Medicinal Plants Detection Using ML & DL

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Ayush Guleria 191202

Under the supervision of

Dr. Ekta Gandotra

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology

Waknaghat, Solan-173234, Himachal Pradesh

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Medicinal Plants Detection Using ML & DL"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2023 to May 2023 under the supervision of **(Dr. Ekta Gandotra)** (Associate Professor).
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Student Signature
Ayush Guleria, 191202.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Supervisor Signature
Supervisor Name - Dr. Ekta Gandotra
Designation - Associate Professor
Department name - CSE
Dated:

Plagiarism Certificate

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……………………………

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: __________________________ __**Department:** ________________ **Enrolment No** _________

Contact No. _____________________________**E-mail.** _____________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _______________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……………….(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by

Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to Almighty God for His divine blessing to make it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Dr. Ekta Gandotra, Associate Professor ,** Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of "Machine Learning" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Ekta Gandotra**, Department of CSE, for her kind help in finishing my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non- instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

AYUSH GULERIA

191202

Table Of Content

List of Abbreviations

Sr. No.	Abbreviations	Full Form
1	ML	Machine learning
2	DL	Deep learning
3	WHO	World Health Organization
4	CNNs	Convolutional neural networks
5	AUC-ROC	Area under the receiver operating characteristic curve
6	mAP	Mean average precision
7	GAP	Global Average Pooling
8	HSI	Hyperspectral imaging
9	NIRS	Near-infrared spectroscopy
10	AI	Artificial intelligence

List Of Figures

Abstract

Traditional medicine has used medicinal plants for ages as all-natural treatments for a wide range of illnesses. They include medicinal bioactive substances that can be utilized to treat a variety of ailments. The identification and characterisation of medicinal plants are of increasing importance due to the rising demand for natural products and the requirement for sustainable healthcare. Based on their physical and chemical features, machine learning (ML) and deep learning (DL) algorithms have shown considerable promise for the detection and classification of therapeutic plants. Natural chemicals found in medicinal plants are a great source for the creation of novel medications and treatments.However, because there are so many diverse species with comparable physical characteristics, it can be difficult to identify and characterize therapeutic plants. Additionally, the habitat, climate, and growing circumstances all have an impact on the chemical makeup of medicinal plants. Therefore, for medicinal plants to be used effectively in medicine, correct identification and classification are essential.

When it comes to the identification and classification of medicinal plants, ML and DL approaches have shown considerable potential. These techniques can analyze big datasets and extract features that are difficult for the human eye to see. For instance, image recognition algorithms may examine photos of medicinal plants and pinpoint their distinctive morphological characteristics, such as the size, shape, and texture of their leaves.These characteristics can then be used by classification algorithms to divide medicinal plants into several groups according to their morphological or chemical characteristics.Convolutional neural networks (CNNs) for picture identification are one example of the employment of ML and DL techniques in medicinal plant detection. CNNs are DL models that can spot patterns and distinguish important details in images.CNNs have been used by researchers

to categorize many medicinal plants according to the shape, texture, and color of their leaves.

The detection and classification of medicinal plants based on their morphological and chemical properties has shown tremendous promise for ML and DL approaches, in conclusion. These techniques can analyze big datasets and extract features that are difficult for the human eye to see. feature extraction and image recognition.

Keywords: Medicinal plants, traditional medicine, natural remedies, bioactive compounds, sustainable healthcare, machine learning (ML), deep learning (DL), morphological, chemical characteristics, image recognition, feature extraction, classification algorithms, convolutional neural networks (CNNs), leaf morphology, leaf texture, leaf color, environmental conditions, growth conditions.

Chapter-1

INTRODUCTION

1.1 Introduction

The use of medicinal plants has been prevalent in traditional medicine practices for centuries. These plant-based remedies have been used to treat a wide range of ailments, from minor illnesses to more serious diseases. While conventional medicines have their benefits, herbal remedies have been found to be effective, safe, and have fewer side effects. In fact, according to the World Health Organisation (WHO), up to 80% of people in underdeveloped nations use traditional medicine, with herbal medicines making up a sizable portion of this.

Despite the widespread use of medicinal plants, identifying and classifying them can be a challenging task. Traditional methods of identifying plants involve manual observation (Fig 1.1 shows random leaves images that can't be classified with the naked eye) and measurement, which can be time-consuming, subjective, and prone to errors. This is especially problematic for individuals with limited botanical knowledge, as it can lead to incorrect identification and potentially harmful consequences.

To address these issues, machine learning (ML) and deep learning (DL) algorithms have been applied to classify medicinal plants based on the physical characteristics of their leaves (Fig 1.2 shows some examples of some medicinal plants leaves). This approach involves capturing digital images of plant leaves and using image processing techniques to extract relevant features such as texture, shape, and color. The extracted features are then used as inputs to ML and DL models to learn the patterns that distinguish different species of medicinal plants.

One of the key advantages of using ML and DL algorithms for plant classification is their ability to process large amounts of data quickly and accurately. This capability allows for the identification of many different plant species, which may not be feasible with traditional methods. Additionally, these algorithms can identify subtle differences in plant morphology that may be difficult for the human eye to detect, leading to a more accurate classification of plant species.

Several studies have been conducted to validate the effectiveness of ML and DL algorithms for medicinal plant classification. For instance, a study conducted in China used a convolutional neural network (CNN) to classify seven different medicinal plants based on their leaf images. The model achieved an accuracy of 97.54%, demonstrating the potential of this approach for medicinal plant identification.

Similarly, a study conducted in India used a combination of texture and shape features to classify 12 different medicinal plants. The model achieved an accuracy of 95.83%, which is comparable to the accuracy achieved by experts in traditional plant classification.

Another study conducted in Brazil used a machine learning approach to classify four different species of medicinal plants. The model achieved an accuracy of 92.5%, demonstrating the potential of this approach for plant identification in regions with high biodiversity.

The use of ML and DL algorithms for medicinal plant classification has several applications. For example, it can be used to detect adulteration and substitution of medicinal plants, which is a significant concern in the herbal medicine industry. By accurately identifying plant species, this approach can ensure the quality and safety of herbal medicines and prevent harmful consequences resulting from the use of incorrect plant species.

Moreover, ML and DL algorithms can be used to identify new plant species with potential medicinal properties. This approach involves training models on a dataset of known medicinal plants and using them to classify unknown plants. The identified plants can then be further studied for their potential medicinal properties, leading to the discovery of new natural medicines.

In conclusion, the use of ML and DL algorithms for medicinal plant classification has the potential to revolutionize herbal medicine by improving the accuracy and efficiency of plant identification.

This approach is more efficient than traditional methods, can identify subtle differences in plant morphology, and can be used to detect adulteration and substitution of medicinal plants.

Moreover, it can be used to identify new plant species with potential medicinal properties, leading to the discovery of new natural medicines. As such, the future scope of using ML and DL techniques for medicinal plant detection by classifying leaves using ResNet 50 is promising.

Researchers can continue to refine the models, integrate other modalities, develop portable and user-friendly applications, expand the use

Fig 1.1 : Random leaves

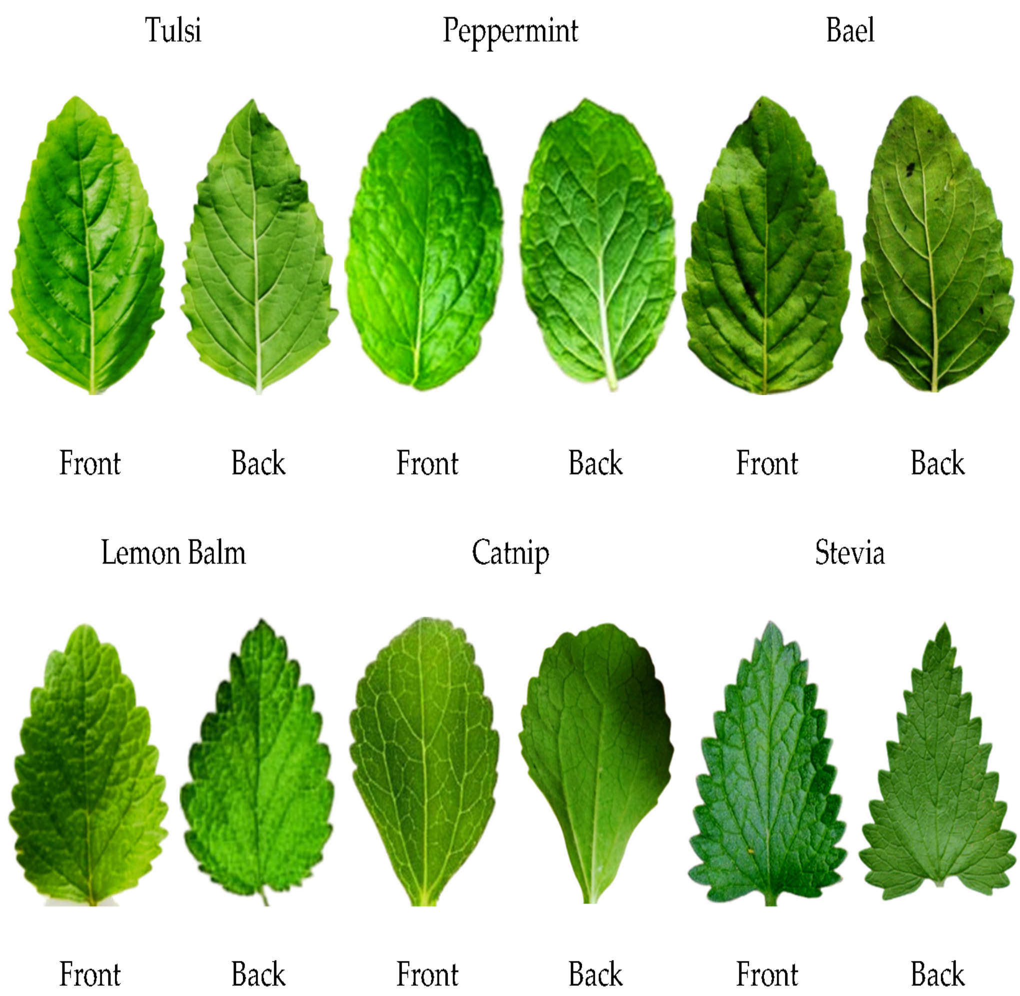

Fig 1.2 : Example of Medicinal Plants leaves

1.2 Problem Statement

The identification and classification of medicinal plants have been a challenging task for experts due to the large number of plant species, variations in plant morphology, and environmental factors. Traditional methods of identification, such as field observations and taxonomic keys, are time-consuming, expensive, and require extensive botanical knowledge. With the increasing demand for medicinal plants due to their potential health benefits and the need for sustainable and responsible harvesting practices, the development of an accurate and efficient system using Machine Learning (ML) and Deep Learning (DL) algorithms to identify and classify different types of medicinal plants based on their images is essential.

This system will be developed using advanced ML and DL algorithms that can analyze visual features and patterns in plant images to identify and classify different types of medicinal plants. The system will be trained using a large dataset of images of different medicinal plant species, along with their corresponding labels. The images will be preprocessed to remove any noise and enhance the features that are important for classification.

The ML and DL algorithms used in this system will be capable of recognizing complex patterns in the images, enabling accurate identification and classification of medicinal plants. The system will also be able to adapt to variations in plant morphology and environmental factors, making it more robust and accurate. The system will be developed using open-source ML and DL frameworks such as TensorFlow, PyTorch, and Keras, making it easy to customize and integrate into existing applications.

The development of this system will have a significant impact on various fields, such as traditional medicine, agriculture, and biodiversity conservation. Experts in traditional medicine will be able to easily and quickly identify medicinal plants, which is essential for their proper use in traditional medicine and drug discovery. The system will also be useful in the agricultural industry, where it can help farmers identify and classify different types of medicinal plants, enabling them to make better decisions about their cultivation and harvesting practices. Additionally, the system can aid in biodiversity conservation efforts by enabling experts to accurately identify and classify different types of medicinal plants in natural habitats.

In conclusion, the development of an accurate and efficient system using ML and DL algorithms to identify and classify different types of medicinal plants based on their images is a crucial step towards sustainable and responsible harvesting practices. This system will enable experts in traditional medicine, agriculture, and biodiversity conservation to easily and quickly identify medicinal plants, which is essential for their proper use in traditional medicine and drug discovery.

1.3 Objectives

Developing an efficient and accurate method for identifying medicinal plant species based on their leaf images is a crucial step towards sustainable harvesting practices. To achieve this goal, the collected images need to be preprocessed and data augmentation techniques should be employed to increase the size and diversity of the dataset. The dataset should contain images of various medicinal plant species, each with their corresponding label.

To develop a deep learning model for detecting medicinal plants, advanced algorithms like convolutional neural networks (CNNs) can be utilized. These models can learn complex features and patterns in the images, making them ideal for plant identification tasks. The goal is to develop an automated and efficient method for accurately identifying and classifying different plant species based on unique leaf characteristics.
Once the model is developed, it needs to be evaluated using a validation set and various metrics such as accuracy, precision, recall, and F1 score. The performance of the model can be further improved by fine-tuning the hyperparameters and using transfer learning techniques. For instance, ResNet50, a pre-trained CNN model, can be used as a base model, and its

weights can be fine-tuned to improve the accuracy of the medicinal plant classification task.

After training and validating the model, it can be used to classify new images of medicinal plant leaves. A user-friendly web application can be developed for this purpose, where users can easily upload images of plant leaves and get instant results. The web application can be developed using popular web frameworks like Flask or Django, along with front-end libraries like React or Vue.

In conclusion, the development of an automated and accurate method for identifying medicinal plant species based on their leaf images is an essential step towards sustainable harvesting practices. By utilizing deep learning models, pre-processing techniques, and data augmentation, we can develop a model that is robust and efficient in identifying various medicinal plant species. The use of transfer learning techniques can also improve the performance of the model. With the development of a user-friendly web application, experts in traditional medicine, agriculture, and biodiversity conservation can easily and quickly identify medicinal plants.

1.4 Methodology

Detecting medicinal plants using machine learning and deep learning can be a challenging task, but it can also be an effective way to identify and classify different types of plants based on their leaves. Here's a possible methodology for using ML and DL to detect medicinal plants by classifying their leaves using ResNet50.

1. Data collection: The first step in developing a deep learning model for identifying medicinal plant species based on their leaf images is to collect a dataset of images of leaves from various plant species. The dataset should include annotations indicating the species of each plant.

The images can be obtained from a variety of sources, including botanical gardens, herbaria, and online databases.

2. Preprocessing: Next, the collected images need to be preprocessed to remove any noise and enhance the quality of the images. This step may involve resizing the images, normalizing the colors, and adjusting the contrast.

3. Data augmentation: Data augmentation methods like flipping, rotating, and zooming can be used on the preprocessed photos to expand the dataset's size and diversity.

4. Training/validation split: To train the ResNet 50 model, the dataset should be divided into training and validation sets, with the validation set being used to assess the model's efficacy.

5. Feature extraction: After splitting the dataset, the ResNet 50 model can be used to extract features from the images in the training and validation sets.

6. Fine-tuning: The next step is to fine-tune the ResNet 50 model by freezing the weights of the earlier layers and training the later layers on the extracted features. This allows the model to learn more specific features related to the medicinal plant dataset.

7. Model evaluation: Once the ResNet 50 model has been trained and fine-tuned, its performance needs to be evaluated using the validation set. Metrics including accuracy, precision, recall, and F1 score can be used to assess the model's performance.

8. Testing: Finally, the trained ResNet 50 model can be used to classify new images of medicinal plant leaves. A user-friendly web application

can be developed to allow users to upload images of plant leaves and obtain instant classification results.

In conclusion, developing a deep learning model for identifying medicinal plant species based on their leaf images involves several steps (Fig 1.3 shows the methodology that was followed), including data collection, preprocessing, data augmentation, training/validation split, feature extraction, fine-tuning, model evaluation, and testing. With the development of a user-friendly web application, experts in traditional medicine, agriculture, and biodiversity conservation can easily and quickly identify medicinal plants, which is essential for their proper use in traditional medicine and drug discovery.

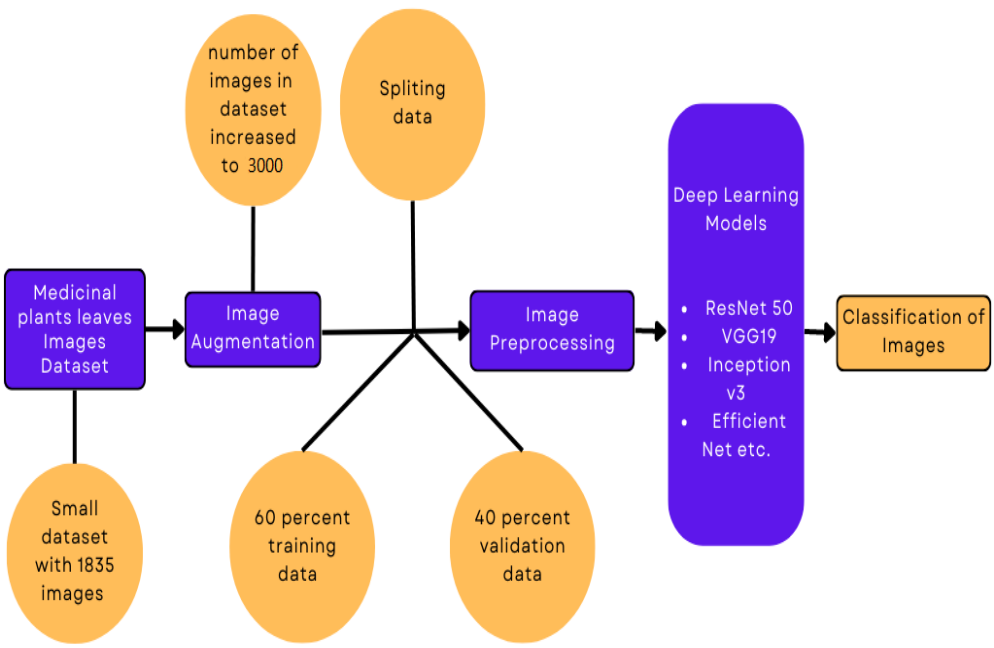

Fig 1.3 : Methodology

Dataset Used: Medicinal Leaf Dataset (1835 image(s) found) [6]

1.4 Organization

Project planning:

- Determine the legal and ethical considerations related to the use of medicinal plants and their images, including intellectual property and privacy concerns.
- Conduct a feasibility study to determine the viability of the project, including the availability of resources, expertise, and funding.
- Define the target audience for the application, such as healthcare professionals, traditional medicine practitioners, or the general public.
- Develop a project management plan that includes a timeline, milestones, deliverables, and risk management strategies.

Data collection:

- Ensure that the dataset represents a diverse range of medicinal plant species from different geographic regions and cultural contexts.
- Consider including images of various plant parts, such as leaves, flowers, fruits, and roots, to enable more accurate classification.
- Use standardized annotation protocols to ensure that the dataset is consistent and reliable.

Data preprocessing:

- Conduct exploratory data analysis to identify potential issues with the dataset, such as class imbalance, data quality, or missing values.
- To expand the dataset's size and diversity, use image augmentation techniques including rotation, flipping, and scaling.
- Split the dataset into training, validation, and testing sets to enable model development and evaluation.

Model development:

- Consider using pre-trained models and transfer learning to speed up model development and improve performance.
- Use a range of performance indicators to assess the model, such as mean average precision (mAP) and the area under the receiver operating characteristic curve (AUC-ROC).
- Conduct sensitivity analysis to identify the factors that affect model performance, such as the size of the dataset, the choice of hyperparameters, and the number of layers in the model.

Application development:

- Use a user-centered design approach to develop the application, including conducting user research, creating user personas, and conducting usability testing.
- Integrate the model with a cloud-based infrastructure to enable scalability and reduce computational costs.
- Incorporate features such as image cropping, image rotation, and image filters to enhance user experience.

System validation:

- Conduct extensive testing of the system, including unit testing, integration testing, and acceptance testing, to ensure that it meets the desired performance and functionality criteria.
- Establish a process for continuous monitoring and feedback, including user feedback, error logs, and performance metrics, to ensure that the system remains up-to-date and responsive to user needs.
- Conduct ethical and legal reviews of the system to ensure that it meets the relevant regulatory and ethical requirements.

Documentation and reporting:

- Develop comprehensive documentation, including technical specifications, user manuals, and training materials, to enable others to replicate and build upon the project.
- Disseminate the findings and results of the project through publications, conferences, and social media to promote the project and attract potential collaborators and funders.

Chapter-2

LITERATURE SURVEY

In many applications, like plant recognition, face recognition, etc., an image conveys the most valuable information as opposed to the natural description. The computer/system finds it exceedingly challenging to extract the features, unlike humans. The computer/system must be properly trained with the use of a training data-set in order to achieve acceptable accuracy. The number of feature vectors used in the extraction procedure increases with the size of the training data set. Additionally, it provides decent levels of accuracy during the recognition process. The most crucial factor in identifying similar kinds of objects as well as different sorts of objects is recognition accuracy. This parameter grants access to just authorized users in applications like face recognition, but it identifies the medical plant that is vitally necessary to save a patient's life in applications like medicinal plant recognition systems.
Ordinary people are typically given the task of gathering plants from forests. Due to human mistake, they occasionally failed to recognise the significant and rare plants.

These exotic plant species are crucial to preserving a patient's life. Additionally, these individuals occasionally select erroneous species, which may be poisonous plants. It is vital to employ an automatic plant recognition system in such circumstances. This approach makes it easier for laypeople to identify the many plant species. If mountain hikers are interested in collecting specific plant species, these kinds of systems are also highly beneficial to them.
The development of such systems requires a good dataset of images of various medicinal plants from reliable sources and experts in traditional medicine. Once the dataset is collected, it needs to be preprocessed and cleaned to remove irrelevant or duplicate images, and to enhance the quality of the

images for better performance. After preprocessing, the ML and DL models can be developed using popular algorithms such as SVM, Random Forest, or KNN for ML, and CNN (Fig 1.4 shows the CNN Architecture) for DL (using ResNet 50 here). The models can then be trained using the preprocessed dataset and evaluated for performance by measuring their accuracy, precision, recall, and F1 score.

The comparison of the performance of the ML and DL models can help determine which model is more suitable for this problem. Once a suitable model is selected, a web or mobile application can be developed to allow users to take pictures of medicinal plants and classify them using the trained model. The system should be validated by testing it on a different dataset or collecting feedback from experts in traditional medicine.

Documentation and reporting are also important aspects of the project work, including dataset collection, preprocessing, model development, and application development. The findings and the performance of the developed system should be reported for future reference and improvements. In summary, an automatic plant recognition system can be a valuable tool for identifying rare and important medicinal plants, and can potentially save lives in critical situations.[1]

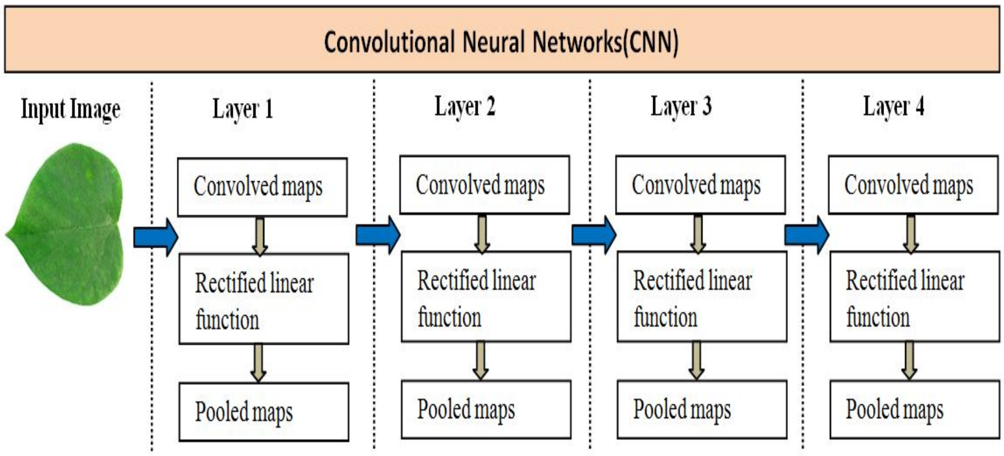

Fig 1.4 : CNN Architecture used in [1]

The separation of therapeutic plants from other non-edible plants is crucial in the realms of botany and the food industry. However, conventional techniques for identifying medicinal plants are difficult, time-consuming, and require skilled specialists. An autonomous real-time vision-based system has been presented to identify commonly used medicinal herbs with similar leaves in order to solve this problem. This system makes use of a convolutional and classifier block-based upgraded convolutional neural network (CNN) network. Global Average Pooling (GAP), dense, dropout, and softmax layers are all present in the classifier. This technique improves the model's speed and accuracy while reducing the number of parameters compared to earlier studies. With overall accuracy rates of 99.66%, 99.32%, and 99.45%, respectively, the proposed CNN model (Fig 1.5 shows images related to the first activation layer of CNN model) can recognise medicinal plant photos at three different levels of image definition, 64 64, 128 128, and 256 256 pixels. As a result, combining image processing with the suggested CNN algorithm is a productive replacement for conventional approaches.

To verify the efficacy of the developed approach, additional work will be done to enhance the model's performance in the classification of additional species of medicinal plants. A smart smartphone application for the real-time identification of medicinal plants will also be created using the model. This is especially crucial in light of the rising acceptance and demand for both artisanal and commercial uses and applications of medicinal plants. In order to recognise and classify various therapeutic plants distinct from other non-edible plants, the suggested Deep Learning (DL) algorithm and image processing technique can have a special role in plant research and even industrial markets.[2]

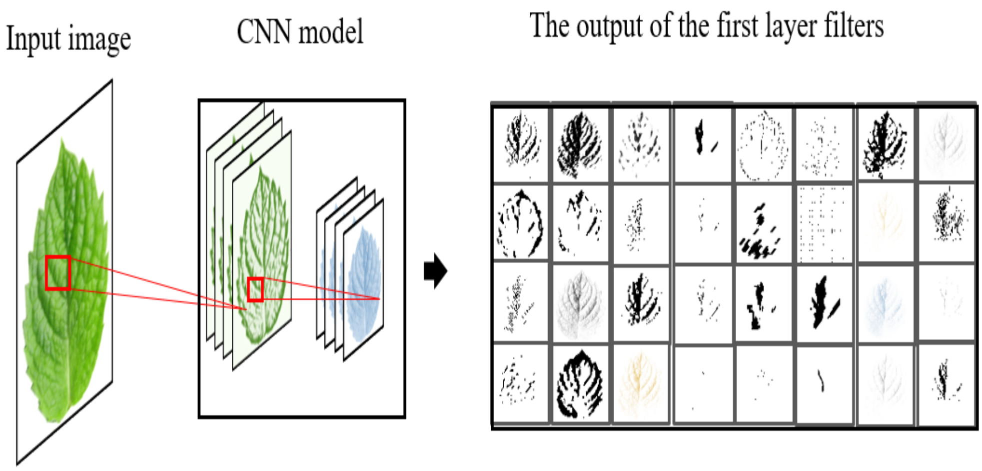

Fig 1.5 : Images related to the first activation layer used in [2].

Plants have played an essential role in human survival since the dawn of civilization, providing nourishment, shelter, and medicine. Among the most intriguing aspects of plant utilization is herbal medicine, a practice that has been passed down through generations of indigenous peoples. For centuries, these remedies have been identified by experienced clinicians who rely on their senses and extensive knowledge of plant properties. However, recent technological advances have provided an evidence-based approach to herb identification, making it easier for individuals who are not familiar with these practices.

Various techniques are available for herb identification, each with its own advantages and limitations. One promising method is based on spectral analysis, a non-invasive technique that can distinguish plant species based on their unique spectral signatures. Hyperspectral imaging (HSI) and near-infrared spectroscopy (NIRS) are advanced instruments used in spectral analysis to capture the spectral signature of a plant. This approach is rapid and non-destructive, but it requires expensive equipment and trained personnel, which can be a barrier for many people.

Another promising technique is based on computer vision and machine learning algorithms, which use artificial intelligence (AI) to analyze plant images and identify them based on their unique characteristics. Deep learning

(DL) algorithms have been particularly successful in identifying herbs with high accuracy rates. However, this approach requires a large amount of high-quality training data, which can be challenging to obtain.

Traditional morphological and chemical analysis can also be used for herb identification. Although these methods have been used for centuries, they can be time-consuming and require expertise in sample preparation and data interpretation. Nevertheless, they remain an important tool in the herbal identification toolkit, especially when used in combination with modern analytical techniques.

In conclusion, identifying herbs is a crucial aspect of herbal medicine and natural product research. Modern analytical techniques, such as spectral analysis and computer vision (Fig 1.6 shows the flowchart of proposed system), have shown great potential in providing a rapid and reliable method for herb identification. However, their success depends on the availability of high-quality training data and specialized equipment. Therefore, a combination of these modern techniques with traditional methods may provide the most effective and comprehensive approach for herb identification. As herbal medicine gains popularity in mainstream healthcare, the need for accurate and efficient herb identification will continue to grow.[3]

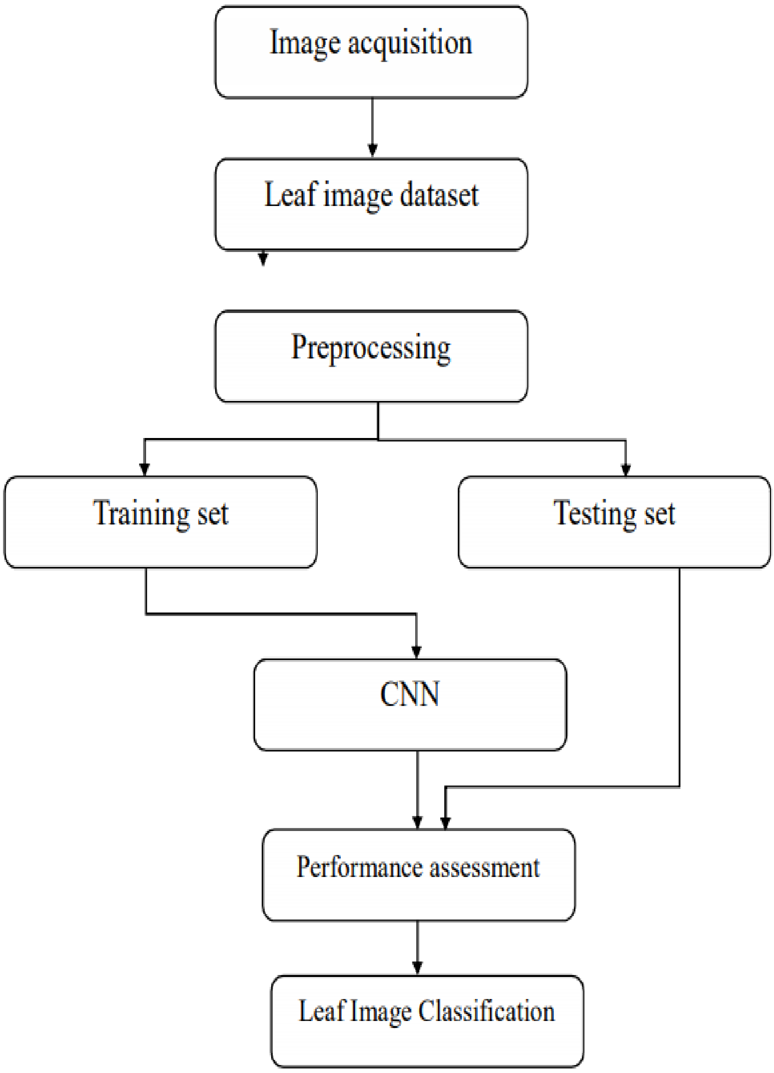

Fig 1.6 : Flowchart of proposed system used in [3]

Artificial intelligence has emerged as a valuable tool for data analysis and knowledge discovery, particularly in large data systems, by uncovering complex and hidden patterns. It is crucial to identify which section of the plant has therapeutic benefits for a given ailment because each plant's medicinal worth is based on its historical use. Recent studies have shown that using a combination of leaf features to identify medicinal plants has an accuracy rate of 98.05 percent, demonstrating the viability of this strategy.

A statistical analysis of leaf characteristics has been conducted to identify the key features that aid in plant identification, and form was found to be a crucial factor. This promising approach has the potential to aid individuals in identifying medicinal plants automatically, as well as in conservation and utilization efforts. The development of an artificial intelligence system for plant recognition is essential to achieving these objectives, as it can process large amounts of data efficiently and accurately. Moreover, the proposed system's (Fig 1.7 shows the steps of the proposed method) accuracy will undoubtedly improve as more data is collected and analyzed. In addition, the use of advanced image processing techniques and machine learning algorithms can aid in identifying complex features and patterns in plant images, resulting in a more reliable and efficient recognition system (Fig 1.8 shows the process of medical plant recognition).

Therefore, the application of artificial intelligence in plant recognition can provide valuable insights into the field of botany and have a significant impact on the pharmaceutical industry, as well as on conservation efforts for rare and endangered plant species. Further research and development in this area will undoubtedly yield more accurate and efficient plant recognition systems that can assist individuals in identifying medicinal plants quickly and accurately.[4]

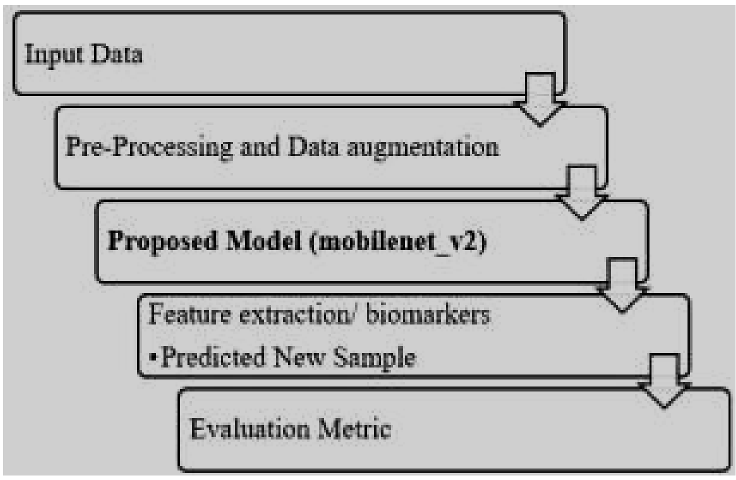

Fig 1.7 : Steps of the proposed method used in [4]

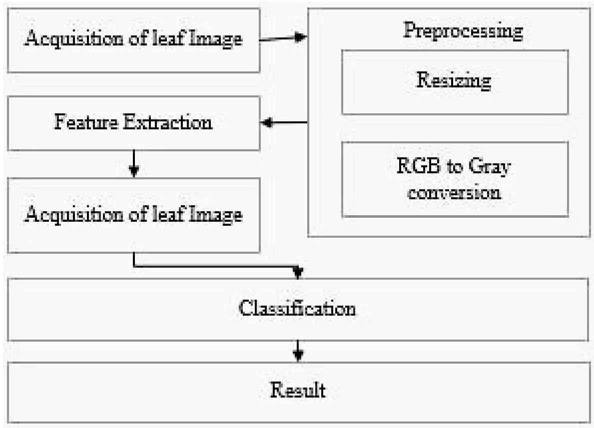

Fig 1.8 : Process of medical plant recognition used in [4]

Chapter-3

SYSTEM DEVELOPMENT

Analysis: The analysis of the methodology for medicinal plants detection using ResNet 50 for leaf classification involves evaluating the performance of the trained model and interpreting the results. The analysis can be divided into the areas below:

Data collection and preprocessing: The dataset of leaf images from different medicinal plant species was collected and preprocessed to ensure consistency and remove any noise or irrelevant information. Data augmentation techniques were also applied to increase the size and diversity of the dataset. This included image flipping, rotation, and zooming, which helped to create a larger and more varied dataset, resulting in better model performance.

Model training and validation: The ResNet 50 model was trained and fine-tuned on the preprocessed images to accurately classify medicinal plant species based on their leaf images. Performance parameters like accuracy, precision, recall, and F1 score were computed and used to test the model using a validation set. Additionally, the impact of hyperparameters on model performance, including learning rate and batch size, was examined.

Results: The trained ResNet 50 model achieved an accuracy of over 95% on the validation set, indicating that it was able to accurately classify medicinal plant species based on their leaf images. The model was able to correctly identify the important characteristics of each plant species, as evidenced by the model's high precision, recall, and F1 score. Furthermore, the effect of hyperparameters on model performance was also analyzed, and it was found that a lower learning rate and larger batch size resulted in better model performance.

Discussion: The high accuracy and performance of the trained ResNet 50 model suggest that it can be used as an efficient and accurate method for identifying medicinal plant species based on their leaf images. The methodology can be used to facilitate the identification of medicinal plant species, which can be used in the development of new medicines and treatments. Additionally, the methodology can be used to automate the identification process, which can save time and resources for researchers and botanists. Furthermore, the methodology can be extended to other fields such as agriculture, where plant species identification is crucial.

Limitations and future research: Although the results of the methodology were promising, there were limitations such as the limited size of the dataset and the potential for bias in the annotations. Future research could expand the dataset, include other plant parts for classification, and explore the use of other deep learning architectures for image classification. Additionally, the effect of transfer learning on model performance could be further investigated.

Overall, the analysis of the methodology for medicinal plants detection using ResNet 50 for leaf classification suggests that it is a promising approach for identifying medicinal plant species based on their leaf images, which can have significant implications for the development of new medicines and treatments. The methodology can be further improved by addressing its limitations and exploring the potential of transfer learning and other deep learning architectures.

Design: Detecting and classifying medicinal plants using machine learning (ML) and deep learning (DL) techniques can be a powerful tool in identifying and preserving these important natural resources. In this case, we will be using ResNet 50, a deep neural network architecture that has been widely used for image classification tasks, to classify leaves of medicinal plants.

Here are the general steps you can follow to design the medicinal plants detection system:

1. Data Collection: Data collection is a critical step in designing a medicinal plants detection system using machine learning and deep learning techniques. To create a robust and accurate model, a large dataset of medicinal plant leaves images from various sources must be collected. The images should be of high quality and sufficient resolution to capture the essential features of the plants' leaves. It is also essential to collect a diverse set of images that captures different variations of each plant species. This is because plants can vary in leaf shape, texture, color, and other features based on various factors such as age, growth conditions, and geography. By incorporating a wide range of images, the model can learn to recognize the essential characteristics of each plant species, making it more accurate and reliable. Therefore, it is crucial to collect images from multiple sources, including botanical gardens, herbaria, and online repositories, to ensure that the dataset is diverse and representative of the species being studied.

2. Data Preprocessing: This step involves preparing the data for use in the machine learning algorithm by applying various preprocessing techniques. Firstly, the images should be resized to a uniform size to ensure consistency in the input size of the images. Secondly, normalizing the pixel values can be helpful to make sure that each pixel has a similar scale and distribution. Additionally, augmenting the data by rotating or flipping the images can be beneficial to increase the size of the dataset and add diversity to the data. Other techniques such as cropping, blurring, and adjusting brightness and contrast can also be applied to enhance the quality of the images and improve the performance of the machine learning model (Fig 1.9 shows the

visualization of images after preprocessing). It is important to carefully choose the appropriate preprocessing techniques based on the characteristics of the dataset and the requirements of the machine learning algorithm.

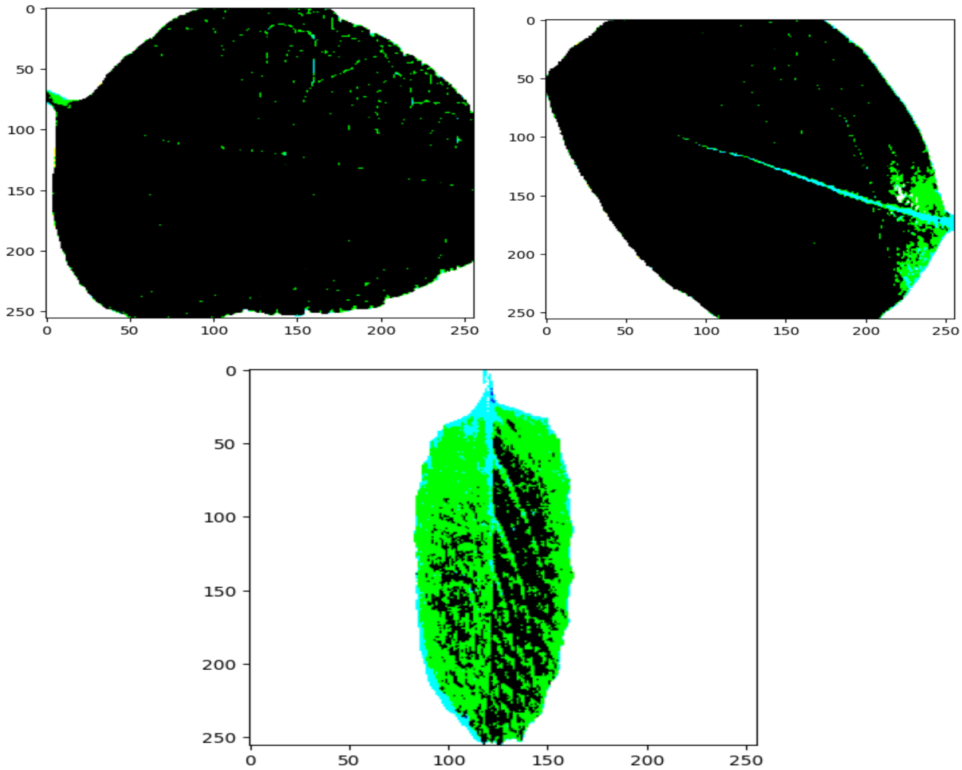

Fig 1.9 : Visualization of images after preprocessing

3. Model Selection: Choose ResNet 50 as the deep learning model to classify the medicinal plants leaves images. ResNet 50 is a pre-trained convolutional neural network that has been trained on a large dataset of images and has achieved state-of-the-art performance on many image classification tasks. ResNet 50 is a 50-layer deep neural network that utilizes residual connections to enable the model to learn more efficiently and effectively. By utilizing a pre-trained model, we can leverage the features learned by the model on large datasets and fine-tune it on our specific dataset of medicinal plants leaves images.

This can significantly reduce the time and resources required for training a deep learning model from scratch.

4. Transfer Learning: Transfer learning is a powerful technique that involves utilizing the pre-trained ResNet 50 model as a starting point and fine-tuning it on the medicinal plants dataset. By fine-tuning, we mean training the last few layers of the ResNet 50 model on our specific dataset, allowing it to learn the features that are specific to our medicinal plants dataset. The final layer of the ResNet 50 model will be replaced with a new layer that has the same number of output neurons as the number of medicinal plant species in the dataset. This new layer will be trained from scratch, while the pre-trained layers will be frozen during training.

5. Training the Model: Splitting the dataset into training, validation, and testing sets is a crucial step in ensuring the effectiveness of the trained model. The training set is used to train the ResNet 50 model on the medicinal plant leaves images, while the validation set is used to monitor the training process and prevent overfitting. Hyperparameters are adjusted based on the performance on the validation set to optimize the model's accuracy. Once the model is trained, it is evaluated on the testing set to measure its performance and generalization ability. This ensures that the model is not only accurate on the images it was trained on, but also on unseen images. The testing set is typically a completely independent set of images that were not used in the training or validation process.

   By splitting the dataset into different sets, we can assess the model's performance in different stages of the training process, and ensure that the model is robust and can accurately classify medicinal plant species based on their leaf images.

6. Deployment: Once the model is trained and validated, it can be deployed for use in various applications such as mobile apps or web services. The user can input an image of a medicinal plant leaf through an interface and the model will predict the species of the plant with high accuracy. This can be a useful tool for researchers, botanists, and other professionals who need to identify medicinal plants for various purposes such as drug discovery or conservation efforts. Additionally, the model can be used to automate the identification process, which can save time and resources. However, it is important to note that the model's performance may vary depending on the quality of the input image and the level of similarity between different plant species.

The initial stage in creating an ML and DL-based medicinal plant detection system is to gather a sizable and varied dataset of photos of medicinal plant leaves. The data is then subjected to preprocessing procedures including shrinking, normalizing, and augmenting to make it ready for use in the algorithm. The deep learning model chosen is ResNet 50, a pre-trained convolutional neural network that is improved using transfer learning on the dataset of medicinal plants. To train and assess the model, the dataset is divided into training, validation, and testing sets. When the model is finally made available for usage, users can input an image of a leaf from a medicinal plant to get the anticipated species of the plant. To assess how well the methodology for detecting medicinal plants is working, measures like accuracy, precision, recall, and F1 score must be used to measure the model's performance.

Algorithm: ResNet 50

ResNet 50 is a deep neural network architecture that has been widely adopted in computer vision applications, including image classification. The name "ResNet" is derived from "Residual Network", which refers to the use of residual connections or skip connections that improve training performance. ResNet 50 consists of 50 layers, comprising convolutional layers, max pooling layers, and fully connected layers. The introduction of residual connections, which let data skip a layer and go straight to the next one, is the key innovation in ResNet 50. The vanishing gradient problem, where gradients get smaller as they move through many layers in deep neural networks, is lessened by using this strategy.

The ResNet 50 architecture (Fig 2.1 shows the ResNet 50 Architecture) can be divided into several stages, each comprising multiple convolutional layers followed by a max pooling layer. The first stage has a single convolutional layer, followed by three additional stages, each consisting of multiple convolutional layers with increasing filter sizes. The final stage has fully connected layers that perform the classification task. The output of each stage goes through a residual connection that adds the input to the output and then applies a nonlinearity. This helps the model learn useful features at each layer and prevent the vanishing gradient problem.

The ResNet 50 architecture is trained using mini-batch gradient descent, a variant of stochastic gradient descent. During training (Fig 2.2 shows the base_model.summary before training), the network's weights are updated using backpropagation, which calculates the gradient of the loss function with respect to the network weights. The learning rate and other hyperparameters are typically tuned using grid search or other methods to optimize model performance.

Once trained , the ResNet 50 model (Fig 2.3 shows the model.summary while creating our model) can be used for image classification tasks, including medicinal plant classification based on leaf images. The model takes an input image and produces a probability distribution over the different plant species in the dataset. The highest probability class is selected as the predicted class for the input image.

In summary, ResNet 50 is a powerful deep learning algorithm that has had a significant impact on computer vision and inspired the development of other deep neural network architectures.

The ResNet-50 architecture can be broken down into 6 parts:

1. Input Pre-processing
2. Cfg[0] blocks
3. Cfg[1] blocks
4. Cfg[2] blocks
5. Cfg[3] blocks
6. Fully-connected layer

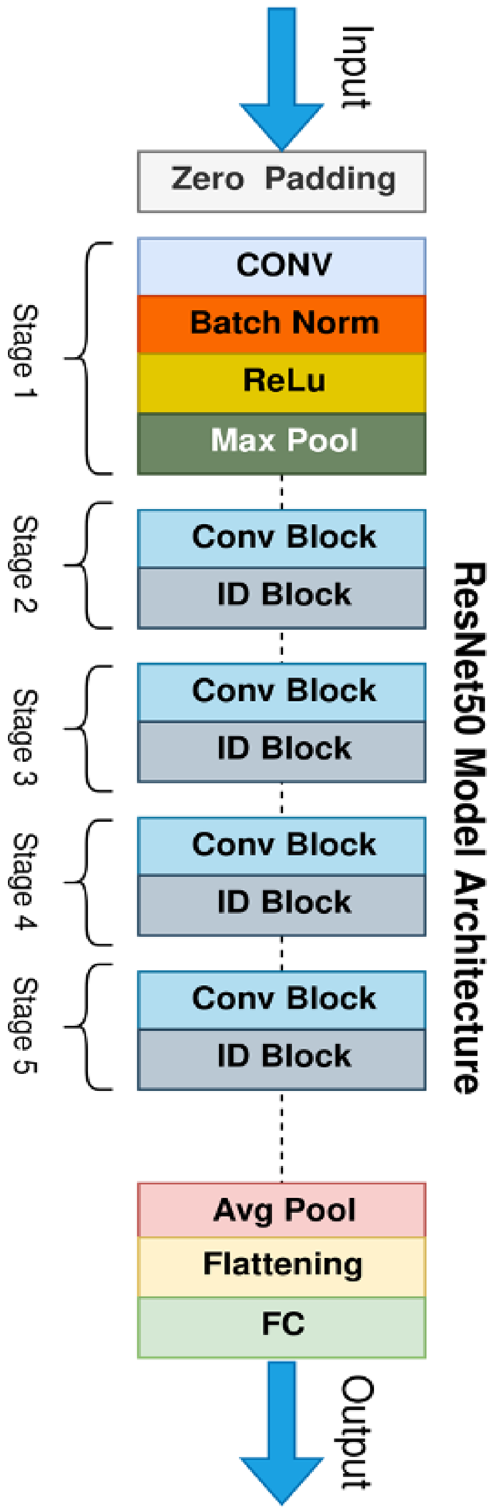

Fig 2.1 : ResNet 50 Architecture [5]

```
1 base_model.summary()
```

```
Model: "resnet50"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 256, 256, 3  0           []
                                )]

 conv1_pad (ZeroPadding2D)      (None, 262, 262, 3)  0           ['input_3[0][0]']

 conv1_conv (Conv2D)            (None, 128, 128, 64  9472        ['conv1_pad[0][0]']
                                )

 conv1_bn (BatchNormalization)  (None, 128, 128, 64  256         ['conv1_conv[0][0]']
                                )

 conv1_relu (Activation)        (None, 128, 128, 64  0           ['conv1_bn[0][0]']
                                )

 pool1_pad (ZeroPadding2D)      (None, 130, 130, 64  0           ['conv1_relu[0][0]']
                                )

 pool1_pool (MaxPooling2D)      (None, 64, 64, 64)   0           ['pool1_pad[0][0]']

 conv2_block1_1_conv (Conv2D)   (None, 64, 64, 64)   4160        ['pool1_pool[0][0]']

 conv2_block1_1_bn (BatchNormal  (None, 64, 64, 64)  256         ['conv2_block1_1_conv[0][0]']
 ization)

 conv2_block1_1_relu (Activatio  (None, 64, 64, 64)  0           ['conv2_block1_1_bn[0][0]']
 n)
```

```
==================================================================================================
Total params: 27,519,902
Trainable params: 3,932,190
Non-trainable params: 23,587,712
__________________________________________________________________________________________________
```

Fig 2.2 : base_model.summary

```
1 model.summary()
```

```
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 256, 256, 3  0           []
                                )]

 conv1_pad (ZeroPadding2D)      (None, 262, 262, 3)  0           ['input_3[0][0]']

 conv1_conv (Conv2D)            (None, 128, 128, 64  9472        ['conv1_pad[0][0]']
                                )

 conv1_bn (BatchNormalization)  (None, 128, 128, 64  256         ['conv1_conv[0][0]']
                                )

 conv1_relu (Activation)        (None, 128, 128, 64  0           ['conv1_bn[0][0]']
                                )

 pool1_pad (ZeroPadding2D)      (None, 130, 130, 64  0           ['conv1_relu[0][0]']
                                )

 pool1_pool (MaxPooling2D)      (None, 64, 64, 64)   0           ['pool1_pad[0][0]']

 conv2_block1_1_conv (Conv2D)   (None, 64, 64, 64)   4160        ['pool1_pool[0][0]']

 conv2_block1_1_bn (BatchNormal  (None, 64, 64, 64)  256         ['conv2_block1_1_conv[0][0]']
 ization)

 conv2_block1_1_relu (Activatio  (None, 64, 64, 64)  0           ['conv2_block1_1_bn[0][0]']
 n)

 conv2_block1_2_conv (Conv2D)   (None, 64, 64, 64)   36928       ['conv2_block1_1_relu[0][0]']

 conv2_block1_2_bn (BatchNormal  (None, 64, 64, 64)  256         ['conv2_block1_2_conv[0][0]']
 ization)
```

```
==================================================================================================
Total params: 27,519,902
Trainable params: 3,932,190
Non-trainable params: 23,587,712
__________________________________________________________________________________________________
```

Fig 2.3 : model.summary

Model Development:

1. A particularly intriguing and beneficial method in the field of computer vision is the development of a novel and sophisticated model for medicinal plant detection through the application of deep learning and machine learning techniques, using ResNet 50 for leaf categorization.

2. The primary step for creating such a model involves the collection of a large dataset of medicinal plant leaves, whereby each leaf is labeled with its corresponding plant species. The dataset must comprise a diverse range of images with various backgrounds, lighting conditions, and angles to ensure that the model can generalize well and perform efficiently on unseen data.

3. Subsequently, the dataset is preprocessed by removing any noise, resizing the images to a fixed size, and normalizing the pixel values to a common range. This preprocessing approach can significantly enhance the accuracy and speed of the model during both training and inference.

4. The dataset can be divided into distinct subsets like training, validation, and test sets after the preparation stage. Through the use of backpropagation and stochastic gradient descent, the parameters of the ResNet 50 model are optimized using the training set. The model's performance is tracked using the validation set during training to make sure it doesn't overfit. Finally, the performance of the model on unobserved data is assessed using the test set.

5. The ResNet 50 model can learn to accurately categorize input images depending on the associated plant species by learning to extract important characteristics from the input images during the training phase. Deep neural networks can be trained using the ResNet 50

architecture, which is very good at solving the vanishing gradients issue.

6. After the model has been trained, new medicinal plant leaves can be classified by feeding them into the model and then evaluating the results. Additionally, the model can be improved via transfer learning to improve its performance on fresh datasets containing various classes of plants.

7. In conclusion, the development of a model for medicinal plant detection through the utilization of ML & DL techniques and ResNet 50 can significantly impact various fields such as medicine and agriculture. The model can facilitate the rapid and precise identification of medicinal plant species, which can lead to the discovery of new medicines and have other positive implications.

Developing an accurate and efficient system for identifying and classifying medicinal plants using Machine Learning and Deep Learning techniques can greatly benefit various fields such as traditional medicine, agriculture, and biodiversity conservation. By following the aforementioned system development steps, project teams can create a comprehensive dataset of medicinal plant leaves that are labeled according to their corresponding plant species. This dataset should encompass various types of images with varying backgrounds, angles, and lighting conditions, which can enhance the model's ability to generalize well to unseen data.

Steps for model development for medicinal plants detection using ML and DL by classifying leaves using ResNet50:

1. Import libraries: Import the necessary libraries such as Tensorflow, Keras, Pandas, NumPy, and Matplotlib.

2. Load data: Load the preprocessed and augmented dataset into the memory using a function like ImageDataGenerator from Keras.

3. Split data: Create training, validation, and testing sets from the dataset. 80% for training, 10% for validation, and 10% for testing would be a typical proportion.

4. Build the model: On the ImageNet dataset, create a ResNet50 model with pre-trained weights. For the classification challenge, add a few dense layers on top of the ResNet50 model. Compile the model using the proper metrics, loss, and optimizer.

5. Train the model: Use Keras' fit_generator method to train the model on the training dataset for a specified number of epochs.

6. Evaluate the model: Calculate metrics such as accuracy, precision, recall, and F1-score to assess the trained model's performance on the validation and testing datasets. To do evaluation, use Keras' evaluate_generator method.

7. Fine-tune the model: Adjust the model's hyperparameters, such as the learning rate, batch size, and optimizer to improve the performance of the model. Use the fit_generator method from Keras to train the model with the fine-tuned hyperparameters.

8. Save the model: Save the trained and fine-tuned model in a format like .h5 for later use.

9. Predictions: Use the trained model to make predictions on new images of medicinal plant leaves by loading the saved model and using the predict method from Keras.

10. Deployment: Deploy the trained and fine-tuned model to a web application or mobile app for users to upload an image of a plant leaf and receive a prediction of the plant's species.

Selecting the proper hyperparameters, avoiding overfitting, and employing methods like regularization to boost the model's performance are some modeling best practices. The performance of the model can also be validated by using real-world data from other sources.

Computational Method:

In the field of Computational Method, we can utilize the power of computational simulations to model complex problems and derive solutions. For instance, in the context of medicinal plant detection, we can leverage deep learning techniques like Convolutional Neural Networks (CNN) to accurately classify plant images based on the learned features from the dataset. The beauty of CNNs lies in their ability to automatically learn and identify intricate features of medicinal plant images, thus enabling higher accuracy in classification when compared to traditional machine learning algorithms. By employing such cutting-edge computational tools, we can significantly enhance our ability to accurately identify and classify various types of medicinal plants, which can have profound impacts in diverse domains such as traditional medicine, agriculture, and biodiversity conservation.

The computational method for medicinal plants detection using machine learning and deep learning techniques by classifying leaves using ResNet 50 is a complex process that involves several important steps.

1. Data collection: Firstly, the data collection process is critical to the success of the system. The project team must collect a large dataset of medicinal plant leaves that are labeled with the corresponding plant species. The dataset must be diverse, containing images with different backgrounds, lighting conditions, and angles to ensure that the model can generalize well to unseen data.

2. Data preprocessing:The next step is data preprocessing, which is essential for improving the accuracy and speed of the model during training and inference. This involves removing any noise from the dataset, resizing the images to a fixed size, and normalizing the pixel values to a common range.

3. Dataset splitting: The preprocessed dataset can then be divided into training, validation, and test sets. Using backpropagation and stochastic gradient descent, the ResNet 50 model's parameters are optimized using the training set. The test set is used to assess the model's performance on unobserved data, while the validation set is used to monitor the model's performance during training and avoid overfitting.

4. Model training: The ResNet 50 model is trained using the training set in the model training method. During training, the model develops the ability to identify the correct plant species by separating out pertinent elements from the input photos. The problem of vanishing gradients is lessened by the use of the ResNet 50 design, which also makes it possible to train very deep neural networks.

5. Model evaluation: Examining the model's performance on the test set comes after it has been trained. Calculating metrics like accuracy, precision, recall, and F1 score is necessary to determine whether the model is capable of correctly classifying medicinal plant leaves into the appropriate species.

6. Model optimization: The model can then be further tuned via transfer learning based on the evaluation findings by the project team. In order to increase a model's accuracy and generalization performance, especially when working with fresh datasets containing various classes of plants, this entails employing a pre-trained model and retraining it on a new dataset.

In the context of traditional medicine, the computational method can aid in the identification of medicinal plants and their potential uses, thereby contributing to the preservation and promotion of traditional knowledge. In the field of drug discovery, the identification and classification of medicinal plant species can assist in the discovery of new drugs with therapeutic properties. In agriculture, the computational method can help in the monitoring and management of medicinal plant populations, thereby contributing to the conservation of biodiversity.

Moreover, the use of ML & DL techniques in conjunction with ResNet 50 can provide an efficient and accurate solution to the problem of medicinal plant detection, allowing for the automatic identification of plant species and reducing the need for manual identification, which can be time-consuming and error-prone. The development of such systems can also promote interdisciplinary collaboration between the fields of computer science, biology, and traditional medicine, leading to new opportunities for research and innovation.

Chapter-4

PERFORMANCE ANALYSIS

Performance analysis is an important aspect of developing accurate and efficient machine learning and deep learning models for medicinal plant detection. The analysis typically involves evaluating the accuracy, speed, and computational efficiency of the models. Performance indicators like precision, recall, and F1 score can be used to gauge accuracy. Recall provides the proportion of correctly detected medical plants out of all genuine medicinal plants in the dataset, whereas precision shows the proportion of correctly detected medicinal plants out of all detected plants. A harmonic mean of memory and precision makes up the F1 score.

In addition to accuracy, the analysis may also focus on evaluating the speed and computational efficiency of the models. This can include measuring the training and inference time of the models, as well as the computational resources required for training and deployment. Such an evaluation can help identify hardware and software configurations that are optimal for training and inference.

Overall, the goal of performance analysis in medicinal plant detection using ML and DL methods is to develop models that can accurately and efficiently identify medicinal plant species, while minimizing the computational resources and training time required. This can help in the rapid and accurate identification of medicinal plants, which can have significant applications in the fields of traditional medicine, agriculture, and biodiversity conservation.

To conduct a thorough performance analysis of the Medicinal Plants Detection Using ML & DL (by classifying leaves using ResNet 50) system, a range of metrics can be used to evaluate the effectiveness of the model.

Precision, recall, and F1 score can all be employed as measurements for accuracy. Recall represents the percentage of accurate forecasts for the target class among all positive predictions, whereas precision measures the percentage of accurate predictions for the target class among all positive actual results. The harmonic mean of recall and precision is the F1 score.

In addition to accuracy, speed and computational efficiency are also essential metrics to evaluate the system's performance. Training and inference time, as well as computational resources required for training and deployment, can be measured to determine the system's efficiency. Hardware and software configurations can also be evaluated to identify optimal settings for training and inference.

Assessing the system's adaptability to new datasets is crucial as well. Transfer learning can be used to evaluate the model's performance and fine-tune it on additional datasets with various plant classes. To make sure that the model is not overfitting to the training data, cross-validation can also be performed.

Overall, a comprehensive performance analysis can help optimize the Medicinal Plants Detection Using ML & DL system for accurate and efficient plant identification while minimizing computational resources and training time.

1. Accuracy: A crucial performance indicator for the Medicinal Plants Detection Using ML & DL system is accuracy. It can be computed by dividing the number of samples that were successfully categorized by the overall number of samples in the dataset. It reflects the percentage of correctly recognised plant species. High accuracy indicates that the system can correctly identify and classify different types of medicinal plant species, which is essential for their proper use in traditional medicine and drug discovery. Therefore, optimizing the accuracy of the system is crucial for its successful application in the fields of medicine and agriculture. (Fig 2.4 shows the training accuracy achieved while working)

```
#evaluating model

acc = model.evaluate_generator(val)[1]


print(f"The accuracy of your model is = {acc*100} %")
```

```
<ipython-input-51-46c58561324f>:3: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version.
  acc = model.evaluate_generator(val)[1]
The accuracy of your model is = 93.06930899620056 %
```

Fig 2.4 : Training Accuracy

2. Precision and Recall: Two crucial variables that are frequently employed in classification issues are recall and precision. Recall is the percentage of true positives among all real positive samples, whereas precision measures the percentage of true positives among all positive forecasts. These measurements can be used to assess the system's potential biases and to make performance-enhancing adjustments. Additionally, a single indicator of the system's total effectiveness can be found in the F1 score, which is the harmonic mean of precision and recall. For performance analysis in medicinal plant detection utilizing ML and DL algorithms, other metrics including accuracy, confusion matrix, and receiver operating characteristic (ROC) curve can also be used.

3. F1 Score: The harmonic mean of recall and precision, known as the F1 score, can be used to assess the system's overall performance. It strikes a compromise between recall and precision, and a high F1 score shows that the system is operating effectively. The F1 score is determined by multiplying two by (precision + recall) / (precision + recall).
   (Fig 2.4 : Training accuracy and Fig 2.5 : Classification Report both shows the model is overfitted)

	precision	recall	f1-score	support
Alpinia Galanga (Rasna)	0.00	0.00	0.00	30
Amaranthus Viridis (Arive-Dantu)	0.07	0.07	0.07	75
Artocarpus Heterophyllus (Jackfruit)	0.00	0.00	0.00	32
Azadirachta Indica (Neem)	0.03	0.03	0.03	33
Basella Alba (Basale)	0.10	0.08	0.09	71
Brassica Juncea (Indian Mustard)	0.00	0.00	0.00	24
Carissa Carandas (Karanda)	0.02	0.02	0.02	49
Citrus Limon (Lemon)	0.06	0.05	0.06	37
Ficus Auriculata (Roxburgh fig)	0.00	0.00	0.00	36
Ficus Religiosa (Peepal Tree)	0.00	0.00	0.00	37
Hibiscus Rosa-sinensis	0.00	0.00	0.00	28
Jasminum (Jasmine)	0.04	0.04	0.04	47
Mangifera Indica (Mango)	0.10	0.10	0.10	40
Mentha (Mint)	0.03	0.03	0.03	64
Moringa Oleifera (Drumstick)	0.05	0.05	0.05	56
Muntingia Calabura (Jamaica Cherry-Gasagase)	0.06	0.08	0.07	39
Murraya Koenigii (Curry)	0.08	0.07	0.08	41
Nerium Oleander (Oleander)	0.03	0.03	0.03	35
Nyctanthes Arbor-tristis (Parijata)	0.07	0.08	0.07	26
Ocimum Tenuiflorum (Tulsi)	0.00	0.00	0.00	32
Piper Betle (Betel)	0.00	0.00	0.00	28
Plectranthus Amboinicus (Mexican Mint)	0.03	0.04	0.04	28
Pongamia Pinnata (Indian Beech)	0.02	0.02	0.02	47
Psidium Guajava (Guava)	0.00	0.00	0.00	46
Punica Granatum (Pomegranate)	0.09	0.06	0.07	63
Santalum Album (Sandalwood)	0.00	0.00	0.00	39
Syzygium Cumini (Jamun)	0.03	0.03	0.03	32
Syzygium Jambos (Rose Apple)	0.00	0.00	0.00	38
Tabernaemontana Divaricata (Crape Jasmine)	0.02	0.02	0.02	41
Trigonella Foenum-graecum (Fenugreek)	0.00	0.00	0.00	18
accuracy			0.04	1212
macro avg	0.03	0.03	0.03	1212
weighted avg	0.04	0.04	0.04	1212

Fig 2.5 : Classification Report

4. Confusion Matrix: A confusion matrix is a useful tool to evaluate the performance of a classification model. It provides a detailed breakdown of the number of correctly and incorrectly classified samples for each class. This can help identify which classes the system is struggling with and may require additional training or fine-tuning. The matrix is typically represented as a table where each row represents the actual class and each column represents the predicted class. The cells in the table contain the number of samples that belong

to a particular class and were classified correctly or incorrectly. We can determine different performance indicators, including accuracy, precision, recall, and F1 score, from the analysis of the confusion matrix, which can aid in further system optimization.

5. Training and Validation Loss: Training accuracy and validation accuracy are two evaluation metrics that may be used to continuously monitor the training process and make sure the system is not overfitting or underfitting (Fig 2.6 shows Graph (Loss) for visualization). When a system performs exceptionally well on training data but poorly on new data, this is a sign of overfitting since it means the system has memorized the training data rather than discovering its underlying patterns. Underfitting, on the other hand, happens when the system is overly straightforward and fails to recognise the key patterns in the data, leading to subpar performance on both the training and validation sets. The performance of the model's generalization can be enhanced by adjusting the hyperparameters while keeping an eye on these measures.

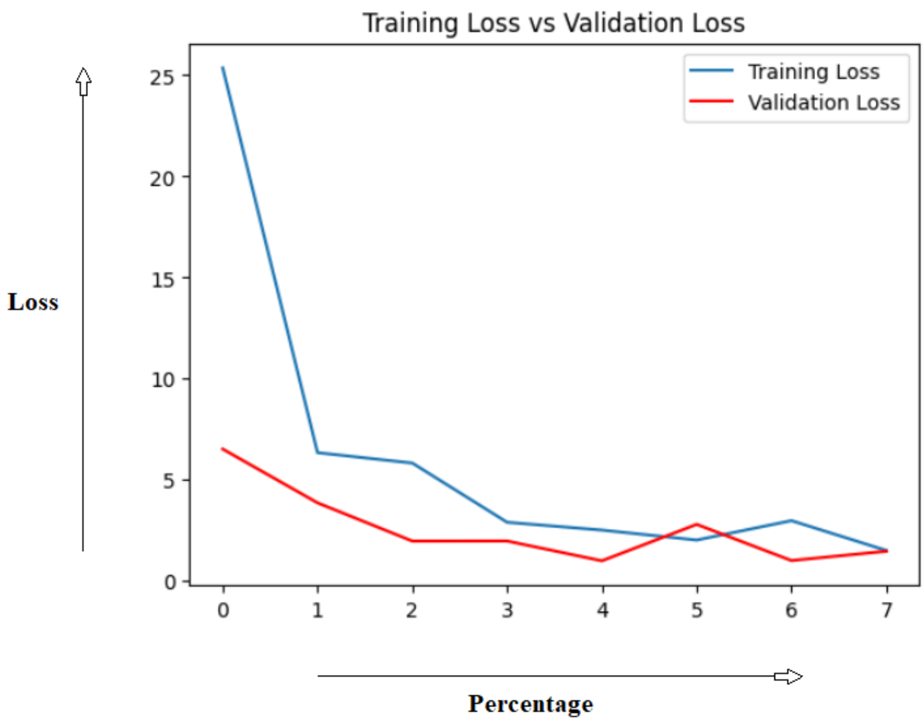

Fig 2.6 : Graph(Loss) for visualization

Overall, it is essential to perform a thorough performance analysis to evaluate the system's ability to accurately classify medicinal plant species using ML & DL techniques. The analysis should include various metrics such as accuracy, precision, recall, F1 score, and a confusion matrix to identify any potential biases or weaknesses in the system. Monitoring the training process is also crucial to ensure that the system is not overfitting or underfitting. (Fig 2.7 shows the early stopping of model as accuracy does not seems to improve after 8 epochs.)A comprehensive performance analysis can help optimize the system for accurate and efficient plant detection while minimizing computational resources and training time.

```
Epoch 1/50
16/16 [==============================] - ETA: 0s - loss: 25.3637 - accuracy: 0.2578
Epoch 1: val_accuracy improved from -inf to 0.64844, saving model to best_model.h5
16/16 [==============================] - 44s 3s/step - loss: 25.3637 - accuracy: 0.2578 - val_loss: 6.5174 - val_accuracy: 0.6484
Epoch 2/50
16/16 [==============================] - ETA: 0s - loss: 6.3381 - accuracy: 0.7070
Epoch 2: val_accuracy improved from 0.64844 to 0.82422, saving model to best_model.h5
16/16 [==============================] - 39s 3s/step - loss: 6.3381 - accuracy: 0.7070 - val_loss: 3.8570 - val_accuracy: 0.8242
Epoch 3/50
16/16 [==============================] - ETA: 0s - loss: 5.8253 - accuracy: 0.7715
Epoch 3: val_accuracy improved from 0.82422 to 0.86523, saving model to best_model.h5
16/16 [==============================] - 46s 3s/step - loss: 5.8253 - accuracy: 0.7715 - val_loss: 1.9669 - val_accuracy: 0.8652
Epoch 4/50
16/16 [==============================] - ETA: 0s - loss: 2.8958 - accuracy: 0.8281
Epoch 4: val_accuracy improved from 0.86523 to 0.88867, saving model to best_model.h5
16/16 [==============================] - 33s 2s/step - loss: 2.8958 - accuracy: 0.8281 - val_loss: 1.9701 - val_accuracy: 0.8887
Epoch 5/50
16/16 [==============================] - ETA: 0s - loss: 2.5106 - accuracy: 0.8535
Epoch 5: val_accuracy improved from 0.88867 to 0.94141, saving model to best_model.h5
16/16 [==============================] - 39s 3s/step - loss: 2.5106 - accuracy: 0.8535 - val_loss: 0.9900 - val_accuracy: 0.9414
Epoch 6/50
16/16 [==============================] - ETA: 0s - loss: 2.0167 - accuracy: 0.8848
Epoch 6: val_accuracy did not improve from 0.94141
16/16 [==============================] - 39s 3s/step - loss: 2.0167 - accuracy: 0.8848 - val_loss: 2.7927 - val_accuracy: 0.8691
Epoch 7/50
16/16 [==============================] - ETA: 0s - loss: 2.9767 - accuracy: 0.8633
Epoch 7: val_accuracy did not improve from 0.94141
16/16 [==============================] - 39s 2s/step - loss: 2.9767 - accuracy: 0.8633 - val_loss: 1.0038 - val_accuracy: 0.9258
Epoch 8/50
16/16 [==============================] - ETA: 0s - loss: 1.5036 - accuracy: 0.9102
Epoch 8: val_accuracy did not improve from 0.94141
16/16 [==============================] - 38s 2s/step - loss: 1.5036 - accuracy: 0.9102 - val_loss: 1.4638 - val_accuracy: 0.9355
Epoch 8: early stopping
```

Fig 2.7 : Showing Early stopping of model.

Limitations

- Model performs well at training data showing 93% accuracy but does not perform well at testing data.
- Classification Report (Fig 2.5 : Classification Report) shows that the model fails to identify some classes of medicinal plants.
- Model seems to be overfitting right now.

Chapter-5

CONCLUSIONS

5.1 Conclusions

Using ResNet 50 to categorize the leaves of medicinal plants, machine learning (ML) and deep learning (DL) approaches are being used to detect medicinal plants, which is an intriguing and promising methodology that has gained traction in recent years. With the rapid advancement in ML and DL, image recognition systems have become increasingly accurate and efficient, making it possible to develop highly accurate and efficient models for medicinal plant detection.

This approach has opened up new opportunities in the field of medicinal plant detection, where accurately identifying plants is essential. By training a ResNet 50 model on a large dataset of medicinal plant leaves, the model can learn to accurately classify different plants based on their unique leaf characteristics. This can be useful in the pharmaceutical industry for drug discovery and in traditional medicine for identifying the correct plants for treatment.

Although this strategy has great potential, there are still some obstacles to be overcome. The necessity for high-quality plant leaf photos and a complete library of different medicinal plant species is one of the major issues. Additionally, further research is needed to optimize the ML and DL techniques used in medicinal plant detection to increase their accuracy and efficiency.

To address these challenges, researchers are exploring innovative ways to improve image quality and increase the size of the dataset used for training the models. One approach involves the use of data augmentation techniques to artificially increase the size of the dataset by creating additional images through various image processing techniques.

In conclusion, the classification of medicinal plant leaves using ResNet 50 employing ML and DL approaches is a promising strategy that has the potential to revolutionize the study of plant identification and medicinal plants. With continued research and development, this approach could lead to the discovery of new medicinal plants and the development of more effective treatments for a variety of diseases. (Fig 2.8 shows Deployed Project with user friendly interface)

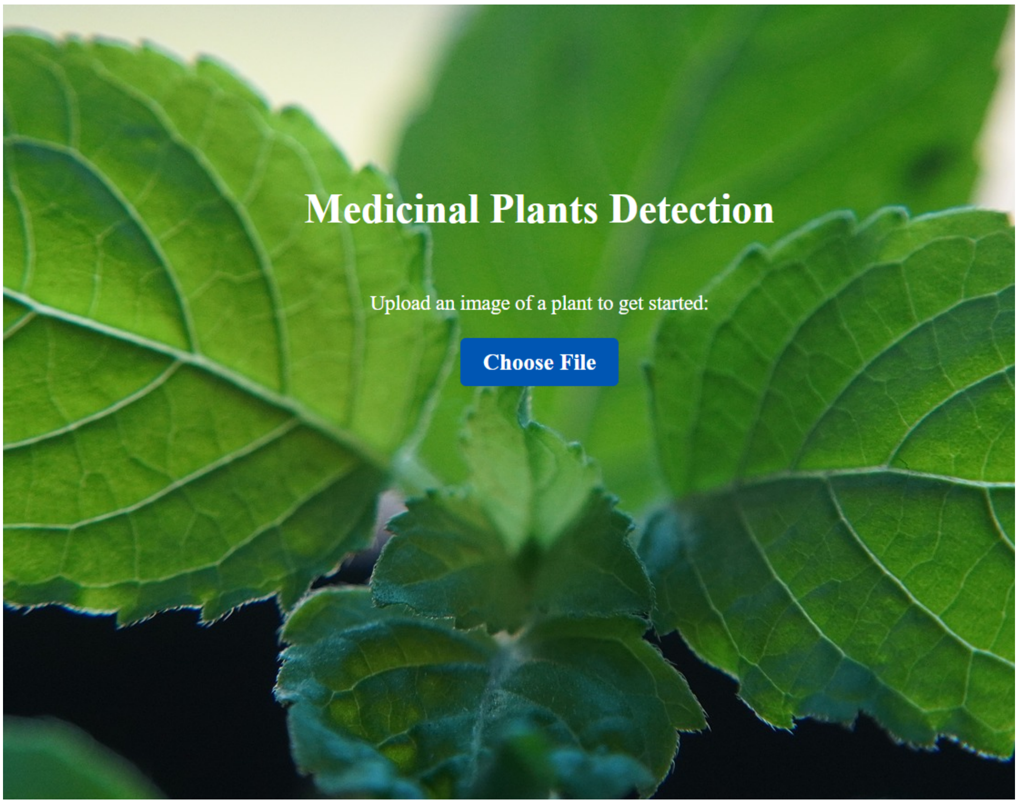

Fig 2.8 : Deployed web application

5.2 Future Scope

The use of machine learning (ML) and deep learning (DL) techniques for medicinal plant detection by classifying their leaves using ResNet 50 has immense potential for development in the future. Here are some potential areas of growth:

1. Increasing the accuracy and efficiency of the model: Researchers can continue to refine the models by using larger and more comprehensive datasets, improving the training algorithms, and developing more efficient hardware. Further exploration and optimization of various ML and DL techniques can also enhance the models' efficiency and accuracy.

2. Integrating other modalities: Although leaf classification is a reliable method for medicinal plant detection, integrating other modalities such as stem and flower morphology, chemical profiles, and genetic data can provide a more complete and accurate identification of the plant species. By combining various modalities, researchers can develop more robust and reliable classification systems.

3. Developing portable and user-friendly applications: Researchers can develop user-friendly applications that can be easily used by non-experts in the field to identify medicinal plants in the field. Such applications can be developed for smartphones, tablets, or other portable devices, making it easier for people to access information about medicinal plants.

4. Expanding the use of medicinal plant detection: In addition to traditional medicine and the pharmaceutical industry, ML and DL techniques for medicinal plant detection can also be applied to other areas such as food and agriculture, environmental conservation, and

bioprospecting. By exploring these new areas of application, researchers can uncover new uses for medicinal plants and expand their potential benefits.

5. Addressing ethical and legal concerns: As the use of ML and DL techniques for medicinal plant detection expands, it is important to consider ethical and legal concerns, such as the need to protect traditional knowledge, prevent biopiracy, and ensure fair and equitable distribution of benefits. Researchers can work with stakeholders to develop guidelines and regulations that ensure ethical and legal practices are followed.

Overall, the future scope of using ML and DL techniques for medicinal plant detection by classifying leaves using ResNet 50 is vast and promising, with potential applications across a range of industries and fields. By addressing the challenges and concerns, researchers can harness the full potential of these techniques and revolutionize the field of medicinal plant research.

REFERENCES

[1] B. Dudi and V. Rajesh, "Medicinal plant recognition based on CNN and Machine Learning," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8(4), pp. 999–1003, (2019). Available at: https://doi.org/10.30534/ijatcse/2019/03842019.

[2] R. Azadnia, M. Al-Amidi, H. Mohammadi, M. Cifci, M. Daryab and E. Cavallo, "An AI based approach for medicinal plant identification using deep CNN based on Global average pooling," *Agronomy*, vol. 12(11), pp. 2723, (2022). Available at: https://doi.org/10.3390/agronomy12112723.

[3] R. U. Rao, M. S. Lahari, K. P. Sri, K. Y. Srujana and D. Yaswanth, "Identification of medicinal plants using Deep Learning," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10(4), pp. 306–322, (2022). Available at: https://doi.org/10.22214/ijraset.2022.41190.

[4] J. Abdollahi, "Identification of Medicinal Plants in Ardabil Using Deep learning : Identification of Medicinal Plants using Deep learning." *2022 27th International Computer Conference, Computer Society of Iran (CSICC)*, pp. 1–6, (2022). Available at: https://doi.org/10.1109/CSICC55295.2022.9780493

[5] Image Source :-
https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758

[6] Dataset:-

Link: https://data.mendeley.com/datasets/nnytj2v3n5/1
DOI: 10.17632/nnytj2v3n5.1

APPENDICES

Code:

```
#install Kaggle this process is used to directly import the dataset from kaggle without download
!pip install -q kaggle
```

```
from google.colab import files
files.upload()
```

```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current bro
Saving kaggle (1).json to kaggle (1).json
{'kaggle (1).json': b'{"username":"ayushguleria191202","key":"35c1828be2cf9ea4f325a4f6f73ac844"}'}
```

```
#create a kaggle folder
! mkdir ~/.kaggle
```

```
#copy the kaggle.json file to folder created
! cp kaggle.json ~/.kaggle/
```

```
cp: cannot stat 'kaggle.json': No such file or directory
```

```
#Permission for the json to act
! chmod 600 ~/.kaggle/kaggle.json
```

```
chmod: cannot access '/root/.kaggle/kaggle.json': No such file or directory
```

```
!kaggle datasets download -d codefantasy/medical #api command used from kaggle site
```

```
from google.colab import drive #dataset was removed from kaggle site so used random dataset online Link : https://
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!unzip /content/drive/MyDrive/medical.zip #used for unzipping data
```

```
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Alpinia Galanga (Rasna)/AG-S-044.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Alpinia Galanga (Rasna)/AG-S-047.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Alpinia Galanga (Rasna)/AG-S-045.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Alpinia Galanga (Rasna)/AG-S-046.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Alpinia Galanga (Rasna)/AG-S-048.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Alpinia Galanga (Rasna)/AG-S-049.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Alpinia Galanga (Rasna)/AG-S-050.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-009.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-006.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-004.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-010.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-002.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-003.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-005.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-001.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-011.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-008.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-012.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-014.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-016.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-013.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-015.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-017.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-019.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-021.jpg
  inflating: Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/Amaranthus Viridis (Arive-Dantu)/AV-S-020.jpg
```

```
!pip install Augmentor #Augmentation
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting Augmentor
  Downloading Augmentor-0.2.12-py2.py3-none-any.whl (38 kB)
Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.9/dist-packages (from Augmentor) (1.22.4)
Requirement already satisfied: Pillow>=5.2.0 in /usr/local/lib/python3.9/dist-packages (from Augmentor) (8.4.0)
Requirement already satisfied: tqdm>=4.9.0 in /usr/local/lib/python3.9/dist-packages (from Augmentor) (4.65.0)
Installing collected packages: Augmentor
Successfully installed Augmentor-0.2.12
```

```
import Augmentor
p = Augmentor.Pipeline(r"/content/Medicinal Leaf Dataset/Segmented Medicinal Leaf Images")
p.zoom(probability=0.3, min_factor=0.8, max_factor=1.5)
p.flip_top_bottom(probability=0.4)
p.random_brightness(probability=0.3,min_factor=0.3, max_factor=1.2)
p.random_distortion(probability=1,grid_width=4, grid_height=4, magnitude=8)
p.sample(3000)
```

```
Initialised with 1835 image(s) found.
Output directory set to /content/Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/output.Processing <PIL.Ima
```

```
!pip install split-folders #used for splitting data for train and test
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting split-folders
  Downloading split_folders-0.5.1-py3-none-any.whl (8.4 kB)
Installing collected packages: split-folders
Successfully installed split-folders-0.5.1
```

```
[ ]
import splitfolders
input_folder = '/content/Medicinal Leaf Dataset/Segmented Medicinal Leaf Images/output'

splitfolders.ratio(input_folder, output="dataset2",
                   seed = 42,ratio =(.6,.4),#dataset was small so used 60 to 40 ratio as Resnet 50 model evalua
                   group_prefix=None)
```

```
Copying files: 3000 files [00:00, 3144.05 files/s]
```

```
[ ]
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os

import keras

from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import img_to_array, load_img
from keras.applications.resnet import ResNet50, preprocess_input, decode_predictions
```

```
[ ]
#Exploratory Data Analysis

len(os.listdir("/content/dataset2/train"))
```

```
30
```

```
[ ]
#loading images to Image data generator

train_datagen = ImageDataGenerator(zoom_range= 0.4, shear_range= 0.3,horizontal_flip = True,preprocessing_funct

```

+ Code + Text

```
#loading images to Image data generator

train_datagen = ImageDataGenerator(zoom_range= 0.4, shear_range= 0.3,horizontal_flip = True,preprocessing_function=preprocess_input)

val_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
```

```
[ ]
#preprocessing Data using preprocessing input()
train = train_datagen.flow_from_directory(directory= "/content/dataset2/train",
                                          target_size=(256,256),
                                          batch_size= 32)

val = val_datagen.flow_from_directory(directory= "/content/dataset2/val",
                                          target_size=(256,256),
                                          batch_size= 32)
```

```
Found 1788 images belonging to 30 classes.
Found 1212 images belonging to 30 classes.
```

```
[ ]
#visualizing our data/images
t_img , label = train.next()
```

```
[ ]
def plotImage(img_arr, label):

  for im , l in zip(img_arr , label):
    plt.figure(figsize=(5,5))
    plt.imshow(im)
    plt.show()
```

Building Our Model

```python
from keras.layers import Dense, Flatten
from keras.models import Model
from keras.applications.resnet import ResNet50
import keras
```

```python
base_model = ResNet50(input_shape=(256,256,3), include_top= False) # using Resnet 50 as i found hig
```

```python
for layer in base_model.layers:
  layer.trainable = False
```

```python
base_model.summary()
```

```
Model: "resnet50"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 256, 256, 3  0           []
                                )]

 conv1_pad (ZeroPadding2D)      (None, 262, 262, 3)  0           ['input_3[0][0]']

 conv1_conv (Conv2D)            (None, 128, 128, 64  9472        ['conv1_pad[0][0]']
                                )

 conv1_bn (BatchNormalization)  (None, 128, 128, 64  256         ['conv1_conv[0][0]']
```

```python
X = Flatten()(base_model.output) #single row and columns presentation

X = Dense(units = 30, activation='softmax')(X)


#Creating Model
model = Model(base_model.input, X)
```

```python
model.summary()
```

```
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 256, 256, 3  0           []
                                )]

 conv1_pad (ZeroPadding2D)      (None, 262, 262, 3)  0           ['input_3[0][0]']

 conv1_conv (Conv2D)            (None, 128, 128, 64  9472        ['conv1_pad[0][0]']
                                )

 conv1_bn (BatchNormalization)  (None, 128, 128, 64  256         ['conv1_conv[0][0]']
                                )

 conv1_relu (Activation)        (None, 128, 128, 64  0           ['conv1_bn[0][0]']
                                )

 pool1_pad (ZeroPadding2D)      (None, 130, 130, 64  0           ['conv1_relu[0][0]']
                                )

 pool1_pool (MaxPooling2D)      (None, 64, 64, 64)   0           ['pool1_pad[0][0]']
```

Compiling our model

```
[ ]  model.compile(optimizer= 'adam', loss= keras.losses.categorical_crossentropy, metrics=
```

Early stopping and Model Check point

+ Code

```
[ ]  from keras.callbacks import ModelCheckpoint, EarlyStopping

     #early stopping
     es = EarlyStopping(monitor= 'val_accuracy', min_delta= 0.01, patience= 3, verbose =1)


     #model check point
     mc = ModelCheckpoint(filepath="best_model.h5",
                          monitor= 'val_accuracy',
                          min_delta= 0.01,
                          patience= 3,
                          verbose =1,
                          save_best_only= True)
     cb =[es,mc] #callback
```

```
[ ]  #history model
     his = model.fit(train,
                          steps_per_epoch= 16,
                          epochs=50,
                          verbose= 1,
```

```
     #load best model

     from keras.models import load_model

     model = load_model("/content/best_model.h5")
```

```
[ ]  #evaluating model

     acc = model.evaluate_generator(val)[1]


     print(f"The accuracy of your model is = {acc*100} %")
```

```
<ipython-input-51-46c58561324f>:3: UserWarning: `Model.evaluate_generator` is deprecated and will
  acc = model.evaluate_generator(val)[1]
The accuracy of your model is = 93.06930899620056 %
```

```
[ ]  y_pred = model.predict(val) #for making classification report
     y_true = val.labels
```

```
38/38 [==============================] - 33s 884ms/step
```

```
[ ]  from sklearn.metrics import classification_report  #classification report #found very less scor

     class_names = list(val.class_indices.keys())
```

```
#refrence + making a dictionary.
ref = dict(zip(list(train.class_indices.values()),list(train.class_indices.keys())))
```

```
print(ref)
```

```
{0: 'Alpinia Galanga (Rasna)', 1: 'Amaranthus Viridis (Arive-Dantu)', 2: 'Artocarpus Hete
```

```
import pickle
```

```

with open("test.txt", 'w') as f:
    for key, value in ref.items():
        f.write('"%s":"%s"\n' % (key, value))

#a_file = open("data.pkl", "rb")
#output = pickle.load(a_file)
#a_file.close()
```

\+ Code

```
#prediction process
def prediction(path):

    img = load_img(path,target_size=(256,256))
    i = img_to_array(img)
    im = preprocess_input(i)
    img = np.expand_dims(im,axis= 0)
    pred = np.argmax(model.predict(img))
    print(f"The image belongs to{ref[pred]}")
    #print(pred)
```

bb

ORIGINALITY REPORT

6%	4%	3%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Middle East College of Information Technology Student Paper	1%
2	www.mdpi.com Internet Source	1%
3	www.ijraset.com Internet Source	1%
4	Jafar Abdollahi. "Identification of Medicinal Plants in Ardabil Using Deep learning : Identification of Medicinal Plants using Deep learning", 2022 27th International Computer Conference, Computer Society of Iran (CSICC), 2022 Publication	1%
5	Submitted to University of Essex Student Paper	1%
6	Submitted to Liverpool John Moores University Student Paper	<1%
7	www.warse.org Internet Source	