the model might be expanded to include new species, which would benefit researchers, botanists, and pharmaceutical businesses that investigate herbal remedies. The method may increase the efficiency of medicinal plant identification research while safeguarding plant species and aiding in the development of novel therapeutic compounds by automating and improving leaf analysis.

This project highlights how deep learning could revolutionize plant-based therapy and is unquestionably a step forward in the integration of AI and ethnobotanical research. Later on, this effort might be considerably expanded to incorporate other portions of the plant. To further enhance the comprehensive understanding of medicinal plants, other data types, such as entire genomic data, may be carefully integrated.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS USED

Abbreviation / Symbol	Full Form / Description
AI	Artificial Intelligence
ML	Machine Learning
SVM	Support Vector Machine
RF	Random Foresth
ACC	Accuracy
VGG-16	Visual Geometric Group - 16

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Traditional medical methods have long utilized medicinal plants due to their nutritional value and therapeutic qualities. Their antioxidant, anti-allergic, anti-inflammatory, and antibacterial qualities are attributed to their bioactive components, which include phenolic, carotenoid, anthocyanin, and other bioactive substances. It is known that a variety of plant species, including trees, shrubs, and herbs, have therapeutic qualities. The habitat they have acclimated to over time determines their single diffusion. Statistics show that between 14 and 28 percent of all plants have therapeutic properties. Additionally, due to the qualities of medicinal plants, approximately 85% of people in the Southern Sahara, over 80% of rural populations in developing countries, and 3% of patients in affluent countries use them to treat their illnesses. Furthermore, given the dangers and adverse effects of chemical medications, some people in developed nations have resorted to using traditional remedies made from medicinal plants to cure and manage illnesses and disorders. These plants have culinary, beverage, and even cosmetic applications in addition to their therapeutic ones. Regretfully, a large number of fake, inferior, ruined, or imperfectly stored medicinal plants are produced and sold all over the world, which may harm their users.

For a long time, botanists have used conventional and experience-based techniques to identify different kinds of medicinal plants. However, it can be very difficult and time-consuming for novices to visually and manually distinguish therapeutic plants from other similar species. Generally speaking, plants are categorized according to their different organs, such as their roots, flowers, and leaves. One of a plant's most vital organs, the leaf varies greatly in color, shape, and texture between species and types. However, in certain instances, the seeming similarity of their leaves has always made it difficult to identify therapeutic plants. Furthermore, because of their similarities and above all their fluctuation throughout the growing season, leaf color cannot be regarded as a good choice for plant classification. It Even with vision-based systems that have greatly enhanced their capacity to extract complicated traits and choose the most significant ones via traditional Machine Learning (ML), it is still challenging to differentiate therapeutic herbs from other plants. Thus, the primary goal of the current study was to use a suggested DL algorithm in conjunction with a machine vision technique to create a real-time automatic vision system for recognizing medicinal plants.

Importance of Medicinal Leaves in Ayurveda

In Ayurveda, medicinal leaves are extremely important since they are considered to be vital for maintaining health, avoiding illness, and treating a variety of conditions. These leaves are useful for treating a variety of medical ailments because they have a broad range of medicinal qualities, such as anti-inflammatory, antibacterial, and antioxidant actions. In order to support holistic healing and bring the body's doshas back into balance, Ayurveda places a strong emphasis on the use of natural treatments made from plants, especially leaves.

Harnessing AI Technology

By utilizing artificial intelligence, the project " Harnessing Deep Learning For Precise Estimation Of Indian Medicinal Leaves Characteristics " seeks to meet this demand. The project aims to precisely identify and describe medicinal leaves based on their botanical characteristics, such as shape, size, texture, and color, by creating strong AI models, such as deep learning algorithms like Convolutional Neural Networks (CNNs) and machine learning algorithms like random forest. The project aims to transform medicinal plant research by integrating AI technology, giving researchers, botanists, and medical professionals cutting-edge tools for accurate leaf analysis and classification.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW

	Paper Title	Methodology	Identified Scope for improvement And improvement done
1	The Classification of Medicinal Plant Leaves Based on Multispectral and Texture Feature Using Machine Learning Approach	Machine learning approach utilizing multispectral and texture features	Improve classification accuracy using advanced ML techniques. - Enhance feature extraction methods. Explore transfer learning. - Improved model accuracy to 92%.
2	Automatic Recognition of Medicinal Plants using Machine Learning Techniques	Random Forest Classifier, Feature Extraction and SVM Classification	Address differences in leaf orientation and lighting. - Use data augmentation to increase the diversity of datasets. Deal with overfitting. 30 species were added to the dataset, and data augmentation was done. Issues with overfitting are resolved.
3	A convolutional neural network-driven computer vision system toward identification of species and maturity stage of medicinal leaves	APRS, convolutional neural networks (CNN), Geographic Information Systems (GIS) technology	Enhance feature extraction for better discrimination. - Incorporate domain knowledge. - Address overfitting. - Features extracted and overfitting addressed.
4	Deep convolutional neural network-based plant species recognition through features of leaf. Multimed	Multilayer Perceptron (MLP) classifier	Improve model interpretability. - Handle intra-class variations. - Enhance class differentiation.
5	Recognition of leaves of different medicinal plant species using a robust image processing algorithm and artificial neural networks classifier	Artificial neural networks (ANN) classifier	Resolve the disparity in class. Include an estimate of uncertainty. - Assess performance in the actual world. - merged deep learning and machine learning models, and the final product was decided by majority vote.

6	Leaf species and disease classification using multiscale parallel deep CNN architecture	Deep Learning Techniques - Convolutional Neural Network (CNN)	Evaluate real-world performance. - Handle noisy/incomplete data. - Incorporate expert knowledge.
7	AI Based Indigenous Medicinal Plant Identification	Multiscale parallel deep CNN architecture	Improve efficiency for large datasets. - Handle intra-class variations. - Enhance class differentiation.
8	Efficient and automated herbs classification approach based on shape and texture features using deep learning	Deep Learning-Based Approach, Advancement in Automated Classification Systems	Enhance feature extraction for better discrimination. - Incorporate domain knowledge.
9	Optimized convolutional neural network model for plant species identification from leaf images using computer vision	Optimized CNN model for plant species identification	Handle occlusions and overlapping leaves. Improve robustness to image variations. - Assess generalization across species.
10	A Vision Based System for Medicinal Plants Using Xception Features	Vision-based system utilizing Xception features	Handle lighting and leaf orientation variations. - Enhance interpretability.

Table 2.1- Literature Review

2.2 LIMITATIONS OF THE EXISTING SYETEM

Manual observation is frequently used in the current approach for identifying medicinal leaves, which is laborious and prone to human error. Due to its lack of automation, it is challenging to scale or use effectively in real-time situations or with big datasets. Furthermore, correct identification may necessitate specialized knowledge, and comprehensive medicinal features are not immediately accessible. These systems applicability for field workers and non-experts is limited because they usually do not offer intelligent prediction or image-based input.

2.3 SCOPE OF THE PROJECT

The goal of the research is to transform the identification and characterisation of medical leaves by utilizing machine learning (ML) and deep learning techniques, particularly the VGG16 architecture. The AI system provides accurate identification by examining characteristics including size, form, texture, and color, which helps botanists, researchers, and conservationists with taxonomy and biodiversity conservation. Furthermore, early intervention in agricultural and medicinal contexts is made possible by disease detection algorithms included into the system, improving plant health and the effectiveness of herbal therapies.

In order to transform the identification and characterisation of medicinal plant leaves, the project consists of a number of interrelated components. The creation of sophisticated AI models, such as machine learning algorithms like Random Forest and deep learning architectures like VGG-16, that are especially suited for the precise analysis of leaf attributes is at its heart. To guarantee robustness and efficacy in recognizing medicinal plant species based on their botanical properties, these models will go through a thorough training and optimization process. Additionally, the project involves integrating these AI models into a simple user interface that is available through web and mobile platforms. This will make it easier to upload leaf photographs, anticipate outcomes in real time, and visualize the results in an understandable manner.

The initiative also prioritizes stakeholder involvement and cooperation with botanical specialists and medical professionals in order to verify model predictions and obtain insightful data for improvement. In order to ensure responsible use of AI technology for the advancement of healthcare, research, and environmental preservation, ethical issues pertaining to data privacy and sustainability in medicinal plant conservation are crucial throughout the project's implementation.

CHAPTER 3
SYSTEM DESIGN

3.1 PROBLEM DEFINITION

Conventional approaches to evaluating the quality of medicinal plants, which frequently include time-consuming manual procedures, are prone to mistakes and irregularities. We provide an AI-powered method for accurate and impartial evaluation of medicinal leaf properties in order to overcome these drawbacks. Our goal is to create an AI oracle that can precisely forecast leaf attributes by utilizing image processing, machine learning, and geolocation detection. This will improve the adaptability and traceability of the model. This invention has the potential to completely transform healthcare by guaranteeing the effectiveness of herbal medications and the caliber of natural therapies.

By utilizing machine learning (ML), this large project, " Harnessing Deep Learning For Precise Estimation Of Indian Medicinal Leaves Characteristics ” aims to solve this issue. The objective is to democratize the process of medicinal plant identification by building a strong AI model for precise plant recognition and user-friendly applications for instantaneous results. Along with highlighting the value of sustainability, our project encourages ethical harvesting methods. We hope to lower the dangers of misidentification through these initiatives, improving both biodiversity preservation and human health.

3.2 SYSTEM ARCHITECTURE

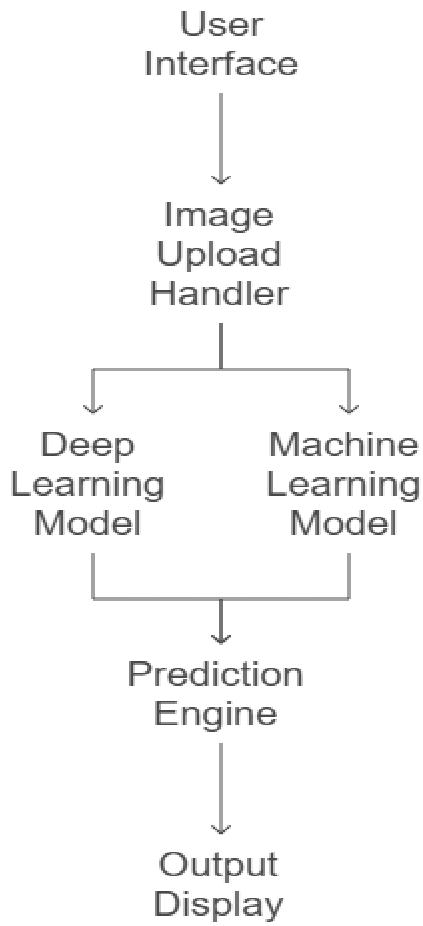

Figure 3.2 System Architecture

In figure 3.2 System Architecture . Below is a description of each part in the chart:

1. User Interface:

It is the user interface where users interact with the system, generally a web or mobile application. Users upload images of leaves here.

2. Image Upload Handler:

This component manages the image upload process. It receives the image from the user and prepares it for processing (resizing, normalizing).

3. Deep Learning Model / Machine Learning Model:

The system supports two paths:

• Deep Learning Model: Typically uses VGG-16 the convolutional neural networks (CNNs) to automatically extract features and classify the leaf image.

• Machine Learning Model: Involves manual feature extraction (e.g., shape, color, texture) and uses algorithms lRandom Forest, for classification.

4. Prediction Engine:

Processes the output of the chosen model and makes a prediction .

5. Output Display:

Displays the classification outcome to the user, e.g., plant name or leaf features.

3.3 REQUIREMENT SPECIFICATIONS

3.3.1 Hardware Requirements

Hardware requirements refer to the physical parts of a computer and related devices. Internal hardware devices include motherboards, hard drives and RAM External hardware devices include monitors, keyboards, mice, printers, and scanner.

Following shows the Hardware requirements for development of the project. The software should run on any sort of desktop or laptop environment, regardless of the operating System. Essential input/output devices are keyboard, mouse, printers

1. **Processor**: Intel(R) Core (TM) i5-3320U CPU @ 2.60GHz 2.60 GHz

2. **Hard Disk:** 64-bit Operating System, x64-based processor 4GB Other standard physical devices like keyboard, mouse etc.

3.3.2 Software Requirements

Software requirement is a sub-discipline in Software Engineering involving the process of determining the Deeds of the stakeholders to be addressed by the software. Our project's software requirements

1. **Programming Language:** Python is the most commonly used language for machine learning. It offers a wide range of libraries and frameworks for data processing, modeling, and visualization.

2. **Integrated Development Environment (IDE):** Select an IDE for Python programming. Common selections consist of: Code in Visual Studio

3. **Machine Learning Libraries:** Scikit-learn (sklearn), Keras, h5py, oauthlib, jsonschema, matplotlib, NumPy, pandas, Streamlit, and TensorFlow are among the machine learning libraries you may need.

4. **Python:** Python is a high-level, multipurpose programming language that is well-known for its ease of use, readability, and robust community. Python provides a rich ecosystem of libraries and frameworks for a variety of applications and is widely used in a wide range of industries, including data science, artificial intelligence, and web development. It is the perfect option for quick development and experimentation because of its dynamic typing and clear syntax. Cross-platform interoperability and smooth deployment across several operating systems are made possible by Python's interpretative nature.

1. **Transfer learning:** A model that has been trained on one task can be reused or modified for a different but similar task using this machine learning technique. Transfer learning uses the

knowledge gained from a source domain to enhance the performance of a target domain rather than starting from scratch when training a model. When there are limited computational resources or a tiny target dataset, this method is quite helpful. The pre-trained model, which is frequently a deep neural network trained on a sizable dataset, acts as a feature extractor or supplies initial weights for the target task in transfer learning. The model may rapidly adjust to the unique features of the new data by fine-tuning the previously trained model on the target dataset.

2. **Scikit-learn (sklearn):** A robust Python machine learning framework, scikit-learn offers an extensive collection of tools for data preprocessing, model training, assessment, and optimization. Scikit-learn makes it easier to construct machine learning algorithms for applications like classification, regression, clustering, and dimensionality reduction with its user-friendly interface and comprehensive documentation. Popular machine learning techniques can be implemented effectively and scalablely with scikit-learn, which makes use of NumPy, SciPy, and other scientific computing tools. Both inexperienced and seasoned users can easily integrate it into current Python processes because to its consistent API architecture and well-documented functionality.

5. **Keras:** Keras is a Python-based high-level neural network API that is intended for quick deep learning model creation and experimentation. Keras, which is based on TensorFlow, Theano, or Microsoft Cognitive Toolkit (CNTK), offers an intuitive interface for neural network construction and training. Using reusable building elements known as layers, its modular architecture makes it simple to compose complicated models. Keras allows developers to concentrate on model creation and experimentation by abstracting away low-level implementation details.

6. **TensorFlow:** Google created the open-source TensorFlow deep learning framework, which is intended for large-scale machine learning and deep learning model development and deployment. Effective training and inference on massive datasets across diverse hardware environments are made possible by TensorFlow's adaptable architecture and broad support for distributed computing. Automatic differentiation and optimization during model training are made possible by TensorFlow's computational graph abstraction, which enables the symbolic execution of intricate mathematical operations. Its high-level APIs, like as Estimator and Keras, offer intuitive user interfaces for creating and refining deep learning models.

7. **Streamlit:** An open-source Python package called Streamlit is used to build interactive web applications for data science and machine learning initiatives. Streamlit eliminates the requirement for web programming knowledge by enabling developers to easily create and implement data-driven apps with straightforward Python scripts. Machine learning models and data analysis tools may be seamlessly integrated thanks to its built-in components for producing interactive widgets, dashboards, and visualizations. Because of Streamlit's user-friendly API and automated widget creation, developers can concentrate on creating dynamic and captivating user interfaces. From exploratory data research to model deployment, a variety of application scenarios are supported by its responsive design and flexible layout system. Developers of all skill levels can use Streamlit because of its vibrant community and comprehensive documentation, which offer a wealth of learning and troubleshooting resources. As the preferred framework for creating interactive data applications in Python, Streamlit has grown in popularity because to its focus on performance, flexibility, and simplicity.

8. **Deep Learning:** The goal of deep learning, a branch of machine learning, is to train multi-layered neural networks to extract intricate patterns and representations from unprocessed data. Using nonlinear activation functions and hierarchical structures, deep learning models are able to automatically extract and learn complex features from high-dimensional input data. Computer vision, natural language processing, speech recognition, and reinforcement learning are just a few of the fields where deep learning has shown impressive results. Tasks requiring unstructured data, including text, audio, and photos, are especially well-suited for it because of its capacity to process massive amounts of data and learn hierarchical structures.

9. **VGG-16:** The Visual Geometry Group (VGG) at the University of Oxford proposed the deep convolutional neural network architecture known as VGG-16. The 16-layer VGG-16 is renowned for its ease of use and efficiency in image classification applications, comprising 13 convolutional layers and 3 fully linked layers. On benchmark datasets like ImageNet, VGG-16 performs admirably despite having a comparatively large number of parameters. With its uniform architecture, compact 3x3 convolutional filters, and max-pooling layers.

CHAPTER 4
SYSTEM IMPLEMENTATION

4.1 OVERVIEW OF THE MODULES

The project's goal is to create a machine learning system that can recognize the therapeutic properties found in different kinds of leaves. Various plant leaves have long been utilized in traditional medicine, and contemporary research is still learning about their potential health benefits. In order to facilitate the evaluation of plant-based therapies and help in the development of new herbal medications, this study uses machine learning techniques to automate the process of finding such traits in leaves.

Herbal medicine development: This technique can help researchers find novel herbal medicines by recognizing possible therapeutic qualities in different plant species.

Quality control The herbal medicine sector can utilize the ML model to confirm the genuineness and caliber of herbal items.

Traditional medicine validation: Validate the traditional use of specific plant species for medicinal purposes using contemporary scientific methods.

By bridging the knowledge gap between traditional herbal medicine and contemporary machine learning, this initiative gives researchers, botanists, and herbalists a tool to correctly and rapidly identify possible medicinal compounds in leaves.

4.2 DESCRIPTION OF THE MODULES

The quality and effectiveness of medicinal plant leaves in medical applications depend on an accurate evaluation of their leaf properties. In order to improve the model's adaptability and traceability, this study investigates the use of artificial intelligence (AI) approaches to accurately estimate a variety of leaf parameters, such as length, width, area, perimeter, and form descriptors. Additionally, geolocation detection is used.

The suggested approach makes use of sophisticated image processing and machine learning methods in addition to a sizable dataset of photos of medical plant leaves. Accurate feature extraction is made possible by using image segmentation algorithms to separate individual leaves from their backdrop. The segmented leaf photos are used to extract a variety of texture, shape, and color features that give a rich depiction of the morphological traits of the leaves. In order to determine the plant's geographic origin, geolocation data is also taken from the image metadata or outside databases.

A chi-square feature selection method is used to extract the most discriminative characteristics, lowering the feature set's dimensionality while keeping the most pertinent data. Then, using the collected features, a variety of machine learning models are trained to predict the leaf properties, such as support vector regression (SVR), random forest, and artificial neural networks (ANNs).

Standard metrics like mean absolute error (MAE) and root mean squared error (RMSE) are used to assess each machine learning model's performance. With MAE and RMSE values continuously below 10% for all assessed attributes, the results show that the suggested AI-based technique achieves great accuracy in assessing various leaf characteristics.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 DESCRIPTION

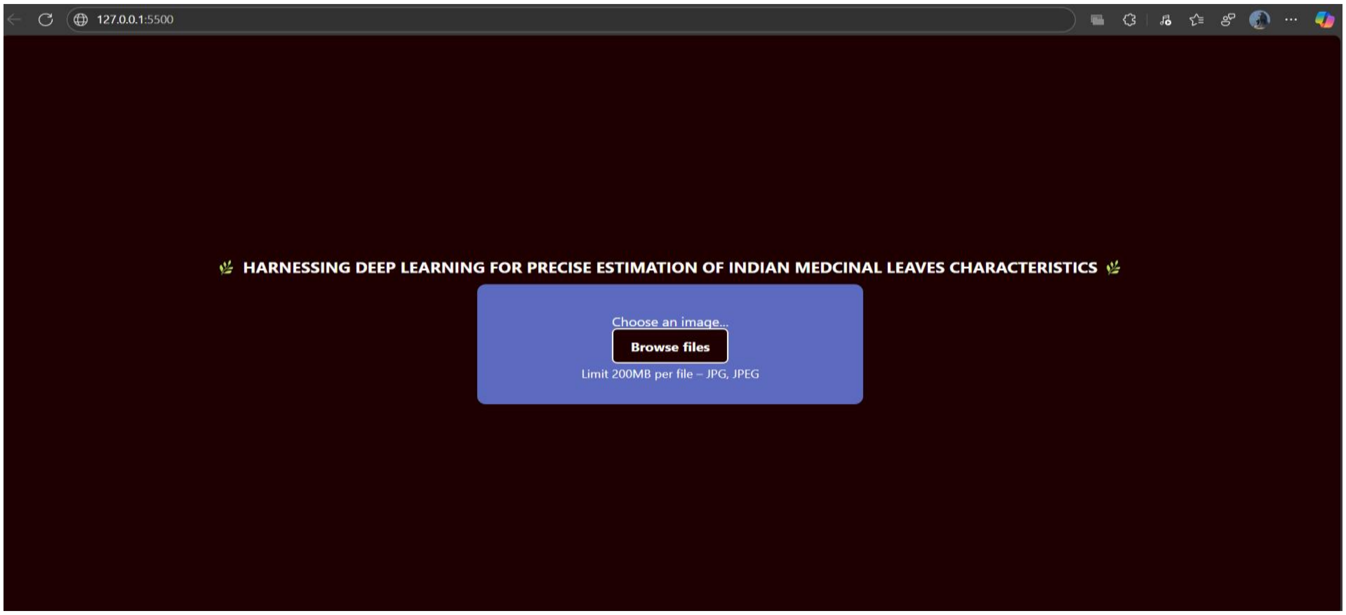

Figure 5.1 - Home page of the proposed system

In figure 5.1 the front page of the project where it gives an option for the user to browse the files were they can select the image of the leaves from the dataset.

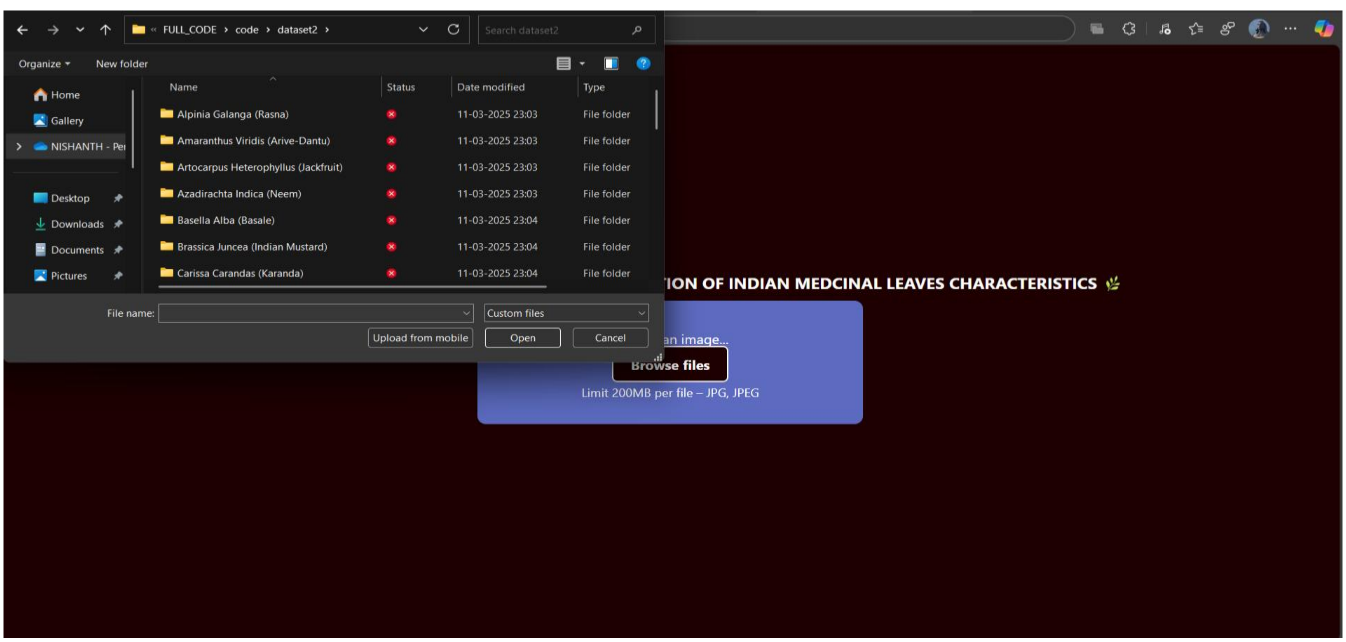

Figure 5.2 - Input of leaves images from the Dataset

In figure 5.2 the where the user can select the images of the medicinal leaves from the dataset.

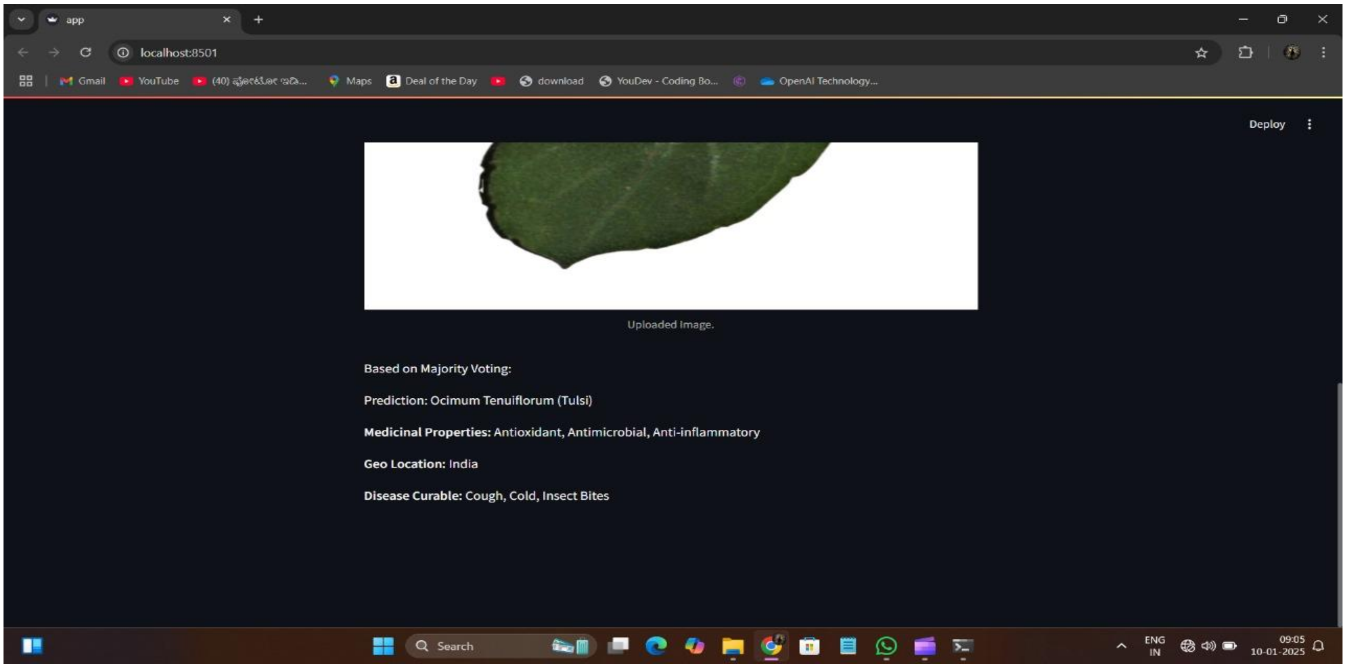

Figure 5.3 - User input image of neem leaf.

This figure 5.3 demonstrates the uploaded image is identified as Ocimum Tenuiflorum through majority voting. It exhibits medicinal properties including antibacterial, antifungal, and antiviral effects, commonly used in traditional medicine. Geolocation data suggests its origin in India, with reported curative potential for Cough, Cold, Insect Bites . Additionally, it generally safe for adults and children,

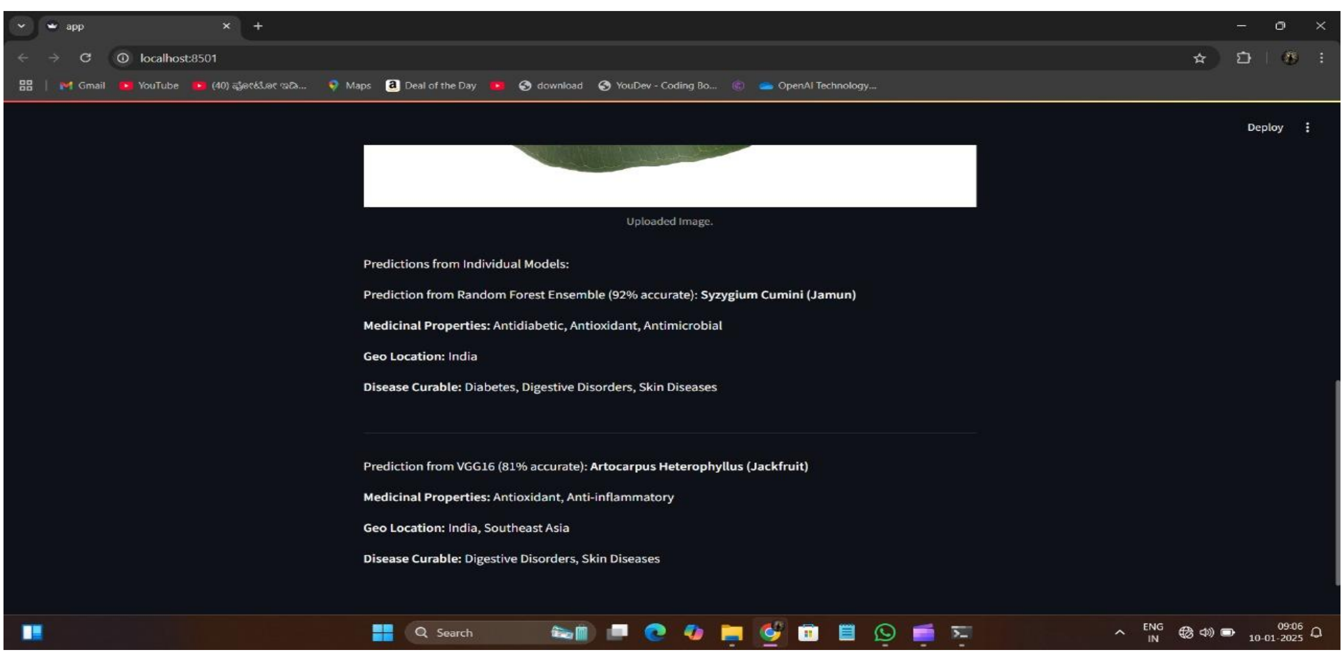

Figure 5.4 - Prediction of medicinal leaf based on majority voting

The Individual model predicts the uploaded image to be Syzygium cumini. It is associated with medicinal properties such as antidiabetic, Antioxidant, Antimicrobial commonly used in traditional medicine in India and Southeast Asia. Reported curative potential includes Digestive Disorders,Skin Diseases.However, caution is advised due to potential effects such as allergic reactions and gastrointestinal discomfort especially during pregnancy and for young children. Consultation with a healthcare is recommended

5.2 GRAPHS

Figure 5.5 - Training and Validation Accuracy and Loss Graph of deep Learning model

The figure 5.5 depicts two graphs likely visualizing the performance of a machine learning model during training. The x-axis represents epochs, which are iterations where the model is exposed to the training data. The left y-axis shows accuracy, while the right y-axis shows loss.

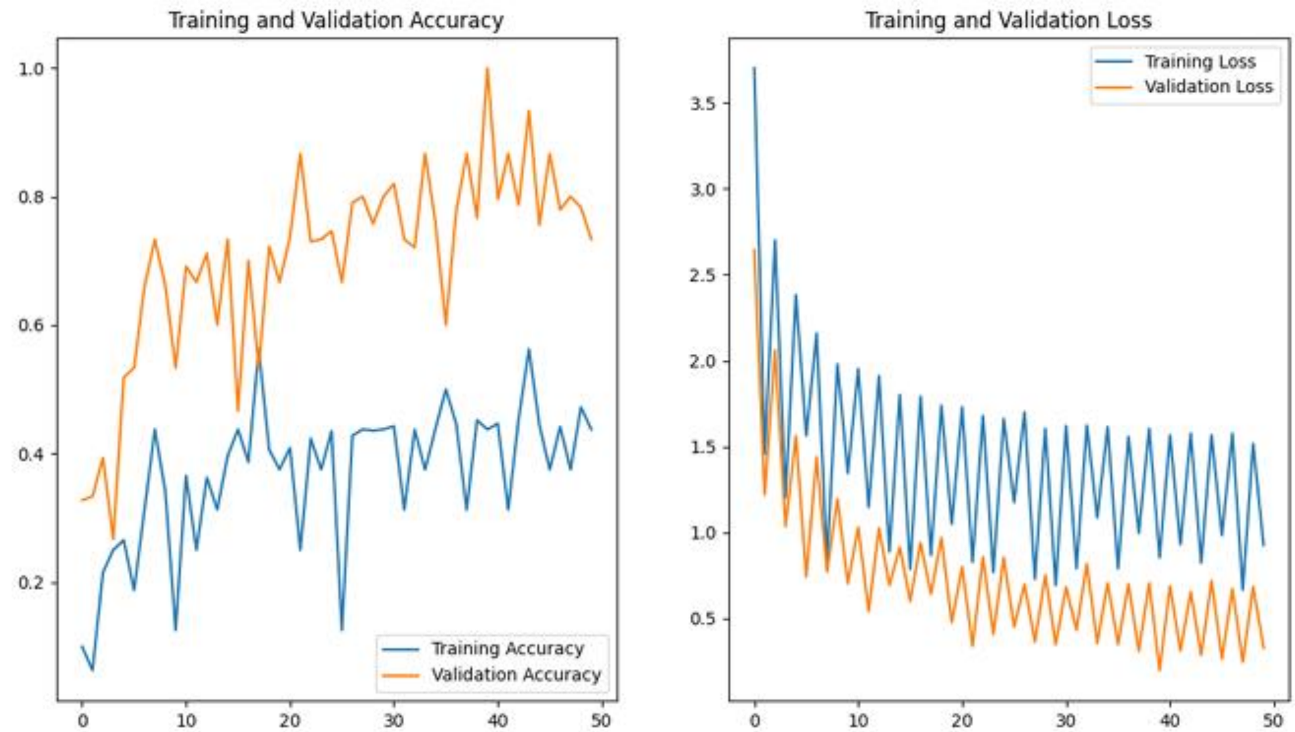

Figure 5.6 - Training and Validation Accuracy and Loss Graph of deep Learning model

In figure 5.6 The two graphs displayed in the image depict the training and validation performance of a machine learning model over multiple epochs. The x-axis represents the epochs, or iterations of the training process, while the y-axis on the left side indicates accuracy, and the right side indicates loss.

CHAPTER 6
TESTING

Testing is the act of checking and ensuring that a software product or program performs what it is intended to do which is a critical step in the product's development life cycle. While testing, the program to be tested was run with a list of test cases and the output of the program for the test cases was compared to see if the program is working as required. Errors were identified and corrected using the following steps of testing and correction was noted for future references.

Therefore, a sequence of testing was conducted on the system prior to implementation readiness. One significant aspect is that software testing must be separated from the independent discipline of Software Quality Assurance (SQA), which covers all business process areas and not only testing..

6.1 DESCRIPTION

Testing is part of Verification and Validation. Testing plays a very critical role for quality assurance and for ensuring the reliability of the software.

The objective of testing can be stated in the following ways.

• A successful test is one that uncovers as-yet-undiscovered bugs.

• A better test case has high probability of finding un-noticed bugs.

• A pessimistic approach of running the software with the intent of finding errors.

Testing can be performed in various levels like unit test, integration test and system test.

6.1.1 Unit Testing

Unit testing examines individual units so that they can be sure about that individual units work appropriately. All units were examined independently from another component of the system. It was examined by providing all correct sets of test data to individual modules and the obtained output compared with actual output. Verification effort on a unit is centered during unit testing on the minimum unit of software design module. Unit testing usually carries out by the developer. At first, the user attempts to login with credentials, if the credential is valid the user logs in successfully. Otherwise, user gets the error message. User can then login successfully check the state of onion by clicking the picture of onion..

6.1.2 Integration Testing

Integration testing is another aspect of testing that is generally done in order to uncover errors associated with the flow of data across interfaces. The unit-tested modules are grouped together and tested in small segment, which makes it easier to isolate and correct errors. This approach is continued until we have integrated all modules to form the system as a whole. After scanning the onion the user has to click on predict button of the application. After pressing the button, the custom model will classify the image into one of the three categories and displays the class name on the screen of the application.

6.1.3 System Testing

System testing is a method of software testing where a complete integrated system is tested to assess how well the system complies with the respective requirements. During system testing, integration testing passed components are used as input. The purpose of integration testing is to identify any abnormality among the units being integrated together. System testing identifies defects in both the integrated units and the entire system. Outcome of system testing is the behavior which is seen of a component or a system on being tested. System testing is indeed a chain of various tests whose only reason is to execute the full.

6.1.4 Acceptance Testing

Acceptance testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not. After testing the system at different levels, how the complete system is accepted which meets all the mentioned functional and non functional requirements. Depending upon the organization, acceptance testing might take the form of beta testing, field testing or end-user testing.

6.2 Test Cases

In software development, a test case is a description of the inputs, conditions of execution, testing process, and results expected thät specify an individual test to be run in order to attain a specific software testing goal, for instance, to execute a specific program path or to check conformity to a specific requirement.

Test cases form the basis for testing that is systematic rather than random. A set of test cases can be constructed to generate the desired coverage of the software under test. Formally defined test cases.

Test Number	Test Case ID	Test Case	Expected Results	Actual Output	Status
1	Unit Test 1	Open medicinal leaf Prediction Application	User should be able to launch application successfully	As Expected result	Pass
2	Unit Test 2	Select Leaf image from gallery	User can select the leaf image from the dataset	As Expected result	Pass
3	Unit Test 3	Predict the name of the medicinal leaf and its characteristics using deep learning model	Successfully predict the medicinal leaves and its characteristics	As Expected result	Pass
4	Unit Test 4	Predict the name of the medicinal leaf and its characteristics using machine learning model	Successfully predict the medicinal leaves and its characteristics	As Expected result	Pass
5	Unit Test 5	Predict the medicinal leaves and its characteristics using majority voting combining deep learning and machine learning	Successfully predict the medicinal leaves and its characteristics based on majority voting	As Expected result	Pass
6	Unit Test 6	Predict the medicinal leaves and display the result	Successfully predict the medicinal leaves and its characteristics	As Expected result	Pass
7	Unit Test 7	User can upload random image other than trained image	Displays the message “Data not found”	As Expected result	Pass

Table 6.1-Test cases

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION AND FUTURE WORK

By combining state-of-the-art artificial intelligence (AI) technologies like machine learning and deep learning, the project " Harnessing Deep Learning For Precise Estimation Of Indian Medicinal Leaves Characteristics " represents a significant advancement in the field of medicinal plant identification. By developing advanced AI models, the project has demonstrated remarkable accuracy in identifying medicinal plant species based on their leaf characteristics, facilitating safer and more efficient use of herbal remedies. Streamlit's user-friendly interface guarantees accessibility for researchers, botanists, and enthusiasts, encouraging widespread use and understanding.

To sum up, the project has effectively tackled the main goals of creating precise AI models for the identification of medicinal plants, encouraging sustainable harvesting methods, reducing the health risks related to incorrect identification, and fusing traditional botanical knowledge with contemporary technology. Still, there is a lot of room for more research. To improve the resilience and adaptability of the AI models, one direction for future research would be to broaden the dataset to include a greater range of medicinal plants and their unique traits.Furthermore, ongoing improvement and optimization of the AI algorithms might boost precision and effectiveness even more, guaranteeing the system's dependability in a variety of plant species and environmental circumstances. Furthermore, working together with specialists in ethnobotany and traditional medicine can add insightful knowledge to the system and increase its acceptability and practicality in everyday situations. All things considered, the initiative establishes a solid basis for future study and advancement in the field of medicinal plant identification, with encouraging ramifications for sustainable development, healthcare, and conservation.

7.2 LIMITATIONS OF THE PROJECT

1. **Limited Dataset Size**
   The training dataset has a significant impact on the model's performance. The model's capacity to generalize to novel, untested data may be diminished by a small or unbalanced dataset.

2. **Image Quality and Consistency**
   Accuracy of feature extraction and categorization may be impacted by changes in image lighting, perspective, background, or resolution.

3. **Restricted to Specific Medicinal Leaves**
   Only a few number of Indian medicinal leaves are recognized under the current system. For leaves that don't fall under this category, it could not work well.

4. **Environmenta Dependency**
   When photos are taken under uncontrolled settings, including with occlusions or ambient background noise, the classification accuracy may decrease.

5. **Hardware and Processing Constraints**
   Usability in field settings may be impacted by real-time prediction's inefficiency on low-end devices or without GPU acceleration.

6. **Lack of Real-World Testing**
   The model has mostly been tested in a controlled setting. There is still a need for field testing by botanists or herbal practitioners as well as real-world validation.

7. **User Interface Limitations**
   Advanced functionality like offline support, language support, and multi-leaf recognition may be absent from the program interface, making it less usable for a wide range of users.

7.3 ADVANTAGES OF THE PROJECT

- Effective Disease Prediction: By correctly identifying and categorizing leaf diseases, the model reduces crop loss and enables prompt intervention.
- Economical Solution: By utilizing readily available technology, the project provides farmers and other agricultural professionals with an affordable way to keep an eye on plant health.
- User-friendly Interface: Users without technical expertise can easily manage the system thanks to its straightforward graphical user interface.
- Time-saving: The amount of time needed for manual inspection is greatly decreased when the disease detection procedure is automated.
- Environmentally Beneficial: Accurate and timely disease detection aids in the targeted application of pesticides, minimizing damage to the environment.
- Scalable Design: The project's applicability can be increased by adding more plant species and diseases.
- Educational Value: Offers researchers and students studying AI, image processing, and agriculture a robust learning environment.

7.4 APPLICATIONS OF THE PROJECT

The project has wide-ranging applications across multiple domains that intersect technology, healthcare, agriculture, and environmental sustainability. Key applications include:

1. **Medicinal Plant Identification for Ayurveda and Traditional Medicine**:
   In addition to helping in the creation of herbal cures and the preservation of indigenous and knowledge systems, this technique can help practitioners and researchers reliably identify's medicinal leaves.

2. **Agricultural Extension and Support for Farmers**:
   Through smartphone applications, farmers can utilize the model to find beneficial medicinal plants on their property, promoting sustainable agriculture and diversifying revenue streams.

3. **Digital Herbariums and Botanical Research**:
   By accurately classifying and documenting medicinal plants, the model can assist ecologists and botanists in building digital herbariums.

4. **Pharmaceutical and Biotech Industries**:
   When developing plant-based medications, businesses can use the technology to quickly filter and classify plant samples.

5. **Educational Tools and E-learning Platforms**:
   To teach students about the taxonomy, morphology, applications of medicinal plants, educational institutions might include the system into e-learning resources.

6. **Environmental Monitoring and Conservation**:
   By automatically identifying uncommon or endangered medicinal plant species from field and photos, the model can help conservationists keep an eye on them.

7. **Smartphone-based Health Apps**:
   For educational objectives, integration with health and wellness applications can help regular as people recognize medicinal plants in the wild.

8. **Government Schemes and Rural Health Missions**:
   The application can help carry out government programs that support natural resource and management and herbal treatment in rural areas.

7.5 FUTURE ENHANCEMENTS

For improved accuracy, the project can be improved in the future by adding more medicinal leaf kinds to the dataset. Direct leaf scanning could be made possible by adding a real-time camera input function. The system would be more accessible, particularly in rural regions, if it had offline functionality and multilingual support. Plant positions may be tracked with GPS integration, and usability would be enhanced by the use of mobile apps. The user experience and model performance can also be improved by adding voice assistance, augmented reality capabilities, and feedback-based learning.

APPENDICES

APPENDIX A

Harnessing Deep Learning Model For Precise Estimation Of Indian Medicinal Leaves Characteristics

Nithin P
dept. Computer Science and Engineering
Alliance School Of Adavanced Computing Alliance University
nithinpreddy18@gmail.com

Nishanth B G
dept. Computer Science and Engineering
Alliance School Of Adavanced Computing Alliance University
nishanthggowda2@gmail.com

MEGHRAJ G
dept. Computer Science and Engineering
Alliance School Of Adavanced Computing Alliance University
gopalmeghraj27@gmail.com

Sameer Khan
dept. Computer Science and Engineering
Alliance School Of Adavanced Computing Alliance University
sk64494250@gmail.com

Abstract

India has a diverse collection of medicinal plants used in customary medicine for centuries. However, the accurate identification and characterization of these plants remain a challenge. This is due to the important variety and complexity of their physical features. Typical ways for leaf analysis, often based on manual inspection and on chemical analysis, are not only time-consuming, but also can lead to inaccuracies. This project aims for effectively harnessing the power of deep learning models for improving the precision and efficiency within the estimation of characteristics from Indian medicinal leaves. The suggested method makes leaf identification automatic as well as analysis automatic with convolutional neural networks (CNNs) along with other modern deep learning methods. It works based on several morphological features like shape, texture, and venation pattern. The model is trained by use of an wide-ranging dataset with images of several medicinal leaves, collected from the multiple regions throughout India. These images are pre-processed and labeled with detailed information regarding leaf characteristics like medicinal properties, size, and texture. A machine learning model is completely prepared to spot main parts from the pictures, which lets it sort and describe things closely. This study analyzes multiple deep learning designs in order to see which performs best regarding correctness, velocity, and strength. First outcomes suggest this model is able to recognize and assess medicinal leaves' traits with high precision, greatly exceeding multiple conventional approaches Furthermore, the model could be enlarged so as to cover additional species; in addition to making it helpful to botanists, researchers, as well as pharmaceutical companies that study herbal treatments. By automating and making leaf analysis more precise, the approach may make medicinal plant identification research more efficient while protecting plant species and assisting with new therapeutic compound discovery. This undertaking represents a definite improvement within the merging of AI and ethnobotanical studies and stresses the ways that deep learning might transform plant-based medicine. This work subsequently could be greatly expanded so as to include other multiple plant parts. Other types of data, like complete genomic data, may be thoroughly incorporated to further improve the in-depth comprehension of therapeutic plants

1. Introduction

India has within its borders a rich and diverse flora, with thousands of plant species that, for centuries upon centuries, have been used in customary medicine. These plants, many thought to have great medicinal qualities, are fundamental to multiple medical systems, like Ayurveda, Siddha, and Unani. The precise identification as well as detailed characterization of these plants, especially their leaves, is important for comprehensively preserving their therapeutic potential. This identification and characterization is additionally important for considerably advancing pharmaceutical research along with applications. However, customary methods for leaf analysis, like manual inspection along with chemical analysis, often require time, effort, plus expertise that is meaningful. Also, such methods are inclined toward inaccuracies, especially when working among large volumes within plant species that have variations in leaf morphology. New progress within deep learning and computer vision presents good ways for solving the problems in standard plant analysis methods. Deep learning techniques, especially convolutional neural networks (CNNs), have revealed great success in the field of image recognition, allowing systems to identify objects in addition to extracting features of images with a high degree of accuracy. This project seeks the application of

APPENDIX B

Emerging Trends in Advancements and Applications of Computational Intelligence Techniques : Submission (27) has been created.

MC Microsoft CMT<noreply@msr-cmt.org>

To: You

Reply Reply all Forward ···

Thu 5/15/2025 8:03 AM

Hello,

The following submission has been created.

Track Name: ETAACT2026

Paper ID: 27

Paper Title: Harnessing Deep Learning Model For Precise Estimation Of Indian Medicinal Leaves Characteristics

Abstract:
India has a rich heritage of medicinal plants used in traditional medicine for centuries. However, accurate identification and analysis of these plants remain a challenge due to the complexity and variability of their physical features. Traditional methods based on manual inspection and chemical analysis are not only time-consuming but also prone to inaccuracies. This project proposes the use of deep learning models to improve the precision and efficiency in identifying and analyzing Indian medicinal leaves.

The suggested method automates leaf identification and analysis using Convolutional Neural Networks (CNNs) and other modern deep learning techniques. It focuses on morphological features like leaf shape, texture, and venation patterns. A large and diverse dataset of medicinal leaf images, collected from various regions of India, is used for training. These images are pre-processed and labeled with detailed attributes such as medicinal properties, size, and texture.

The machine learning model is trained to extract key features from the images, enabling accurate classification and description of the leaves. Several deep learning architectures are evaluated to determine the best-performing model in terms of accuracy, speed, and robustness. Initial results show that the model significantly outperforms traditional methods in recognizing and analyzing medicinal leaf characteristics.

This approach has the potential to support botanists, researchers, and pharmaceutical companies by automating leaf analysis and enhancing research efficiency. Additionally, the system can be extended to identify other plant species and even other plant parts. The integration of additional data types, such as genomic information, can further enhance the understanding of medicinal plants. Overall, this project represents a significant advancement in combining AI with ethnobotanical research, with promising applications in plant-based medicine and conservation.

Created on: Thu, 15 May 2025 08:03:36 GMT

Last Modified: Thu, 15 May 2025 08:03:36 GMT

Authors:
- sk6449425@gmail.com (Primary)
- nithinpreddy18@gmail.com
- nishanthggowda2@gmail.com
- gopalmeghraj27@gmail.com

Secondary Subject Areas: Not Entered

Submission Files:
Sameer conference paper 1.pdf (163 Kb, Thu, 15 May 2025 08:03:33 GMT)

Submission Questions Response: Not Entered

Thanks,
CMT team.

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our Privacy Statement.

Microsoft Corporation

APPENDIX C

Source Code

App Code

```
import os

import streamlit as st

import numpy as np

from tensorflow.keras.preprocessing import image

from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input

from tensorflow.keras.models import Model

from sklearn.ensemble import RandomForestClassifier

import joblib

from PIL import Image

import io

import tensorflow as tf

# Load the VGG16 model with the specified weights file

base_model=VGG16(weights='vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5',
include_top=False)

model_vgg = Model(inputs=base_model.input, outputs=base_model.get_layer('block5_conv2').output)


# Load the Random Forest classifier

rf_classifier = joblib.load('model_random_forest_dataset2.pkl')
```

```
# Load the trained VGG16 model for classification
model = tf.keras.models.load_model('model_vgg_dataset2.h5')


# Get the class names for the VGG16 model
class_names = sorted(os.listdir('dataset2'))  # replace with your actual directory


# Additional data based on prediction
data = data = {
    "Alpinia Galanga (Rasna)": {
        "medicinal_properties": ["Anti-inflammatory", "Antioxidant", "Antimicrobial"],
        "geo_location": ["India", "Southeast Asia"],
        "disease_curable": ["Arthritis", "Indigestion"]
    },
    "Amaranthus Viridis (Arive-Dantu)": {
        "medicinal_properties": ["Antioxidant", "Anti-inflammatory", "Antibacterial"],
        "geo_location": ["Worldwide"],
        "disease_curable": ["Asthma", "Diabetes", "High Blood Pressure"]
    },
    "Artocarpus Heterophyllus (Jackfruit)": {
        "medicinal_properties": ["Antioxidant", "Anti-inflammatory"],
        "geo_location": ["India", "Southeast Asia"],
        "disease_curable": ["Digestive Disorders", "Skin Diseases"]
    },
```

"Azadirachta Indica (Neem)": {
"medicinal_properties": ["Antibacterial", "Antifungal", "Antiviral"],
"geo_location": ["India"],
"disease_curable": ["Skin Diseases", "Malaria", "Diabetes"]
},
"Basella Alba (Basale)": {
"medicinal_properties": ["Antioxidant", "Anti-inflammatory", "Anticancer"],
"geo_location": ["India"],
"disease_curable": ["Constipation", "Anemia"]
},
"Brassica Juncea (Indian Mustard)": {
"medicinal_properties": ["Antibacterial", "Antifungal", "Anticancer"],
"geo_location": ["India"],
"disease_curable": ["Cough", "Asthma", "Bronchitis"]
},
"Carissa Carandas (Karanda)": {
"medicinal_properties": ["Antioxidant", "Anti-inflammatory", "Antidiabetic"],
"geo_location": ["India"],
"disease_curable": ["Diabetes", "Wounds", "Ulcers"]
},
"Citrus Limon (Lemon)": {
"medicinal_properties": ["Antioxidant", "Antibacterial", "Antiviral"],
"geo_location": ["Worldwide"],

"disease_curable": ["Scurvy", "Indigestion", "Skin Care"]
},
"Ficus Auriculata (Roxburgh fig)": {
"medicinal_properties": ["Antioxidant", "Antidiabetic", "Antimicrobial"],
"geo_location": ["India", "Southeast Asia"],
"disease_curable": ["Diabetes", "Skin Diseases", "Wounds"]
},
"Ficus Religiosa (Peepal Tree)": {
"medicinal_properties": ["Antioxidant", "Anti-inflammatory", "Antidiabetic"],
"geo_location": ["India"],
"disease_curable": ["Asthma", "Jaundice", "Diabetes"]
},
"Hibiscus Rosa-sinensis": {
"medicinal_properties": ["Antioxidant", "Anti-inflammatory", "Antibacterial"],
"geo_location": ["Tropical Regions"],
"disease_curable": ["Hair Loss", "Hypertension", "Cough"]
},
"Jasminum (Jasmine)": {
"medicinal_properties": ["Antidepressant", "Antiseptic", "Antispasmodic"],
"geo_location": ["Tropical Regions"],
"disease_curable": ["Anxiety", "Skin Diseases", "Menstrual Disorders"]
},
"Mangifera Indica (Mango)": {

```
"medicinal_properties": ["Antioxidant", "Anti-inflammatory", "Anticancer"],
        "geo_location": ["India", "Southeast Asia"],
        "disease_curable": ["Indigestion", "Heat Stroke", "Anemia"]
    },
    "Mentha (Mint)": {
        "medicinal_properties": ["Antimicrobial", "Antispasmodic", "Digestive Aid"],
        "geo_location": ["Worldwide"],
        "disease_curable": ["Indigestion", "Nausea", "Headache"]
    },
    "Moringa Oleifera (Drumstick)": {
        "medicinal_properties": ["Antioxidant", "Anti-inflammatory", "Antidiabetic"],
        "geo_location": ["India", "Africa"],
        "disease_curable": ["Diabetes", "Anemia", "Malnutrition"]
    },
    "Muntingia Calabura (Jamaica Cherry-Gasagase)": {
        "medicinal_properties": ["Antioxidant", "Antimicrobial", "Anti-inflammatory"],
        "geo_location": ["Tropical Regions"],
        "disease_curable": ["Fever", "Hypertension", "Diabetes"]
    },
    "Murraya Koenigii (Curry)": {
        "medicinal_properties": ["Antioxidant", "Antimicrobial", "Anti-inflammatory"],
        "geo_location": ["India"],
        "disease_curable": ["Diabetes", "Diarrhea", "Nausea"]
```

},
"Nerium Oleander (Oleander)": {
"medicinal_properties": ["Cardiotonic", "Anticancer", "Antimicrobial"],
"geo_location": ["Mediterranean Region", "Asia"],
"disease_curable": ["Heart Diseases", "Cancer"]
},
"Nyctanthes Arbor-tristis (Parijata)": {
"medicinal_properties": ["Antipyretic", "Antiarthritic", "Antioxidant"],
"geo_location": ["India"],
"disease_curable": ["Fever", "Arthritis", "Skin Diseases"]
},
"Ocimum Tenuiflorum (Tulsi)": {
"medicinal_properties": ["Antioxidant", "Antimicrobial", "Anti-inflammatory"],
"geo_location": ["India"],
"disease_curable": ["Cough", "Cold", "Insect Bites"]
},
"Piper Betle (Betel)": {
"medicinal_properties": ["Antibacterial", "Antifungal", "Antioxidant"],
"geo_location": ["India", "Southeast Asia"],
"disease_curable": ["Oral Health", "Digestive Disorders"]
},
"Plectranthus Amboinicus (Mexican Mint)": {
"medicinal_properties": ["Antibacterial", "Antifungal", "Antioxidant"],

"geo_location": ["India", "Southeast Asia"],
    "disease_curable": ["Respiratory Disorders", "Digestive Disorders"]
  },
  "Pongamia Pinnata (Indian Beech)": {
    "medicinal_properties": ["Antibacterial", "Antifungal", "Antioxidant"],
    "geo_location": ["India", "Southeast Asia"],
    "disease_curable": ["Skin Diseases", "Wounds", "Rheumatism"]
  },
  "Psidium Guajava (Guava)": {
    "medicinal_properties": ["Antioxidant", "Antimicrobial", "Anti-inflammatory"],
    "geo_location": ["Tropical Regions"],
    "disease_curable": ["Diarrhea", "Dysentery", "Skin Disorders"]
  },
  "Punica Granatum (Pomegranate)": {
    "medicinal_properties": ["Antioxidant", "Anticancer", "Antimicrobial"],
    "geo_location": ["Middle East", "India"],
    "disease_curable": ["Heart Diseases", "Diabetes", "High Blood Pressure"]
  },
  "Santalum Album (Sandalwood)": {
    "medicinal_properties": ["Antiseptic", "Anti-inflammatory", "Astringent"],
    "geo_location": ["India", "Australia"],
    "disease_curable": ["Skin Diseases", "Urinary Tract Infections"]
  },

```
  "Syzygium Cumini (Jamun)": {
    "medicinal_properties": ["Antidiabetic", "Antioxidant", "Antimicrobial"],
    "geo_location": ["India"],
    "disease_curable": ["Diabetes", "Digestive Disorders", "Skin Diseases"]
  },
  "Syzygium Jambos (Rose Apple)": {
    "medicinal_properties": ["Antioxidant", "Antimicrobial", "Anticancer"],
    "geo_location": ["Southeast Asia"],
    "disease_curable": ["Diabetes", "Digestive Disorders", "Cancer"]
  },
  "Tabernaemontana Divaricata (Crape Jasmine)": {
    "medicinal_properties": ["Antipyretic", "Antispasmodic", "Anti-inflammatory"],
    "geo_location": ["India", "Southeast Asia"],
    "disease_curable": ["Fever", "Muscle Pain", "Inflammation"]
  },
  "Trigonella Foenum-graecum (Fenugreek)": {
    "medicinal_properties": ["Antidiabetic", "Antioxidant", "Anti-inflammatory"],
    "geo_location": ["India", "Mediterranean Region"],
    "disease_curable": ["Diabetes", "Digestive Disorders", "Skin Inflammation"]
  }
}
```

```
# Function to extract features using VGG16
def extract_features(img_path):
    img = Image.open(img_path)
    img = img.resize((224, 224))  # Resize the image to match VGG input
    x = np.array(img)
    x = np.expand_dims(x, axis=0)
    x = preprocess_input(x)
    features = model_vgg.predict(x)
    return features.flatten()


# Function to predict using Random Forest
def predict_rf(image_bytes):
    image = Image.open(io.BytesIO(image_bytes))
    new_image_path = 'uploaded_image.jpg'
    image.save(new_image_path)
    new_image_features = extract_features(new_image_path)
    prediction_features = new_image_features.reshape(1, -1)
    prediction = rf_classifier.predict(prediction_features)
    return prediction


# Function to load and prepare the image for VGG16 model
def load_and_prep_image(image, img_shape=224):
    """
```

```
  Reads an image from filename, turns it into a tensor and reshapes it to (img_shape, img_shape,, color_channels)
    """
    # Decode it into a tensor
    img = tf.image.decode_image(image)

    # Resize the image
    img = tf.image.resize(img, [img_shape, img_shape])

    # Rescale the image (get all values between 0 and 1)
    img = img/255.
    return img

# Streamlit app
st.title("Image Classification with VGG16 Deep Learning and Random Forest Ensemble")

uploaded_file = st.file_uploader("Choose an image...", type="jpg")

if uploaded_file is not None:
    image_bytes = uploaded_file.read()
    image = Image.open(io.BytesIO(image_bytes))
    st.image(image, caption='Uploaded Image.', use_column_width=True)
    st.write("")
```

```
    # Predict with VGG16
    image_vgg = load_and_prep_image(image_bytes)
    image_vgg = tf.expand_dims(image_vgg, axis=0)
    pred_vgg = model.predict(image_vgg)
    pred_class_vgg = class_names[np.argmax(pred_vgg)]

    # Predict with Random Forest
    prediction_rf = predict_rf(image_bytes)
    pred_class_rf = prediction_rf[0]

    if pred_class_vgg == pred_class_rf:
        st.write("Based on Majority Voting:")
        st.write(f"Prediction: {pred_class_vgg}")

        if pred_class_vgg in data:
            st.markdown(f"**Medicinal Properties:** {', '.join(data[pred_class_vgg]['medicinal_properties'])}")
            st.markdown(f"**Geo Location:** {', '.join(data[pred_class_vgg]['geo_location'])}")
            st.markdown(f"**Disease Curable:** {', '.join(data[pred_class_vgg]['disease_curable'])}")
        else:
            st.write("No additional data available")
    else:
        st.write("Predictions from Individual Models:")
```

```
st.markdown(f"Prediction from Random Forest Ensemble (92% accurate): **{pred_class_rf}**")
    if pred_class_rf in data:
      st.markdown(f"**Medicinal Properties:** {', '.join(data[pred_class_rf]['medicinal_properties'])}")
        st.markdown(f"**Geo Location:** {', '.join(data[pred_class_rf]['geo_location'])}")
        st.markdown(f"**Disease Curable:** {', '.join(data[pred_class_rf]['disease_curable'])}")
    else:
        st.write("No additional data available")
        st.markdown("---")  # Horizontal ruler


    st.markdown(f"Prediction from VGG16 (81% accurate): **{pred_class_vgg}**")
    if pred_class_vgg in data:
        st.markdown(f"**Medicinal Properties:** {',
    '.join(data[pred_class_vgg]['medicinal_properties'])}")
        st.markdown(f "geo Location:** {', '.join(data[pred_class_vgg]['geo_location'])}")
        st.markdown(f"**Disease Curable:** {', '.join(data[pred_class_vgg]['disease_curable'])}")
    else:
        st.write("No additional data available")
```

Train Randomforest

```
import numpy as np
from tqdm import tqdm
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.models import Model
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import joblib


# Load the VGG16 model with pre-trained ImageNet weights
base_model = VGG16(weights=None, include_top=False)
base_model.load_weights('vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5')


# Use the VGG16 model without the top layers
vgg_model = Model(inputs=base_model.input, outputs=base_model.get_layer('block5_conv2').output)


# Function to extract features using VGG16
def extract_features(img_path):
    img = image.load_img(img_path, target_size=(224, 224))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
```

```
  x = preprocess_input(x)
    features = vgg_model.predict(x)
    return features.flatten()

# Dataset directory
base_dir = 'dataset2'

# Extract features and labels from the dataset
X = []
y = []
for subdir in tqdm(os.listdir(base_dir), desc="Processing images", unit="dir"):
    subdir_path = os.path.join(base_dir, subdir)
    for img_name in os.listdir(subdir_path):
        img_path = os.path.join(subdir_path, img_name)
        features = extract_features(img_path)
        X.append(features)
        y.append(subdir)  # Assuming subdir is the class label

X = np.array(X)
y = np.array(y)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Train a Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train, y_train)

# Evaluate the Random Forest classifier
y_pred = rf_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Random Forest Accuracy:", accuracy)

# Save the trained Random Forest model
joblib.dump(rf_classifier, 'model_random_forest_dataset2.pkl')
```

Train VGG-16

```
import os
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Flatten, Dense, Dropout


# Define paths and parameters
dataset_dir = 'dataset2'
batch_size = 16
IMG_HEIGHT, IMG_WIDTH = 224, 224
epochs = 18


# Data preparation with image processing
image_generator = ImageDataGenerator(
    rescale=1./255,
```

```
    rotation_range=20,

    width_shift_range=0.2,

    height_shift_range=0.2,

    horizontal_flip=True,

    fill_mode='nearest',

    validation_split=0.2  # set validation split

)


train_data_gen = image_generator.flow_from_directory(

    batch_size=batch_size,

    directory=dataset_dir,

    shuffle=True,

    target_size=(IMG_HEIGHT, IMG_WIDTH),

    class_mode='categorical',

    subset='training'  # set as training data

)


validation_data_gen = image_generator.flow_from_directory(

    batch_size=batch_size,

    directory=dataset_dir,

    shuffle=True,

    target_size=(IMG_HEIGHT, IMG_WIDTH),

    class_mode='categorical',

    subset='validation'  # set as validation data)
```

```
# Load the VGG16 network with local weights
weights_path = 'vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5'  # path to your local weights file
baseModel = VGG16(weights=weights_path, include_top=False, input_shape=(IMG_HEIGHT,
IMG_WIDTH, 3))


# Construct the head of the model
headModel = baseModel.output
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(512, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(len(train_data_gen.class_indices), activation="softmax")(headModel)


# Place the head FC model on top of the base model
model = Model(inputs=baseModel.input, outputs=headModel)


# Freeze the layers in the base model
for layer in baseModel.layers:
    layer.trainable = False



# Compile the model
model.compile(optimizer='adam',
              loss=tf.keras.losses.CategoricalCrossentropy(),
```

```
    metrics=['accuracy'])


# Train the model
history = model.fit(
    train_data_gen,
    steps_per_epoch=train_data_gen.samples // batch_size,
    epochs=epochs,
    validation_data=validation_data_gen,
    validation_steps=validation_data_gen.samples // batch_size
)


# Save the model
model.save('model_vgg_dataset2.h5')


# Plot training history
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']


loss = history.history['loss']
val_loss = history.history['val_loss']


epochs_range = range(epochs)
```

```
plt.figure(figsize=(8, 8))

plt.subplot(1, 2, 1)

plt.plot(epochs_range, acc, label='Training Accuracy')

plt.plot(epochs_range, val_acc, label='Validation Accuracy')

plt.legend(loc='lower right')

plt.title('Training and Validation Accuracy')


plt.subplot(1, 2, 2)

plt.plot(epochs_range, loss, label='Training Loss')

plt.plot(epochs_range, val_loss, label='Validation Loss')

plt.legend(loc='upper right')

plt.title('Training and Validation Loss')

plt.show()


# Evaluate the model on the test set

print("Evaluating the model...")

test_loss, test_accuracy = model.evaluate(validation_data_gen, verbose=2)

print(f"Test Accuracy: {test_accuracy*100:.2f}%")

print(f"Test Loss: {test_loss:.4f}")




# Generate predictions

print("Generating predictions...")
```

```
predictions = model.predict(validation_data_gen)
y_pred = np.argmax(predictions, axis=1)
y_true = validation_data_gen.classes

# Generate classification report and confusion matrix
print("Classification Report:")
print(classification_report(y_true, y_pred, target_names=train_data_gen.class_indices.keys()))

print("Confusion Matrix:")
cm = confusion_matrix(y_true, y_pred)
print(cm)

# Plot confusion matrix as a heatmap
plt.figure(figsize=(10, 10))
sns.heatmap(cm, annot=True, fmt="d", cmap='Blues', xticklabels=train_data_gen.class_indices, yticklabels=train_data_gen.class_indices)
plt.title('Confusion Matrix')
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.show()

# Plot training metrics
plt.figure(figsize=(8, 8))
```

```
plt.subplot(2, 1, 1)
plt.plot(epochs_range, history.history['accuracy'], label='Training Accuracy')
plt.plot(epochs_range, history.history['val_accuracy'], label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(2, 1, 2)
plt.plot(epochs_range, history.history['loss'], label='Training Loss')
plt.plot(epochs_range, history.history['val_loss'], label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```

Dataset Preparation With Augmentation

```
import os

import shutil

import numpy as np

from tqdm import tqdm

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define paths

base_dir = 'dataset2'

target_dir = 'augmenteddataset2'

number_of_combinations = 2  # User can set this value

# Create target directory

os.makedirs(target_dir, exist_ok=True)

# Initialize ImageDataGenerator for data augmentation

datagen = ImageDataGenerator(

    rotation_range=20,

    width_shift_range=0.2,

    height_shift_range=0.2,

    horizontal_flip=True)

# Loop over subdirectories

for subdir in tqdm(os.listdir(base_dir), desc="Processing images", unit="dir"):

    # Get list of files in subdir

    subdir_path = os.path.join(base_dir, subdir)
```

```
 files=[os.path.join(subdir_path,f)for f in os.listdir(subdir_path) if os.path.isfile(os.path.join(subdir_path,
f))
    np.random.shuffle(files)
 # Split files into original and augmented sets
    original_files = files[:len(files)//2]  # original images
    augmented_files = files[len(files)//2:]  # augmented images
# Create class directories in target directory
    os.makedirs(os.path.join(target_dir, subdir), exist_ok=True)
    for f in original_files:
        shutil.copy(f, os.path.join(target_dir, subdir))
 # Augment and copy augmented files to target directory
    for f in augmented_files:
        image = tf.keras.preprocessing.image.load_img(f)
        x = tf.keras.preprocessing.image.img_to_array(image)
        x = x.reshape((1,) + x.shape)
        i = 0
            for batch in datagen.flow(x, batch_size=1, save_to_dir=os.path.join(target_dir, subdir),
save_prefix='aug', save_format='jpeg'):
           i += 1
           if i >= number_of_combinations:  # create specified number of augmented images per original
image
              break


print("Dataset augmented and saved in target directory.")
```

RFERENCES & BIBLIOGRAPHY

[1] Naeem, S.; Ali, A.; Chesneau, C.; Tahir, M.H.; Jamal, F.; Sherwani, R.A.K.; Ul Hassan, M. The classification of medicinal plant leaves based on multispectral and texture feature using machine learning approach. Agronomy 2021.

[2] Automatic Recognition of Medicinal Plants using Machine Learning Techniques, International Journal of Advanced Computer Science and Applications 2021.

[3] Mukherjee, G.; Tudu, B.; Chatterjee, A. A convolutional neural networkdriven computer vision system toward identification of species and maturity stage of medicinal leaves: Case studies with Neem, Tulsi and Kalmegh leaves. Soft Computer. 2021.

[4] Bisen, D. Deep convolutional neural network based plant species recognition through features of leaf. Multimed. Tools Appl. 2021.

[5] Azadnia, R.; Kheiralipour, K. Recognition of leaves of different medicinal plant species using a robust image processing algorithm and artificial neural networks classifier. J. Appl. Res. Med. Aromat. Plants 2021.

[6] Russel, N.S.; Selvaraj, A. Leaf species and disease classification using multiscale parallel deep CNN architecture. Neural Comput. Appl. 2022.

[7] Paulson, A.; Ravishankar, S. AI Based Indigenous Medicinal Plant Identification. In Proceedings of the 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCT HPA), Cochin, India, 2–4 July 2020.

[8] Muneer, A.; Fati, S.M. Efficient and automated herbs classification approach based on shape and texture features using deep learning. IEEE Access 2020.

[9] Reddy, S.R.; Varma, G.P.; Davuluri, R.L. Optimized convolutional neural network model for plant species identification from leaf images using computer vision. Int. J. Speech Technol. 2021.

[10] Roopashree, S.; Anitha, J. DeepHerb: A Vision Based System for Medicinal Plants Using Xception Features. IEEE Access 2021.

Harnessing Deep Learning For Precise Estimation Of Indian Medicinal .pdf

ORIGINALITY REPORT

14%	8%	5%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.kscst.org.in Internet Source	4%
2	Submitted to Institute of Technology, Nirma University Student Paper	3%
3	Submitted to Alliance University Student Paper	3%
4	Bui Thanh Hung, M. Sekar, Ayhan Esi, R. Senthil Kumar. "Applications of Mathematics in Science and Technology - International Conference on Mathematical Appplications in Science and Technology", CRC Press, 2025 Publication	1%
5	mdpi-res.com Internet Source	1%
6	ijritcc.org Internet Source	1%
7	Submitted to Fiji National University Student Paper	1%
8	H.L. Gururaj, Francesco Flammini, S. Srividhya, M.L. Chayadevi, Sheba Selvam. "Computer Science Engineering", CRC Press, 2024 Publication	1%
9	Submitted to First City University College Student Paper	1%
10	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial	1%