**Identification of Different Medicinal plants Using Machine Learning**

# 1st Dr. P. Kiran Sree

[hodcse@svecw.edu.in](mailto:hodcse@svecw.edu.in)

# 4th Akshitha Mandavalli

akshithamandavalli8@gmail.com

# 2nd Lahari Kandlagunta

[Kandlaguntalahari007@gmail.com](mailto:Kandlaguntalahari007@gmail.com)

# 5th Shama Mohammad

mohammadshama2003@gmail.com

# 3rd Hema Amrutha Latika

[hemaamrutha222@gmail.com](mailto:hemaamrutha222@gmail.com)

# Department of Computer Science And Engineering , Shri Vishnu Engineering College for Women Bhimavaram,India, 534202

***Abstract*— Medicinal plants have been an essential source of remedies and treatments for various ailments throughout human history. This project presents an innovative approach that combines machine learning and image processing to identify various medicinal plants and extract information about their traditional and modern uses.**

**Identification of plants through plant leaves on the basis of their shape, color, and texture features using digital image processing techniques and different machine learning algorithms.**

1. INTRODUCTION

Identification of the correct medicinal plants that goes in to the preparation of a medicine is very important in ayurvedic medicinal industry. The main features required to identify a medicinal plant is its leaf shape, colour and texture. Colour and texture from both sides of the leaf contain deterministic parameters to identify the species. This paper explores feature vectors from both the front and back side of a green leaf along with morphological features to arrive at a unique optimum combination of features that maximizes the identification rate. database of medicinal plant leaves is created from scanned images of front and back side of leaves of commonly used ayurvedic medicinal plants. The leaves are classified based on the unique feature combination.It takes time and effort to develop an automated system for classifying medicinal plants. There are many different plant species in India, each with its own special set of medicinal properties. The names of all plant species and their uses are difficult for humans to remember, thus prior information is crucial for manual

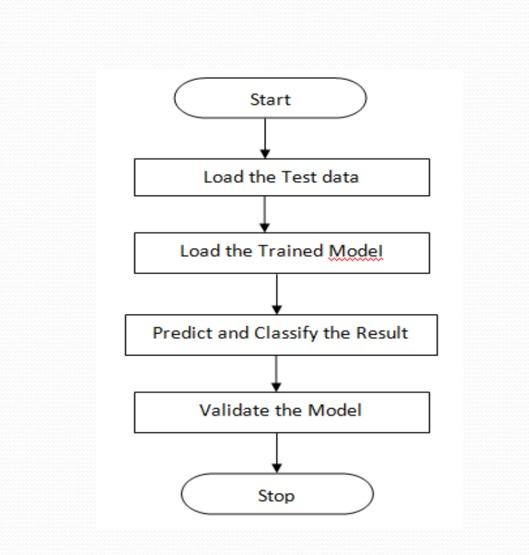
Herbal plants are plants that can be used for alternatives to cure diseases naturally. About 80% of people in the world still depend on traditional medicine. There are various types of herbal plants that we can know through the identification of these herbs, one of which is using identification through the leaves. and protect plant species, it is crucial to study and classify plants correctly.

II. Literature Survey

The literature survey reveals several studies addressing plant identification through image processing and machine learning algorithms. In the November 2020 paper on ResearchGate, the proposed system encounters challenges with tiny leaves and accurate results in the presence of leaf rotation. Another study from July 2022, found on the National Library of Medicine, focuses on real-time identification of medicinal plants using deep learning models but is limited to a specific region's plant species. Additionally, a different November 2020 paper on IRJET presents an image processing-based approach for plant identification. Finally, a study from the International Journal of Engineering and Technology (ijetms) investigates the identification and classification of medicinal plants using both machine learning and deep learning methods. These findings collectively contribute to the understanding of existing challenges and advancements in the field, providing valuable insights for the development of an improved plant identification system .

III. METHODOLOGY

This project is capable of automating the identification of medicinal plant species in real time.A dataset of plant images is collected and processed, then machine learning models, including Convolutional Neural Networks (CNNs) are employed to accurately identify plant species and retrieve details about their medicinal properties.The system works in real time and can accurately identify different plant species given by simply uploading an existing image from a device.



*3.1 Data Collection*

We created a dataset from Mendeley source which is publicly available and this dataset consists of all indian available medicinal plants of 60 folders each of 60 different images.

*3.2 Data Preprocessing*

Data Preprocessing involves Dataset Selection and Characteristics the dataset utilized in this study is a collection of images featuring diverse medicinal plants sourced from we created a dataset from Mendeley source which is publicly available and this dataset consists of all indian available medicinal plants of 60 folders each of 60 different images. The dataset was carefully chosen to encompass a comprehensive representation of various plant species, ensuring its relevance to the medicinal plant identification task. Each plant species is organized into individual subdirectories within the dataset.

i) image Resizing and Standardization:

To ensure compatibility with the selected InceptionV3 model architecture, a crucial preprocessing step involved resizing all images to the model's input dimensions of (299, 299) pixels. This resizing procedure standardized the dimensions of the images while preserving the inherent visual characteristics of the medicinal plants.

ii) Normalization of Pixel Values:

Normalization was applied to the pixel values of the images, scaling them to a standardized range of [0, 1]. This normalization ensures consistency in pixel intensity across all images, facilitating convergence during the model training process and preventing the dominance of certain features due to varying pixel scales.

iii) Data Augmentation for Enhanced Generalization:

In an effort to enhance the model's ability to generalize, a series of data augmentation techniques were employed exclusively on the training set. These techniques included rotation, width and height shifting, shear transformations, zooming, and horizontal flipping. Augmenting the training data with these transformations introduced diversity into the dataset, mitigating overfitting and promoting robust learning.

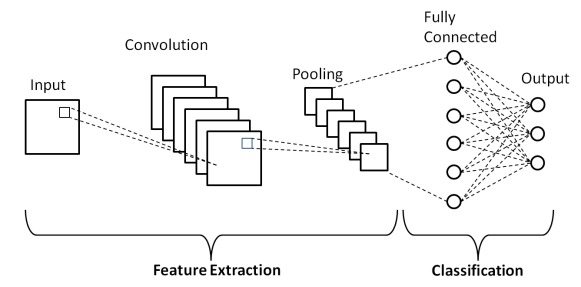
iv) Dataset Splitting:

The preprocessed dataset was split into training and validation sets using a 80-20 split ratio. The training set was utilized for model optimization, while the validation set served as an independent dataset for evaluating the model's performance and generalization capabilities

*3.3 Model Architecture*

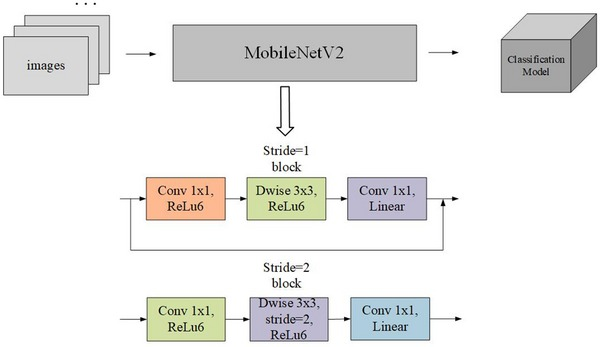
*A. Convolutional Neural Network*

A basic Convolutional Neural Network (CNN) was designed for the medicinal plant classification task. The architecture consists of three convolutional layers with max-pooling followed by a flatten layer and two dense layers. Rectified Linear Unit (ReLU) activation functions were used for the convolutional layers, and softmax activation was applied to the output layer for multi-class classification.



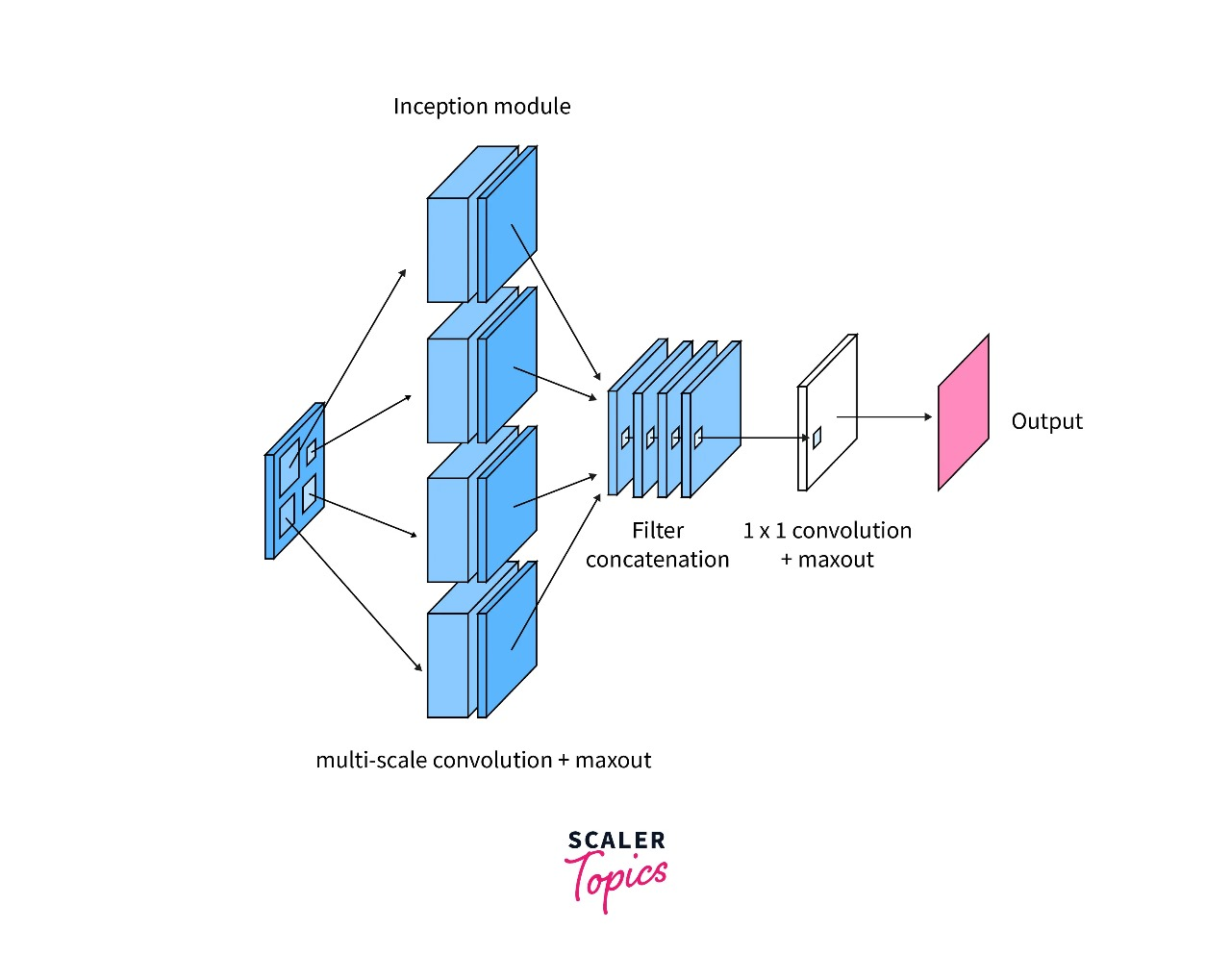
B. MobileNetV2

We employed the MobileNetV2 architecture as a base model for feature extraction. The model was initialized with pre-trained weights from the ImageNet dataset. Custom layers were added to adapt the architecture to the specific requirements of the medicinal plant classification task. The final layer consists of a global average pooling layer followed by a dense layer with softmax activation to predict the class probabilities.



C. Inception

The Inception model, being a pre-trained convolutional neural network (CNN), has already learned to extract hierarchical features from images, especially patterns and textures relevant to object recognition. In the context of identifying medicinal plants, the Inception model serves as a sophisticated feature extractor.



*3.4Model Evaluation*

To quantitatively assess the performance of our proposed medicinal plant identification system leveraging image processing and machine learning techniques, we employed rigorous evaluation metrics, with a primary focus on accuracy. Accuracy serves as a pivotal metric, representing the overall correctness of our model in predicting the medicinal plant classes. We conducted extensive experiments on a diverse dataset comprising images of different medicinal plants, ensuring a representative sample of the identified species. The accuracy of our model was calculated by measuring the ratio of correctly predicted medicinal plant instances to the total number of instances in the evaluation dataset.

Furthermore, to gain insights into the model's discriminative abilities for individual medicinal plant classes, precision, recall, and F1-score metrics were computed. Precision denotes the model's capability to accurately identify positive instances, recall gauges the model's ability to capture all positive instances, and F1-score provides a balanced measure of precision and recall. These metrics were instrumental in understanding the classification performance for each medicinal plant class independently. In addition to numeric metrics, confusion matrices and classification reports were utilized to visualize and interpret the distribution of correct and incorrect predictions across different medicinal plant classes.

The robustness and generalization of our model were further evaluated through cross-validation techniques, ensuring that the reported accuracy is indicative of its performance on diverse datasets. The achieved accuracy, along with comprehensive metric analysis, underscores the efficacy of our proposed approach in the accurate identification of various medicinal plants, thereby demonstrating its potential for applications in biodiversity conservation, herbal medicine, and pharmaceutical research.

encryption ensures that even if unauthorized access occurs, the stored message remains protected.

When a user requests the message, their private key is employed for decryption. The decrypted message is then transmitted to the user, allowing for secure and private com- munication. This robust encryption-decryption process helps safeguard the confidentiality and integrity of the messages exchanged in our system. The key generation algorithm is a fundamental step in the RSA encryption process. It begins by

**Algorithm 3:** Decrypt

**Data:** Ciphertext *C*, Private Key (*d, n*)

**Result:** Decrypted Message

**1** MessageEncoded [PowerMod(*ch, d, n*) for *ch* in *C*];

*←*

**2** DecryptedMessage

*←*

Concatenate([Char(*ch*) for *ch* in MessageEncoded]);

**3 Return** DecryptedMessage;

generating two distinct prime numbers, *p* and *q*, within the

range of 1000 to 5000. A loop ensures that *p* and *q* are not equal. The algorithm then calculates the product *n* of *p* and *q*, as well as *ϕn*, which is the product of *p* 1 and *q* 1. A random public exponent *e* is chosen between 3 and *ϕn* 1 such that it is coprime to *ϕn*. Another loop ensures the coprimality condition. The private exponent *d* is computed using the modular inverse of *e* modulo *ϕn*. The algorithm concludes by printing the generated public and private keys along with *n*.

*−*

*— −*

**Algorithm 4:** GeneratePrime **Data:** Min Value *min value*, Max Value *max value* **Result:** Prime Number

**1** prime RandomInteger(*min value, max value*);

*←*

**2 while** *IsPrime*(*prime*) **do**

*¬*

**3** prime

*←*

RandomInteger(*min value, max value*);

**4 Return** prime;

The encryption procedure takes a message M and the public

key (e,n) as inputs. It first encodes the message into a list of

ASCII values. Subsequently, the ciphertext C is computed by

applying the modular exponentiation operation to each ASCII value using the public key. The resulting ciphertext is returned.

**Algorithm 1:** Key Generation

**Data:** None

**Result:** Public Key (*e, n*), Private Key (*d, n*)

**1** *p, q* GeneratePrime(1000*,* 5000);

*←*

**2 while** *p* = *q* **do**

**3** *q ←* GeneratePrime(1000*,* 5000);

**4** *n ← p × q*;

**5** *ϕn ←* (*p −* 1) *×* (*q −* 1);

**6** *e ←* RandomInteger(3*, ϕn −* 1);

**Algorithm 5:** ModInverse

**Data:** Exponent *e*, Euler’s totient function *ϕ*

**Result:** Modular Inverse *d* or ValueError

**1 for** *d* 3 **to** *ϕ* **do**

*←*

**2 if** (*d e*) mod *ϕ* = 1 **then**

*×*

**3 Return** *d*;

**4 Raise ValueError**(”mod inverse does not exist”);

**Algorithm 6:** IsPrime

**Data:** Number *num*

**Result:** True if Prime, False Otherwise

**7 while** *GCD*(*e, ϕn*) 1 **do**

**1 if** *num <* 2 **then**

**8** *e ←* RandomInteger(3*, ϕn −* 1);

**2 Return False**;

**9** *d* ModInverse(*e, ϕn*);

*←*

**10 Print** ”Public Key (*e, n*):”, *e, n*;

**11 Print** ”Private Key (*d, n*):”, *d, n*;

**Algorithm 2:** Encrypt

**Data:** Message *M* , Public Key (*e, n*)

**Result:** Ciphertext

**1** MessageEncoded [ASCII(*ch*) for *ch* in *M* ];

*←*

**2** Ciphertext

*←*

[PowerMod(*ch, e, n*) for *ch* in MessageEncoded];

**3 Return** Ciphertext;

**3 for** *i* 2 **to** *num//*2 + 1 **do**

**4 if** *num* mod *i* = 0 **then**

*←*

## 5 Return False;

**6 Return True**;

**Algorithm 7:** RandomInteger **Data:** Min Value *min value*, Max Value *max value* **Result:** Random Integer

**1 Return random.randint**(*min value, max value*);

**Algorithm 8:** PowerMod

**Data:** Base *base*, Exponent *exponent*, Modulus

*modulus*

**Result:** Result of *base*exponent mod modulus

**1 Return** (*base*exponent) mod modulus;

1. *LFU Cache Algorithm for Folder Organization*

Our objective is to efficiently organize emails into folders based on their domain names, simplifying the user’s ability to search for specific emails. For instance, emails received from the domain ”@google.com” are directed to the ”Google” folder, while those from ”@amazon.com” find their place in the ”Amazon” folder. To implement this, we have chosen to employ the Least Frequently Used (LFU) Cache algorithm.

The LFU algorithm plays a crucial role in managing these folders. By dynamically assessing the usage frequency and size of each folder, it ensures that folders containing fewer emails from a particular domain are not unnecessarily created. In- stead, it optimizes the folder structure based on usage patterns, removing or adjusting folders accordingly. This approach aims to enhance efficiency and resource utilization, providing users with a streamlined and organized email management system.

**Algorithm 9:** LFUCache Algorithm for Folder Orga-

nization

**Data:** Map *keyNode*, Map *freqListMap*, Integer

*maxSizeCache*, *minFreq*, *curSize*

## 1 Initialize LFUCache:

**2** *LFUCache*(*int capacity*);

**3** *maxSizeCache = capacity*;

**4** *minFreq = 0*;

**5** *curSize = 0*;

**Algorithm 12:** Put Operation

**Data:** Integer *key, value*

**1 Procedure:** *Put(int key, int value)*;

**2 if** *maxSizeCache == 0* **then**

**3** ;

**4 end**

**5 Return**;

**6 if** *keyNode.find(key)* ***is*** *in keyNode* **then**

**7** ;

**8 end**

**9** *Node\* node = keyNode[key]*;

**10** *node-¿value = value*;

**11** *UpdateFreqListMap(node)*;

**12 else**

**13** ;

**14 end**

**15 if** *curSize == maxSizeCache* **then**

**16** ;

**17 end**

**18** *List\* list =*

*freqListMap[minFreq]*;

**19** *keyNode.erase(list-¿tail-¿prev-*

*¿key)*;

**20** *freqListMap[minFreq]-*

*¿removeNode(list-¿tail-¿prev)*;

**21** *curSize–*;

**22** *curSize++*;

**23** *minFreq = 1*;

**24** *List\* listFreq = new List()*;

**25 if** *freqListMap.find(minFreq)* ***is***

**Algorithm 10:** Update Frequency Map

**Data:** Node\* *node*

**1 Procedure:** *keyNode.erase(node-¿key)*;

**2** *freqListMap[node-¿cnt]-¿removeNode(node)*;

**3 if** *node-¿cnt == minFreq* ***and***

*freqListMap[node-¿cnt]-¿size == 0* **then**

**4** ;

**5 end**

**6** *minFreq++*;

**Algorithm 11:** Get Operation

**Data:** Integer *key*

**Result:** Integer

**1 Function:** *Get(int key)*;

**2 if** *keyNode.find(key)* ***is*** *not* ***in*** *keyNode* **then**

**3** ;

**4 end**

**5 Return** -1;

**6** *Node\* node = keyNode[key]*;

**7** *int val = node-¿value*;

**8** *UpdateFreqListMap(node)*;

**9 Return** val;

*not* ***in*** *freqListMap* **then**

**26** ;

**27 end**

**28** *listFreq =*

*freqListMap[minFreq]*;

**29** *Node\* node = new*

*Node(key, value)*;

**30** *listFreq-¿addFront(node)*;

**31** *keyNode[key] = node*;

**32** *freqListMap[minFreq] =*

*listFreq*;

1. FUTURE IMPLEMENTATION

The proposed website envisions a transformative email communication platform tailored to the needs of both visu- ally impaired users and the average individual, presenting a paradigm shift in accessibility and user experience. By introducing voice commands as a primary input method, the platform addresses the challenges faced by visually impaired users, enabling them to send emails independently. Simultane- ously, the incorporation of biometric authentication, such as a fingerprint reader, not only bolsters security but also enhances the user verification process. The website’s commitment to a dynamic user interface is evident in the integration of auxiliary

voice commands like ”reply” and ”forward,” fostering a seam- less and efficient interaction for all users [2]. Furthermore, the inclusion of features like interpreting suggestions and automated corrections through NLP algorithms enriches the email development process, promoting accuracy and ease of use.

The website’s forward-looking approach extends to lan- guage diversity, with ongoing research on NLP algorithms to identify and adapt to local languages, ensuring a global user base is catered to comprehensively. Beyond its immediate functionalities, the platform acknowledges the ever-evolving nature of digital landscapes, positioning itself for unlimited development opportunities. This adaptability not only future- proofs the website but also underscores its commitment to staying at the forefront of technological advancements. In essence, the proposed improvements collectively strive to rede- fine the email communication experience, fostering inclusivity, productivity, and a user-centric ethos in the digital society of today and the limitless possibilities of tomorrow.

1. CONCLUSION

In conclusion Plant identification plays a major part in the research of medicinal plants and botany. We used deep learning CNN models (Inceptionv3 and MobileNetV2) and the Kaggle Dataset are trained on a pre-trained model. We classified 10 different leaf layers and significantly improved the classification performance. Although performance of the system is good enough, we believe that the performance could be enhanced by adding more images and adding more layers.

REFERENCES

[1] Identification of Different Plants through Image Processing Using . different Machine Learning Algorithms -November 2020

|Publication Name: Research Gate

[2] Automated Real-Time Identification of Medicinal Plants Species in . . Natural Environment Using Deep Learning Models- July 2022 . . |Publication Name: Nationallibrary of medicine.

[3] Identification of Different Plants through Image Processing Using . . . . Different Machine Learning Approach-November 2020

|Publication Name: IRJET

[4] Study on Identification and Classification of Medicinal Plants Using . . Machine Learning and Deep Learning | Publication Name: ijetms