

# PLANT IDENTIFICATION USING MACHINE LEARNING

Harisha K S<sup>1</sup>, Pavitra M H<sup>2</sup>, Spandana S R<sup>3</sup>, Shruthi B S<sup>4</sup>, Archana B S<sup>5</sup>

<sup>1</sup> Assistant Professor, Department of Electronics and communication engineering<sup>1</sup>

<sup>2,3,4,5</sup> B.E. Students, Department of Electronics and Communication Engineering

Government Engineering College Haveri,

Visvesvaraya technological university, Belagavi

\*\*\*

**Abstract** - The preservation of therapeutic plants is crucial as it will help a broad variety of sectors, including medicine, botanic research, and plant taxonomy studies, among others. Existing technologies cannot imitate the range of therapeutic plant species present in India. The suggested technique facilitates the classification of medicinal plants by exploiting textural aspects crucial in leaf recognition and identification. The three key phases of the proposed technique are picture enhancement, feature extraction, and classification. The photographs of the leaves are shot using cellphones and then processed using digital image processing algorithms to extract the features that may be compared between them. Finally, the CNN classifier is employed to develop an automated classifier.

**Key Words:** Plant identification, CNN algorithm, Image processing.

## 1. INTRODUCTION

The living beings on the earth are dependent upon the plants. The plants are the main source of oxygen. Apart from this, the plants are used in a variety of industrial applications such as herbs, ingredients in biofuels, biomass, pharmaceuticals etc. People have been using plant as a traditional medicine. It's considered non-expensive and does not cause any side effects. Different kinds of plant species are available on the earth; recognition of these plants is difficult. There are several methods to recognize a plant. At present, plants are identified manually by taxonomist, which are prone to human errors. To avoid this, automated plant identification system has been developed by several researchers. Medicinal plants are those plants that are used in treating and preventing specific ailments and diseases that affect human beings. Knowledge accumulated by the villagers and tribal on herbal medicine remains unknown to the scientists and urban people. Many plant species associated with the rural people are on the verge of disappearing and are on vulnerable list. The impact of deforestation, urbanization and modernization is shifting the rural people from their natural habitats and their very knowledge particularly

with respect to herbal drugs is slowly disappearing. Our immediate goal is to preserve this knowledge. Whatever knowledge exists today is mostly confined to the older generation. In this context, some approaches are needed to preserve and develop traditional knowledge.

Plant classification is an active research area, with plants being used in agriculture, medicine and food industry, as well as in the preparation of cosmetics and a range of food products. Individuals cannot characterize plants as effectively as botanists, who do so by classifying those utilizing leaves, flowers, seeds, and roots. Agent-based systems classify plants into species that can be used in medicine and as food. Keeping in mind the end goal, which is to provide data on therapeutic plants, it is critical to have an intelligent system framework that recognizes natural species with the assistance of their digitized databases. An intelligent system is a key strategy utilized in plant based recognition systems to create real models from plants, incorporating pattern classification and object recognition. Researchers have created a plant acknowledgment framework utilizing plant leaves, flowers, fruits and seeds and by taking into consideration the visual content of their images such as color, texture, and shape.

Plant leaf identification using machine learning refers to the process of training a computer system to recognize different types of plant leaves using algorithms and statistical models. This process involves feeding the computer system with large amounts of data containing images of plant leaves, and then using machine learning algorithms to identify patterns and features that distinguish one type of leaf from another. The goal of plant leaf identification using machine learning is to develop a system that can accurately identify a plant's species based on its leaves. This can be useful in a variety of contexts, such as agriculture, horticulture, and environmental conservation. For example, farmers can use the technology to identify and manage weeds, while conservationists can use it to monitor endangered species and their habitats.

Ayurveda is an ancient medicinal system that is practiced in India and has its origins in the Vedic times, approximately 5000 years ago. Ayurveda is considered to be the oldest healing science. This widespread and

extensive use in the field of medicine makes Ayurveda the 'mother of healing'. These Ayurvedic plants are used for preparing medicines on a commercial basis. This has resulted in the production and marketing of Ayurvedic medicines to become a thriving industry with its annual turnover exceeding Rs 4000 crores. Because of this, the number of licensed Ayurvedic medicine manufacturers in India now exceeds 8500. Because of this increased commercialization of the Ayurvedic sector, several issues regarding the raw material quality used for their preparation need to be focused.

These plants are usually collected by tribal masses that are not professionally trained in the work of identifying the correct plants. Even the manufacturing units, at times, receive improper or substituted medicinal plants. Most of these manufacturing units do not have proper quality control mechanisms that can screen these plants. In addition to this, confusion due to the different local names of these plants makes the matters worse. Because of the usage of improper raw materials, Ayurvedic medicine becomes inefficient. It may result in certain unpredictable side effects too. As a result, an intelligent system needs to be developed that can properly identify the ayurvedic plant based on the leaf samples. This will, in turn, improve the quality of the medicine and will also maintain its credibility.

## 2. PROBLEM STATEMENT

The rapid pace of urbanization and industrialization has led to the loss of biodiversity and the endangerment of many plant species. It is essential to develop a system that can accurately

and quickly identify plant species to help with conservation efforts and restoration of degraded ecosystems. The use of machine learning in plant identification can significantly improve the accuracy and speed of the identification process. The challenge, however, lies in developing a robust and reliable algorithm that can accurately identify plant species even when dealing with variations in environmental conditions, growth stages, and the presence of pests and diseases. To identify the plant by their leaf, a type of artificial neural network called convolutional neural network(CNN) is used.

## 3.OBJECTIVES

- Plant identification using a user-uploaded photograph would be one major goal of the proposed system.
- The objective of this project is to reduce manual work and increase the efficiency by the automatic identification of plants.
- Detect the Ayurvedic Medicinal Plants using CNN.

- To give formers a cost-effective and time-saving option.

- To deliver the most accurate results possible

## 4. LITERATURE REVIEW

Skanda H N[1] has proposed a leaf identification technique in which Leaves can be identified using digital fingerprint. This method works the same way a media recognition app works. By scanning the leaf by lasers, different depth points can be marked and connected to form an image which can be plotted against a graph. The area enclosed by graph form the unique digital fingerprint of the leaf which can be used to recognize the plant.PM KUMAR[2] describes image processing technique for identifying ayurvedic medicinal plants by using leaf samples. Forests and wastelands sources for over 80% of ayurvedic plants. There exists no predefined database of Ayurvedic plant leaves. A set of leaf images of medicinal plants were collected from the botanical garden. To improve the efficiency of plant identification system, machine learning techniques can be used over human visual perception as it is more effective. Weka is a collection of machine learning algorithms for data mining. It contains feature selection, regression, classification and pre-processing tools. Graphic user interface is used for accessing the functions. This proposed scheme uses some of the classifiers such as Support Vector Machine (SVM) and Multilayer perceptron (MLP). For reverting and classifying of data SVM is used. MLP is an artificial neural network which helps in routing the input data of one set to appropriate output pertaining to another set. The highest identification rate in SVM is 98.8% and 99% obtained in MLP.

H. A. Chathura Priyankara and D. K. Withanage [3] discusses the Computer-assisted android system for plant identification based on leaf image using features of SIFT along with Bag of Word (BOW) and SVM as classifiers. This identification method for android involves 8 stages. It employs client-server model of architecture. Server involves 2 main activities. The first activity is to train the SVM classifier to generate feature vector required for classification and then save it. The second activity is generation of feature vector with the help of photographs uploaded. These are uploaded by android client. The generated vector is used for identification by the SVM classifier. The process of training SVM involves SIFT descriptors along with Bag of Feature model that helps in generation of classifier. The generation of classifier involves 4 steps. In the first step, using the reduction method of data space SIFT descriptors are extracted from each leaf image belonging to the training data set. The second step is to cluster all the extracted features into feature bags using BOW methods. In the next step bow histograms are

generated by taking all the images in the training dataset into consideration. In the final step all the histograms are passed to the SVM as the classification feature vector. SVM creates and saves the classifier in the server storage. The RGB image is converted into a greyscale image before extracting SIFT feature points as a pre-processing step. Following which involves extraction of key point and generating of descriptors by using SIFT algorithm that involves CBIR (content-based image retrieval) algorithm. Using k-means clustering method all the collected SIFT features from training dataset are clustered into several clusters. A histogram represents each image in the training dataset. Histograms are classified using multi-class linear support vector machine. Android implementation involves client application that consumes algorithm of leaf recognition. Dynamic Link Library (DLL) application is used to invoke

The Rafael Rojas [4] discusses the general steps for plant identification using pre-processing, feature extraction and their classifications. The availability of classic classification algorithms are not accessible, therefore it gave way for new methodologies applying data mining methods in specific domain. Considering the extraction process, initially we come across pre-processing where extraction of the available data is done to form images. These leaf images are transformed into quality binary images using normalization and segmentation processes. Most of the leaf datasets is available online and here we scale it in order to constrain the size. We also consider image normalization where brightness and contrast features are considered. Binary images of the leaves are obtained using leaf segmentation that is necessary in order to eliminate noise using morphological features. By using contour extraction, the geometric features of leaves are obtained. The Feature extraction process is used for plant recognition which considers various parameters such as area convexity, perimeter convexity and so on describing the leaf characteristics. Classification process is a supervised learning technique where we use ANN, SVM and KNN classifiers which improves classification accuracy.

The Rafael Rojas-Hernández and Asdrúbal López-Chau [4] describes the methods of shape feature extraction that is Scale Invariant Feature Transform (SIFT) and colour feature extraction Grid Based Colour Moment (GBCM) to identify plants which comprises of phases such as image acquisition, image processing, feature extraction, identification and performance measurement. The Image acquisition process mainly deals with acquiring datasets of different tree species. Image processing mainly aims to enhance image data required for further processing by discarding the

undesired distortions. This process includes the phases of rotation, scaling and variations of leaf samples for further testing. Shape features and colour features are extracted using scale invariant feature transform and grid-based colour moment respectively. In SIFT both domains of spatial and frequency are considered. Geometric transforms makes it robust to illumination and noise. It also considers varying views of the object taking into consideration that helps in detection of the scale space extrema and an elaborate analysis is performed with respect to various features allowing the rejection of points corresponding to low contrast regions. The gradient magnitude and orientation is measured for each image sample. The orientation ranges from 360 degree and the Gaussian weighted circular window is used to measure the magnitude. The Grid-based colour moment is extracted using colour moment technique. Three parameters are used to calculate skewness, mean and standard deviation of an image. After acquiring these data, we go for an identification process based on Euclidian distance that determines the root square differences between values of a pair of objects considered. This methodology achieved an accuracy of 87.5% .

K. Li, Y. Ma and J. C. Príncipe [6] divides the identification of plant into three stages, they are: synthetic plant collection, spatiotemporal evolution model and automata extraction. In the first step, finite set of elements characterizes the plant development and growth in synthetic collection of plants. This finite set takes the indeterminate and complex shape. The mathematical formulation of underlying rules is named as L-system. An L-system is defined as the 3-tuple  $G = (V, w, P)$  system. The artificial regularity, also it introduces randomness to its production. In a synthetic plant collection, image processing and feature extraction method is also used. The L-systems are also visualized using truth table using turtle interpretation and saved as JPEG images to simulate the real plants. To detect the main axis and root of the plant, Hough transform is used. In the second step, that is, the spatiotemporal evolution model, KAARMA network models a dynamic system as defined by the general continuous non-linear state transitions functions and an observation function. To train a STEM, kernel adaptive KAARMA is used. In the third step, that is, the automata extraction, the discrete finite automation (DFA) is used where all the state transitions are uniquely determined by input symbols, from an initial state. The DFA is used to model the discrete time dynamical system in the discrete state space. A DFA can be represented in two ways, state transitions or lookup table. The analytical descriptor of a languages known as an Automata. The

DFA also validates the corresponding regular grammar produced by the language.

**5.METHODOLOGY**

1.DATA SET: The dataset for training is obtained from Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI). LIDC and IDRI consist of 1000 CT scans of both large and small tumors saved in Digital Imaging and Communications in Medicine (DICOM) format.

2. IMAGE SEGMENTATION: The segmentation of photographs is the phase where the visual image is partitioned into several parts. This normally helps to identify artifacts and boundaries. The aim of segmentation is to simplify the transition in the interpretation of a picture into the concrete picture that can be clearly interpreted and quickly analyzed.

3.PRE-PROCESSING: In preprocessing stage, the median filter is used to restore the image under test by minimizing the effects of the degradations during acquisition. Various preprocessing and segmentation techniques of lung nodules are discussed in. The median filter simply replaces each pixel value with the median value of its neighbors including itself. Hence, the pixel values which are very different from their neighbors will be eliminated.

the convolutional layer reducing the memory footprint and increasing the performance of the network. The important features of CNN lie with the 3D volumes of neurons, local connectivity, and shared weights. A feature map is produced by the convolution layer through the convolution of different sub-regions of the input image with a learned kernel. Then, a non-linear activation function is applied through the ReLu layer to improve the convergence properties when the error is low. In the pooling layer, a region of the image/feature map is chosen as the pixel with the maximum value among them or average values is chosen as the representative pixel so that a 2x2 or 3x3 grid will be reduced to a single scalar value. This results a large reduction in the sample size. Sometimes, traditional Fully-Connected (FC) layer will be used in conjunction with the convolutional layers towards the output stage.

Front-End Development Using Python Tkinter: Modern computer applications are user-friendly. User interaction is not restricted to console-based I/O. They have a more ergonomic graphical user interface (GUI) thanks to highspeed processors and powerful graphics hardware. These applications can receive inputs through mouse clicks and can enable the user to choose from alternatives with the help of radio buttons, dropdown lists, and other GUI elements

The data training in our CNN model has to satisfy following constraints: 1) No missing values in dataset. 2) The dataset must distinctly be divided into training and testing sets, either the training or the testing set shouldn't contain any irrelevant data out of our model domain in case of an image dataset all the images must be of the same size, one uneven distribution of image size in our dataset can decrease the efficiency of our neural network. 3) The images should be converted into black and white format before feeding it into the convolution layer because reading images in RGB would involve a 3-D num Py matrix which will reduce the execution time of our model by a considerable amount. Any kind of corrupted or blurred images should also be trimmed from the database before feeding it into the neural network. This is all about data pre-processing rules, let us dive right into the working of the convolution neural network

A CNN IS COMPOSED OF SEVERAL KINDS OF LAYERS: Convolutional layer: creates a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image, few pixels at a time. Pooling layer (down-sampling): scales down the amount of information the convolutional layer generated for each feature and maintains the most essential information (the process of the convolutional and pooling layers usually repeats several times). Fully connected input layer: flattens the outputs generated

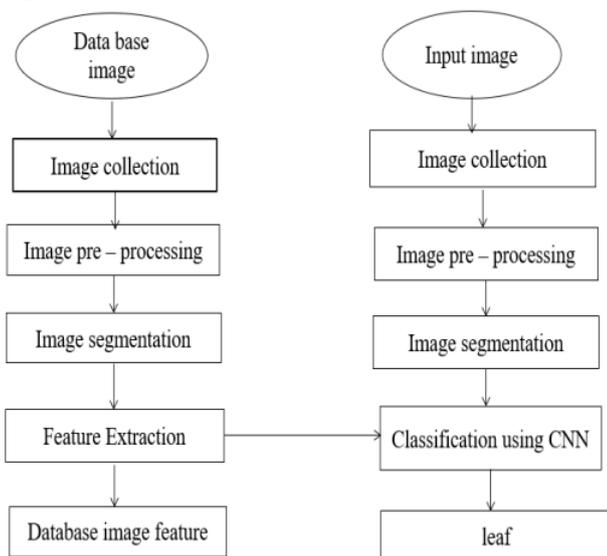


Figure1: Data flow Diagram of Training and Testing Phase

Image processing stages involves

1. Image Collection
2. Image Preprocessing
3. Image Segmentation
4. Feature Extraction
5. Training
6. Classification

4. CONVOLUTIONAL NEURAL NETWORKS: A CNN is type of a DNN consists of multiple hidden layers such as convolutional layer, RELU layer, Pooling layer and fully connected a normalized layer. CNN shares weights in

by previous layers to turn them into a single vector that can be used as an input for the next layer. Fully connected layer: Applies weights over the input generated by the feature analysis to predict an accurate label.

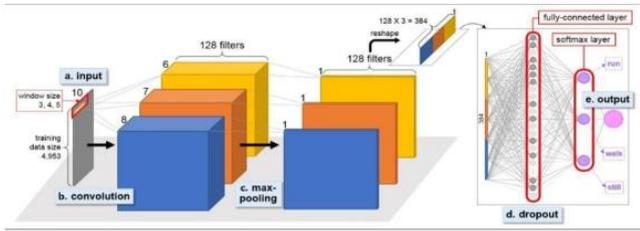


Figure2: Convolutional Neural Network General Architecture

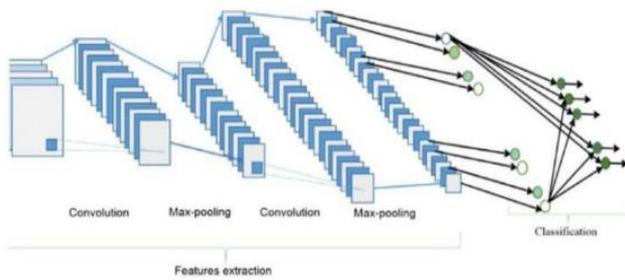


Figure2: Convolutional Neural Network Layers

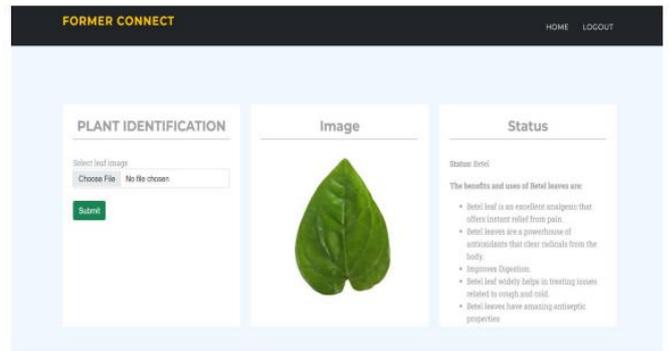
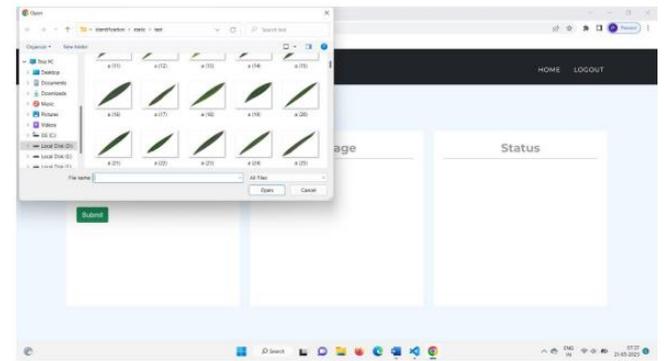
## 6.RESULTS

Results of this project are as show in following figures.

```

1 # Importing the necessary libraries
2 import tensorflow as tf
3 import numpy as np
4 import cv2
5 import os
6 import random
7 import shutil
8 import glob
9 import time
10 import sys
11 import argparse
12 import pickle
13 import math
14 import logging
15 import warnings
16 import itertools
17 import multiprocessing
18 import concurrent.futures
19 import threading
20 import queue
21 import copy
22 import json
23 import yaml
24 import argparse
25 import sys
26 import os
27 import glob
28 import time
29 import sys
30 import argparse
31 import sys
32 import argparse
33 import sys
34 import argparse
35 import sys
36 import argparse
37 import sys
38 import argparse
39 import sys
40 import argparse
41 import sys
42 import argparse
43 import sys
44 import argparse
45 import sys
46 import argparse
47 import sys
48 import argparse
49 import sys
50 import argparse
51 import sys
52 import argparse
53 import sys
54 import argparse
55 import sys
56 import argparse
57 import sys
58 import argparse
59 import sys
60 import argparse
61 import sys
62 import argparse
63 import sys
64 import argparse
65 import sys
66 import argparse
67 import sys
68 import argparse
69 import sys
70 import argparse
71 import sys
72 import argparse
73 import sys
74 import argparse
75 import sys
76 import argparse
77 import sys
78 import argparse
79 import sys
80 import argparse
81 import sys
82 import argparse
83 import sys
84 import argparse
85 import sys
86 import argparse
87 import sys
88 import argparse
89 import sys
90 import argparse
91 import sys
92 import argparse
93 import sys
94 import argparse
95 import sys
96 import argparse
97 import sys
98 import argparse
99 import sys
100 import argparse
101 import sys
102 import argparse
103 import sys
104 import argparse
105 import sys
106 import argparse
107 import sys
108 import argparse
109 import sys
110 import argparse
111 import sys
112 import argparse
113 import sys
114 import argparse
115 import sys
116 import argparse
117 import sys
118 import argparse
119 import sys
120 import argparse
121 import sys
122 import argparse
123 import sys
124 import argparse
125 import sys
126 import argparse
127 import sys
128 import argparse
129 import sys
130 import argparse
131 import sys
132 import argparse
133 import sys
134 import argparse
135 import sys
136 import argparse
137 import sys
138 import argparse
139 import sys
140 import argparse
141 import sys
142 import argparse
143 import sys
144 import argparse
145 import sys
146 import argparse
147 import sys
148 import argparse
149 import sys
150 import argparse
151 import sys
152 import argparse
153 import sys
154 import argparse
155 import sys
156 import argparse
157 import sys
158 import argparse
159 import sys
160 import argparse
161 import sys
162 import argparse
163 import sys
164 import argparse
165 import sys
166 import argparse
167 import sys
168 import argparse
169 import sys
170 import argparse
171 import sys
172 import argparse
173 import sys
174 import argparse
175 import sys
176 import argparse
177 import sys
178 import argparse
179 import sys
180 import argparse
181 import sys
182 import argparse
183 import sys
184 import argparse
185 import sys
186 import argparse
187 import sys
188 import argparse
189 import sys
190 import argparse
191 import sys
192 import argparse
193 import sys
194 import argparse
195 import sys
196 import argparse
197 import sys
198 import argparse
199 import sys
200 import argparse
201 import sys
202 import argparse
203 import sys
204 import argparse
205 import sys
206 import argparse
207 import sys
208 import argparse
209 import sys
210 import argparse
211 import sys
212 import argparse
213 import sys
214 import argparse
215 import sys
216 import argparse
217 import sys
218 import argparse
219 import sys
220 import argparse
221 import sys
222 import argparse
223 import sys
224 import argparse
225 import sys
226 import argparse
227 import sys
228 import argparse
229 import sys
230 import argparse
231 import sys
232 import argparse
233 import sys
234 import argparse
235 import sys
236 import argparse
237 import sys
238 import argparse
239 import sys
240 import argparse
241 import sys
242 import argparse
243 import sys
244 import argparse
245 import sys
246 import argparse
247 import sys
248 import argparse
249 import sys
250 import argparse
251 import sys
252 import argparse
253 import sys
254 import argparse
255 import sys
256 import argparse
257 import sys
258 import argparse
259 import sys
260 import argparse
261 import sys
262 import argparse
263 import sys
264 import argparse
265 import sys
266 import argparse
267 import sys
268 import argparse
269 import sys
270 import argparse
271 import sys
272 import argparse
273 import sys
274 import argparse
275 import sys
276 import argparse
277 import sys
278 import argparse
279 import sys
280 import argparse
281 import sys
282 import argparse
283 import sys
284 import argparse
285 import sys
286 import argparse
287 import sys
288 import argparse
289 import sys
290 import argparse
291 import sys
292 import argparse
293 import sys
294 import argparse
295 import sys
296 import argparse
297 import sys
298 import argparse
299 import sys
300 import argparse
301 import sys
302 import argparse
303 import sys
304 import argparse
305 import sys
306 import argparse
307 import sys
308 import argparse
309 import sys
310 import argparse
311 import sys
312 import argparse
313 import sys
314 import argparse
315 import sys
316 import argparse
317 import sys
318 import argparse
319 import sys
320 import argparse
321 import sys
322 import argparse
323 import sys
324 import argparse
325 import sys
326 import argparse
327 import sys
328 import argparse
329 import sys
330 import argparse
331 import sys
332 import argparse
333 import sys
334 import argparse
335 import sys
336 import argparse
337 import sys
338 import argparse
339 import sys
340 import argparse
341 import sys
342 import argparse
343 import sys
344 import argparse
345 import sys
346 import argparse
347 import sys
348 import argparse
349 import sys
350 import argparse
351 import sys
352 import argparse
353 import sys
354 import argparse
355 import sys
356 import argparse
357 import sys
358 import argparse
359 import sys
360 import argparse
361 import sys
362 import argparse
363 import sys
364 import argparse
365 import sys
366 import argparse
367 import sys
368 import argparse
369 import sys
370 import argparse
371 import sys
372 import argparse
373 import sys
374 import argparse
375 import sys
376 import argparse
377 import sys
378 import argparse
379 import sys
380 import argparse
381 import sys
382 import argparse
383 import sys
384 import argparse
385 import sys
386 import argparse
387 import sys
388 import argparse
389 import sys
390 import argparse
391 import sys
392 import argparse
393 import sys
394 import argparse
395 import sys
396 import argparse
397 import sys
398 import argparse
399 import sys
400 import argparse
401 import sys
402 import argparse
403 import sys
404 import argparse
405 import sys
406 import argparse
407 import sys
408 import argparse
409 import sys
410 import argparse
411 import sys
412 import argparse
413 import sys
414 import argparse
415 import sys
416 import argparse
417 import sys
418 import argparse
419 import sys
420 import argparse
421 import sys
422 import argparse
423 import sys
424 import argparse
425 import sys
426 import argparse
427 import sys
428 import argparse
429 import sys
430 import argparse
431 import sys
432 import argparse
433 import sys
434 import argparse
435 import sys
436 import argparse
437 import sys
438 import argparse
439 import sys
440 import argparse
441 import sys
442 import argparse
443 import sys
444 import argparse
445 import sys
446 import argparse
447 import sys
448 import argparse
449 import sys
450 import argparse
451 import sys
452 import argparse
453 import sys
454 import argparse
455 import sys
456 import argparse
457 import sys
458 import argparse
459 import sys
460 import argparse
461 import sys
462 import argparse
463 import sys
464 import argparse
465 import sys
466 import argparse
467 import sys
468 import argparse
469 import sys
470 import argparse
471 import sys
472 import argparse
473 import sys
474 import argparse
475 import sys
476 import argparse
477 import sys
478 import argparse
479 import sys
480 import argparse
481 import sys
482 import argparse
483 import sys
484 import argparse
485 import sys
486 import argparse
487 import sys
488 import argparse
489 import sys
490 import argparse
491 import sys
492 import argparse
493 import sys
494 import argparse
495 import sys
496 import argparse
497 import sys
498 import argparse
499 import sys
500 import argparse
501 import sys
502 import argparse
503 import sys
504 import argparse
505 import sys
506 import argparse
507 import sys
508 import argparse
509 import sys
510 import argparse
511 import sys
512 import argparse
513 import sys
514 import argparse
515 import sys
516 import argparse
517 import sys
518 import argparse
519 import sys
520 import argparse
521 import sys
522 import argparse
523 import sys
524 import argparse
525 import sys
526 import argparse
527 import sys
528 import argparse
529 import sys
530 import argparse
531 import sys
532 import argparse
533 import sys
534 import argparse
535 import sys
536 import argparse
537 import sys
538 import argparse
539 import sys
540 import argparse
541 import sys
542 import argparse
543 import sys
544 import argparse
545 import sys
546 import argparse
547 import sys
548 import argparse
549 import sys
550 import argparse
551 import sys
552 import argparse
553 import sys
554 import argparse
555 import sys
556 import argparse
557 import sys
558 import argparse
559 import sys
560 import argparse
561 import sys
562 import argparse
563 import sys
564 import argparse
565 import sys
566 import argparse
567 import sys
568 import argparse
569 import sys
570 import argparse
571 import sys
572 import argparse
573 import sys
574 import argparse
575 import sys
576 import argparse
577 import sys
578 import argparse
579 import sys
580 import argparse
581 import sys
582 import argparse
583 import sys
584 import argparse
585 import sys
586 import argparse
587 import sys
588 import argparse
589 import sys
590 import argparse
591 import sys
592 import argparse
593 import sys
594 import argparse
595 import sys
596 import argparse
597 import sys
598 import argparse
599 import sys
600 import argparse
601 import sys
602 import argparse
603 import sys
604 import argparse
605 import sys
606 import argparse
607 import sys
608 import argparse
609 import sys
610 import argparse
611 import sys
612 import argparse
613 import sys
614 import argparse
615 import sys
616 import argparse
617 import sys
618 import argparse
619 import sys
620 import argparse
621 import sys
622 import argparse
623 import sys
624 import argparse
625 import sys
626 import argparse
627 import sys
628 import argparse
629 import sys
630 import argparse
631 import sys
632 import argparse
633 import sys
634 import argparse
635 import sys
636 import argparse
637 import sys
638 import argparse
639 import sys
640 import argparse
641 import sys
642 import argparse
643 import sys
644 import argparse
645 import sys
646 import argparse
647 import sys
648 import argparse
649 import sys
650 import argparse
651 import sys
652 import argparse
653 import sys
654 import argparse
655 import sys
656 import argparse
657 import sys
658 import argparse
659 import sys
660 import argparse
661 import sys
662 import argparse
663 import sys
664 import argparse
665 import sys
666 import argparse
667 import sys
668 import argparse
669 import sys
670 import argparse
671 import sys
672 import argparse
673 import sys
674 import argparse
675 import sys
676 import argparse
677 import sys
678 import argparse
679 import sys
680 import argparse
681 import sys
682 import argparse
683 import sys
684 import argparse
685 import sys
686 import argparse
687 import sys
688 import argparse
689 import sys
690 import argparse
691 import sys
692 import argparse
693 import sys
694 import argparse
695 import sys
696 import argparse
697 import sys
698 import argparse
699 import sys
700 import argparse
701 import sys
702 import argparse
703 import sys
704 import argparse
705 import sys
706 import argparse
707 import sys
708 import argparse
709 import sys
710 import argparse
711 import sys
712 import argparse
713 import sys
714 import argparse
715 import sys
716 import argparse
717 import sys
718 import argparse
719 import sys
720 import argparse
721 import sys
722 import argparse
723 import sys
724 import argparse
725 import sys
726 import argparse
727 import sys
728 import argparse
729 import sys
730 import argparse
731 import sys
732 import argparse
733 import sys
734 import argparse
735 import sys
736 import argparse
737 import sys
738 import argparse
739 import sys
740 import argparse
741 import sys
742 import argparse
743 import sys
744 import argparse
745 import sys
746 import argparse
747 import sys
748 import argparse
749 import sys
750 import argparse
751 import sys
752 import argparse
753 import sys
754 import argparse
755 import sys
756 import argparse
757 import sys
758 import argparse
759 import sys
760 import argparse
761 import sys
762 import argparse
763 import sys
764 import argparse
765 import sys
766 import argparse
767 import sys
768 import argparse
769 import sys
770 import argparse
771 import sys
772 import argparse
773 import sys
774 import argparse
775 import sys
776 import argparse
777 import sys
778 import argparse
779 import sys
780 import argparse
781 import sys
782 import argparse
783 import sys
784 import argparse
785 import sys
786 import argparse
787 import sys
788 import argparse
789 import sys
790 import argparse
791 import sys
792 import argparse
793 import sys
794 import argparse
795 import sys
796 import argparse
797 import sys
798 import argparse
799 import sys
800 import argparse
801 import sys
802 import argparse
803 import sys
804 import argparse
805 import sys
806 import argparse
807 import sys
808 import argparse
809 import sys
810 import argparse
811 import sys
812 import argparse
813 import sys
814 import argparse
815 import sys
816 import argparse
817 import sys
818 import argparse
819 import sys
820 import argparse
821 import sys
822 import argparse
823 import sys
824 import argparse
825 import sys
826 import argparse
827 import sys
828 import argparse
829 import sys
830 import argparse
831 import sys
832 import argparse
833 import sys
834 import argparse
835 import sys
836 import argparse
837 import sys
838 import argparse
839 import sys
840 import argparse
841 import sys
842 import argparse
843 import sys
844 import argparse
845 import sys
846 import argparse
847 import sys
848 import argparse
849 import sys
850 import argparse
851 import sys
852 import argparse
853 import sys
854 import argparse
855 import sys
856 import argparse
857 import sys
858 import argparse
859 import sys
860 import argparse
861 import sys
862 import argparse
863 import sys
864 import argparse
865 import sys
866 import argparse
867 import sys
868 import argparse
869 import sys
870 import argparse
871 import sys
872 import argparse
873 import sys
874 import argparse
875 import sys
876 import argparse
877 import sys
878 import argparse
879 import sys
880 import argparse
881 import sys
882 import argparse
883 import sys
884 import argparse
885 import sys
886 import argparse
887 import sys
888 import argparse
889 import sys
890 import argparse
891 import sys
892 import argparse
893 import sys
894 import argparse
895 import sys
896 import argparse
897 import sys
898 import argparse
899 import sys
900 import argparse
901 import sys
902 import argparse
903 import sys
904 import argparse
905 import sys
906 import argparse
907 import sys
908 import argparse
909 import sys
910 import argparse
911 import sys
912 import argparse
913 import sys
914 import argparse
915 import sys
916 import argparse
917 import sys
918 import argparse
919 import sys
920 import argparse
921 import sys
922 import argparse
923 import sys
924 import argparse
925 import sys
926 import argparse
927 import sys
928 import argparse
929 import sys
930 import argparse
931 import sys
932 import argparse
933 import sys
934 import argparse
935 import sys
936 import argparse
937 import sys
938 import argparse
939 import sys
940 import argparse
941 import sys
942 import argparse
943 import sys
944 import argparse
945 import sys
946 import argparse
947 import sys
948 import argparse
949 import sys
950 import argparse
951 import sys
952 import argparse
953 import sys
954 import argparse
955 import sys
956 import argparse
957 import sys
958 import argparse
959 import sys
960 import argparse
961 import sys
962 import argparse
963 import sys
964 import argparse
965 import sys
966 import argparse
967 import sys
968 import argparse
969 import sys
970 import argparse
971 import sys
972 import argparse
973 import sys
974 import argparse
975 import sys
976 import argparse
977 import sys
978 import argparse
979 import sys
980 import argparse
981 import sys
982 import argparse
983 import sys
984 import argparse
985 import sys
986 import argparse
987 import sys
988 import argparse
989 import sys
990 import argparse
991 import sys
992 import argparse
993 import sys
994 import argparse
995 import sys
996 import argparse
997 import sys
998 import argparse
999 import sys
1000 import argparse

```



- Here user has to Login with user ID and Password.
- Then select the leaf which is to be identified.
- The CNN algorithm used identifies leaf and it displays name of plant and its medicinal features.

## 7.APPLICATIONS

Applications of this project are listed as bellow

- 1.Agriculture: In agriculture we can use this project to identify plant diseases
2. Conservation: We can use this project to conserve some traditional ayurvedic plants.
3. Forestry: We can use this project to identify medicinal plants and preserve them.
- 4.Horticulture: Here we can use this project to Identify flowers in gardening.
- 5.Education: It can be used in Botony labs to demonstrate identification and features of plants.

## CONCLUSIONS

Plants are necessary for human survival. Herbs, particularly, are employed by indigenous populations as folk medicines from old period .Recent improvements in analytical technology have made it much easier to identify herbs depending on scientific evidence. As a result, a simple and reliable method for identifying herbs is required. Herbal identification anticipated to benefit from the combination of computation and statistical examination. This nondestructive technique will be the preferred approach for quickly identifying

herbs, especially for individuals who cannot able to use expensive analytical equipment

## ACKNOWLEDGEMENT

We take this opportunity to Acknowledge Our principal head of the department and All authors take part in the discussion of the work described in this article. All The authors would like to thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

## REFERENCES

1. Skanda H N , Smitha S Karanth , Suvijith S , Swathi K S, 2019, Plant Identification Methodologies using Machine Learning Algorithms, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 08, Issue 03 (March – 2019),
2. P. M. Kumar, C. M. Surya and V. P. Gopi, "Identification of ayurvedic medicinal plants by image processing of leaf samples," 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Kolkata, 2017, pp. 231-238. DOI: 10.1109/ICRCICN.2017.8234512
3. H. A. Chathura Priyankara and D. K. Withanage, "Computer assisted plant identification system for Android," 2015 Moratuwa Engineering Research Conference (MERCon), Moratuwa, 2015, pp. 148-153. DOI: 10.1109/MERCon.2015.7112336
4. Rafael Rojas-Hernández and Asdrúbal López-Chau, "Plant identification using new geometric features with standard data mining methods", Networking Sensing and Control (ICNSC) 2016 IEEE 13th International Conference on, pp. 1-4, 2016.
5. Che Hussin, N. A., Jamil, N., Nordin, S., & Awang, K. (2013). Plant species identification by using scale invariant feature transform (SIFT) and grid based colour moment (GBCM). In 2013 IEEE Conference on Open Systems, ICOS 2013 (pp. 226-230). [6735079]
6. K. Li, Y. Ma and J. C. Príncipe, "Automatic plant identification using stem automata," 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP), Tokyo, 2017, pp. 1-6. DOI: 10.1109/MLSP.2017.8168147. J. X. Du, D. S. Huang, X. F. Wang, and X. Gu, "Computer-aided plant species identification (CAPSI) based on leaf shape matching technique," *Transactions of the Institute of Measurement and Control*, vol. 28, no. 3, pp. 275–285, 2006.
7. S. Prasad, P. Kumar, and R. C. Tripathi, "Plant leaf species identification using curvelet transform," in *Proceedings of the 2011 2nd IEEE International Conference on Computer and Communication Technology*, pp. 646–652, Harbin, China, August 2011.
8. S. G. Wu, F. S. Bao, E. Y. Xu, Y. X. Wang, Y. F. Chang, and Q. L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology*, pp. 11–16, Giza, Egypt, December 2007.
9. H. Fu, Z. Chi, D. Feng, and J. Song, "Machine learning techniques for ontology-based leaf classification," in *Proceedings of the 8th IEEE International Conference on Control, Automation, Robotics and Vision Conference*, vol. 1, pp. 681–686, Niigata, Japan, April 2004.
10. F. Han, X. Qiao, Y. Ma et al., "Grass leaf identification using dbN wavelet and CILBP," *Advances in Multimedia*, vol. 2020, Article ID 1909875, 8 pages, 2020.
11. X. Gu, J. X. Du, and X. F. Wang, "Leaf recognition based on the combination of wavelet transform and Gaussian interpolation," in *Proceedings of the ICIC 2005: Advances in Intelligent Computing*, pp. 253–262, Anhui, China, August 2005.
12. S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, 2022.
13. Hossain and M. A. Amin, "Leaf shape identification based plant biometrics," in *Proceedings of the 2010 13th IEEE International Conference on Computer and Information Technology*, pp. 458–463, Bradford, UK, December 2010.
14. A. H. Kulkarni, H. M. Rai, K. A. Jahagirdar, and P. S. Upparamani, "A leaf recognition technique for plant classification using RBPNN and Zernike moments," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, pp. 984–988, 2013.