Medi-Plant: A Deep Learning Approach for Medicinal Plant Classification with Pix2Pix Generative Adversarial Network

Swathika P

swathikap@mepcoeng.ac.in

Mepco Schlenk Engineering College https://orcid.org/0000-0002-1833-2359

Ajithar A

Mepco Schlenk Engineering College

Hishaamudin Z

Mepco Schlenk Engineering College

Nitheeswaran E

Mepco Schlenk Engineering College

Research Article

Keywords: Medicinal plant, deep learning, classification, convolutional neural networks, cGAN, image-to-image translation, feature extraction, transfer learning, ensemble learning, leaf images

Posted Date: May 3rd, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4245022/v1

Medi-Plant: A Deep Learning Approach for Medicinal Plant Classification with Pix2Pix Generative Adversarial Network

Mrs. P. Swathika[1*], Ajithar A[1†], Hishaamudin Z[1†], Nitheeswaran E[1†]
[1]Department of Artificial Intelligence and Data Science,
Mepco Schlenk Engineering College (Autonomous), Sivakasi, 626005, Tamil Nadu, India
****Corresponding author(s). E-mail(s): swathikap@mepcoeng.ac.in;***
Contributing authors : ajitharmahes@gmail.com; hishaaofficial1107@gmail.com; nitheeswaran56@gmail.com;
[†]These authors contributed equally to this work.

Abstract

Background
Biodiversity conservation is crucial due to the risk of extinction faced by many plant species. Traditional medicinal systems heavily rely on the diverse array of plants, offering an alternative to manufactured medications and promoting healthy living. Despite the significance of these plants, datasets for therapeutic herbs are not readily available. To address this gap, this study proposes an automatic system for identifying medicinal plants based on computer vision and deep learning techniques, utilizing various neural network approaches.

Results
The study introduces the Medi-plant dataset, comprising 6000 leaf images from 50 different Indian plant species. To validate the dataset, pre-trained deep convolutional neural network architectures including MobileNetV2, ResNet-50, and Xception were employed. The proposed Medi-Plant model, leveraging all three architectures, achieved an impressive accuracy of 97.96%.

Conclusions
The findings demonstrate the effectiveness of the Medi-plant dataset and the proposed Medi-Plant model in accurately identifying medicinal plants. Additionally, a cross-platform application named Medi-Plant Identification was developed, capable of swiftly identifying herb images and providing pertinent information from the database. By continuing to expand the dataset, this research aims to benefit stakeholders and society at large by fostering awareness and understanding of herbs and their therapeutic properties.

Keywords – Medicinal plant, deep learning, classification, convolutional neural networks, cGAN, image-to-image translation, feature extraction, transfer learning, ensemble learning, leaf images

1. Background

India is a developing nation, with a wide variety of species. Biodiversity conservation protects the sustainability of natural resources by increasing ecosystem productivity. The State of Plants report highlights the vast number of plant species (approximately 390, 000) [1], which are known in modern science. In the twentieth century, the poisonous and dose-dependent impact of medicinal plants resulted in great danger to the use of herbal medicines [2]. Identifying and classifying such a multitude of species poses a challenge for botanists and experts, especially considering the subtle differences between similar species. Using a fine-grained categorization strategy is crucial, particularly when certain plant species exhibit a great deal of similarity. Since allopathic and surgical treatments respond quickly to illnesses, this notion has significantly shifted people's views toward them. Unfortunately, the low cost and lower likelihood of adverse effects associated with herbal remedies have focused on global pharmacopeias to characterize and encapsulate many active plant elements and synthesize nanomedicines to accelerate the search for new drugs. Over time, a more advanced medical system can be achieved through the fusion of traditional herbal knowledge with contemporary technical methods.

The leaves of plants contain substantial information about their respective species, offering valuable insights. Plants, integral to human existence since their creation, serve diverse purposes including food, medicine, and shelter. Plants are the primary source of both food and medicinal materials. We rely on plants and the things they produce for daily nourishment. Most people depend on plant-based remedies to address the requirements for healthcare, such as supplemental and alternative medicine, or alternative medicine. Humans must possess solid knowledge of plants to differentiate and identify various plant species. This will support and safeguard the pharmaceutical industry from the environment. Therefore, there is a need for sustainable development. Moreover, the threat of extinction exacerbated by global warming necessitates the preservation of both non-endangered and endangered plant species. The conventional method of identifying Indian medicinal plants is time-consuming, tedious, and challenging because they may be incorrect often. The variety of species in the Medi-Plant dataset is displayed in Figure 1. Leveraging image-processing techniques for the two-dimensional nature of plant leaves allows automatic identification. Recently, Deep Learning, particularly Convolutional Neural Networks (CNNs), has proven to be effective in addressing computer vision problems, including plant classification. This eliminates the need for extensive domain expertise and intricate feature extraction by botanists. Feature maps were created through consecutive convolution operations to extract discriminative patterns from individual plant leaves. Various deep learning-based image recognition methods have gained attention, with potential applications in botany and agriculture. The evolution of computer vision methods has been notable in automated plant recognition, particularly in industrial applications on mobile devices. In particular, the efficacy of deep learning methods has led to substantial achievements in this domain. Researchers have predominantly focused on extracting local features from leaves, flowers, and bark to facilitate plant classification by leveraging characteristic variations in leaves.

Fig. 1 Sample leaf images of the Medi-Plant Dataset consist of 50 classes

Table 1: Overview of Key Leaf Features Employed in Classification

Main Features	Sub-Features
Shape	Area Perimeter Diameter Aspect ratio Ovate Elliptical Lanceolate
Color	Intensity Histogram Mean and Standard Deviation of Red, Green and Blue (RGB) channels
Texture	Contrast Coarseness Homogeneity
Pattern	Vein Margin Symmetric

Table 1 comprehensively summarizes the widely researched leaf features utilized in the classification. Distinctive characteristics encompassing shape, color, and texture are fundamental factors that differentiate plant leaves. Additionally, various sub-features contribute to these primary distinctions, and Table 1 organizes them within the overarching categories of shape, color, and texture. This work introduces an effective approach based on an ensemble of deep-learning techniques for the efficient classification of plant leaf images.

2. Related Work

A survey serves as a scientific method for gathering data from a large population, allowing for generalization. Several studies contributed to the development of models and algorithms for plant identification and classification. Herdiyeni et al. (2020) [3] introduced a mobile-based system utilizing leaf images for medicinal plant identification. Their model incorporates leaf characteristics and employs a probabilistic neural network for plant recognition. Y. R. Azee et al. (2020) [4] proposed a learning algorithm for recognizing various herbs in Sri Lanka. They utilized the inception model for feature extraction and achieved a maximum accuracy of 95.5% using ResNet among other deep learning models. Karen Simonyan et al. (2020) [5] explored the use of extremely deep CNN models for large image classification, emphasizing the impact of representation depth on classification accuracy. Sandeep Kumar, V. T. E., et al. (2020) [6] developed software using a training set of ten plant species and fifty leaf images. Liu, Albert et al. (2019) [7] created a leaf recognition system using CNN codes and SVM for classification, mainly effective for smooth images with well-aligned leaves. Patel et al. (2019) [8] focused on flower color and texture, utilizing image filtering, classification, region recognition, and feature extraction for accurate flower identification and classification. Habiba et al. (2019) [9] classified leaves using deep convolutional neural networks and various models. Their approach included eight datasets for training, with VGG16 achieving classification accuracy of approximately 96%.

Sabu et al. (2019) [10] highlighted the importance of image pre-processing in plant image analysis, covering standardization, conversion to grayscale or binary, noise removal, contrast adjustment, and graph modification. Jing Wei Tan et al. (2019) [11] proposed DLeaf, a CNN-based model for plant leaf classification, incorporating edge detection for venation segmentation and achieving a classification accuracy of 94.88%. Divya Tomar et al. (2017) [12] suggested a leaf species management system utilizing a dual SVM classifier based on a direct acyclic graph. Their approach involved a Hierarchical Feature Selection (HFS) method to identify discriminating features. Where leaves are so plainly observable, variations in leaf features are ideally used in automated plant identification systems using computer vision techniques. relative to other plant organs, it is more observable and comprehensible. Kadir et al. [13], Cope et al. [14], and Ahmed et al. [15] provide in-depth analysis of various automated plant identification techniques. Nevertheless, given the tremendous diversity of botanical data, plant identification is still regarded as a difficult and unsolvable subject since all traditional computer vision uses handcrafted algorithms that rely on selected natural traits, Jin et al. [16], employed a classical image processing chain of image binarization to separate the background and the leaf from a leaf image, detection of contours and contour corners, and geometrical derivations of special leaf tooth features.

Plant species can be distinguished from one another according to their various leaf forms, as demonstrated by Neto et al. [17] using Elliptic Fourier and discriminant analysis. Two methods for shape modeling based on the centroid radius and invariant-moments models were suggested in Reference [18]. Du et al. [19] also went on to offer a technique based on combining geometrical and invariant moments data to derive morphological structures of leaves.

Due to the development of efficient general-purpose computing on graphics processing units (GPUs provide high degrees of parallelization) and the availability of large-scale image data (in publicly available datasets, on the internet, in social media, in (specialized) social networks, etc.), deep learning convolutional neural networks (CNNs) have made significant strides in computer vision in recent years, particularly in the field of visual object categorization, Krizhevsky et al. [20]. Lee et al. [21] demonstrated a CNN method for taxonomic recognition based on leaf images and found an average accuracy of 99.7%. CNN was employed by Zhang et al. [22] to categorize the Flavia dataset, with a 94.69% accuracy rate.

Goodfellow et al. [23] initially suggested the Generative Adversarial Network (GAN), and they have achieved great success in the Deep Learning discipline. Like this, the many layers incorporated into a model enable a deep Convolutional Neural Network (CNN) to gradually extract higher-level features from the input images supplied to it. Since its creation a few years ago, the Residual Network [24] architecture has proven to be a dependable architecture, as evidenced by its first-place finish on the ILSVRC 2015 classification job. The technique of creating new training data from the available, current data is known as data augmentation [25]. This is usually accomplished by applying annotation-preserving transformations to the input data, like scaling, flipping the photos vertically and horizontally, applying random zooming, and randomly rotating, deforming, or translating the image. Conditional Generative Adversarial Networks (cGANs), which extend GANs into a conditional model, were introduced by Mirza and Osindero [26]. Controllable image synthesis is made possible by this conditional variant of GANs, or cGANs, which lets users create images based on a variety of conditional inputs, including user sketches, class labels, or textual descriptions.

The generator G and discriminator D in cGANs are dependent on some additional data, C is added as an additional input to both G and D to accomplish this. CGANs offer more control over the type of data that are generated, in contrast to vanilla GANs which may or may not have these controls. It increases the use of cGANs in image synthesis and editing applications. The goal function, which is identical to Pix2Pix's [27] A generator network that creates synthetic leaf pictures and a discriminator network that determines if an image is the real leaf image or a leaf image created by the generator network make up the cGAN model. The generator network has 15 layers with skip connections and was modeled after U-Net [28]. When the encoder-decoder structure had n layers total, skip connections were strategically added between layers I and N-I to prevent losing low-level information that would otherwise be lost due to progressive downsampling, also known as the vanishing gradient problem, which could occur when training very deep convolutional networks. As demonstrated in [29], the goal function reflects the L1 distance between the generated image and the ground truth image after the generator has been trained. The most advanced technique available today for a variety of tasks, such as segmentation, localization, detection, and classification, is Convolutional Neural Networks [30].

The most common applications of their structures are in image processing and other situations requiring shift-invariance or covariance. CNNs use convolutional kernels for the layer-wise affine transformation to capture this translational invariance since they are motivated by the idea that an object on an image can be moved within the image and remain the same object. The results demonstrated excellent performance across the range of assessment metrics that we employed.

3. Dataset Description

Identifying medicinal plants is crucial for various fields such as pharmacology, botany, and traditional medicine. With the rise of technological advancements, particularly in machine learning and image processing, medicinal plant identification has become more efficient and accurate. We have collected Three datasets –Mepco Tropic leaf, Indian segmented leaf, and Swedish dataset – that provide valuable resources for researchers and practitioners in this domain. Collectively, these three datasets represent a global effort to document and understand medicinal plants that form the backbone of traditional medicinal systems worldwide. By combining botanical expertise with cutting-edge technologies, researchers can unlock the therapeutic potential of these plants and pave the way for discoveries in healthcare and pharmacology. Moreover, these datasets play a pivotal role in promoting biodiversity conservation and sustainable practices, ensuring that future generations can benefit from the wealth of medicinal plants that our planet must offer.

I. Mepco Tropic Leaf Dataset

Mepco Tropic Leaf is a recently developed, open-access annotated database of photos of leaves from Indian plants and consists of 75 classes. Agronomists, practitioners of Ayurvedic medicine, and the Ayurvedic pharmaceutical business all depend on the accurate identification of medicinal plants, which is why this database was created. A standardized database that is specifically designed for Indian Ayurvedic plant species is lacking, despite the abundance of freely available plant leaf databases. Mepco Tropic Leaf seeks to close this gap by offering an extensive tool for the identification of Indian medicinal plants via labeled photos of leaves.

Fig. 2 Sample leaf images of the Mepco Tropic leaf Dataset

II. Segmented Medicinal leaf Dataset

The Segmented Medicinal Leaf dataset consists of 30 classes of meticulously curated leaf samples, each with a dimension of 256 × 256 pixels. This dataset serves as a comprehensive resource for researchers and practitioners in the rich domain of traditional Indian medicine. Every leaf is a unique species of plant known for its therapeutic qualities. This dataset is a useful resource for studying herbal medicine, botany, and computer vision because of its high-resolution photography and varied depictions of medicinal plants.

Fig. 3 Sample leaf images of the Segmented Medicinal Dataset

4. Methods

The proposed technology for medicinal plant identification encompasses four crucial modules designed to enhance the accuracy and efficiency of plant recognition. The Integration module serves as the backbone and seamlessly combines the functionalities of subsequent modules. Augmentation, which employs Conditional Generative Adversarial Networks (CGAN), refines the dataset by generating diverse yet realistic synthetic images, thereby improving model robustness Preprocessing, the second module, employs advanced techniques to eliminate background noise and ensure a clean input for subsequent analysis. Feature extraction followed, utilizing state-of-the-art deep learning architectures such as MobileNetV2, ResNet-50, and Xception. These networks discern intricate patterns and relevant features from plant images, contributing to a comprehensive understanding of the unique characteristics of each specimen. The final module involves Average Ensemble, harmonizing the outputs from the diverse feature extractors to create a more robust and accurate model. The system, empowered by this ensemble approach, predicts medicinal plant species based on their leaves. Each prediction is accompanied by a detailed description, providing valuable information on the identified plant's properties and medicinal uses. This comprehensive approach ensures the Medicinal Plant Identification System delivers precise and informative results, advancing the field of plant identification for medicinal purposes.

Table 2: Configuration of the machine used.

Name	Parameter
Memory	64 GB
Processor	AMD Ryzen 7 5800H with Radeon Graphics @3.20 GHz
Server Model	HP Pavilion Gaming Laptop
Graphics	NVIDIA GeForce RTX 3050 (4 GB GDDR6 dedicated)
OS	Windows 11
Language	Python 3
Framework	Tensorflow, and Keras

4.1 Data Integration

A crucial phase in the creation of the Medicinal Plant Identification System integrates three different datasets: the Indian Segmented dataset (which has 30 classes) and Mepco Tropic dataset (75 classes that we have taken 20 classes). The goal of this integration method is to provide a unified dataset that represents regional variations in plant varieties. With approximately 5000 photos in total, the combined dataset is a valuable tool for system testing and training. A standardization method was used for the images from all three datasets to guarantee homogeneity and the best model performance. Every image was resized to a uniform 256x256 size, resulting in a coherent and standardized dataset. This is a crucial step in reducing image-size fluctuations and enabling the smooth integration of various botanical data sources. The resultant integrated dataset established the foundation for a strong medicinal plant identification system with a variety of classes and consistent image dimensions. In addition to reflecting the diversity of plant species found in various geographical areas, this cohesive collection of botanical photos guarantees that the system can provide precise and comprehensive identification based on a common image format.

4.2 Pix2Pix Based Method of Balancing and Augmentation of The Training Data

In deep learning applications such as plant species classification, data balancing is an important topic that has been extensively researched. This is motivated by the need for a strong algorithm that enables efficient classification and supports good generalization. However, cost-effectiveness, usability, accuracy, and sensitivity are crucial factors that need to be considered. Generating extra training data from existing available data is known as data augmentation [1]. To achieve this, annotation-preserving modifications are applied to the input data. These transformations include random zooming, scaling, flipping, image deformation or rotation, and several other operations. Theoretically, an "infinite" amount of training data can be generated by augmenting the available data owing to the random nature of the data augmentation.

In this paper, we used conditional Generative Adversarial Networks (cGANs) to perform data augmentation, which can overcome the issue of insufficient data and imbalanced distribution and improve model performance for tasks such as plant species classification, detection, and recognition. We investigated the GANs in the context of conditional settings. Conditional GANs (cGANs), such as GANs, learn a conditional generative model in addition to a generative data model [2]. For image-to-image translation jobs in which we condition an input image and generate an equivalent output image, cGANs are appropriate. The generator uses noise input to create false images. The discriminator receives the fictitious images produced by the generator and uses them to differentiate between real and fake images through binary classification. The discriminator was trained to maximize the likelihood of accurately differentiating between the actual and fraudulent images. Equation (1)[2] simultaneously simulates the generator to minimize log (1 D (G (z))).

$$min_G max_D V(D,G) = E_{x\sim Pdata(x)}[logD(x)] + E_{z\sim Pz(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (1)$$

where $Pz(z)$ represents the noise distribution, and $Pdata(x)$ represents the genuine data distribution. Assuming a given noise (z), the discriminator's estimate of the probability that a real data instance (x) is real is $D(x)$), generator's output is $G(z)$, and discriminator's estimate of the probability that a fake instance is real is $D(G(z))$. The generator can produce false images that resemble original authentic photos through this iterative, competitive training process. However, because it generates images based on the input noise z, it is difficult to regulate the images generated by the generator, and it is challenging to generate high-resolution images. To address this issue, conditional Generative Adversarial Networks (cGANs), which extend GANs into conditional models, were introduced by Mirza and Osindero [2]. These conditional GANs, called cGANs [2], provide customizable picture synthesis, allowing a user to synthesize images with various conditional inputs such as class labels, user doodles, or textual descriptions. Discriminators D and G in cGANs depend on additional information, c. C is added as an additional input to both G and D. Unlike vanilla GANs, which may not have these controls, cGANs offer more control over the type of data created. This has increased the popularity of cGANs in applications that involve picture synthesis and manipulation. Figure (CGAN structure) shows the construction of cGANs. The objective function is the same as that of Pix2Pix [3] and is represented by:

$$L_{pix2pix}(G,D) = E_{x,y\sim pdata(x,y)}[logD(x)] + E_{x\sim pdata(x),z\sim p_z(z)}[\log(1 - D(x, G(x,z))))] \quad (2)$$

A generator network that creates synthetic leaf pictures and a discriminator network that determines whether an image is a real leaf image or a leaf image created by the generator network constitute the cGAN model. The architectures are described in detail below:

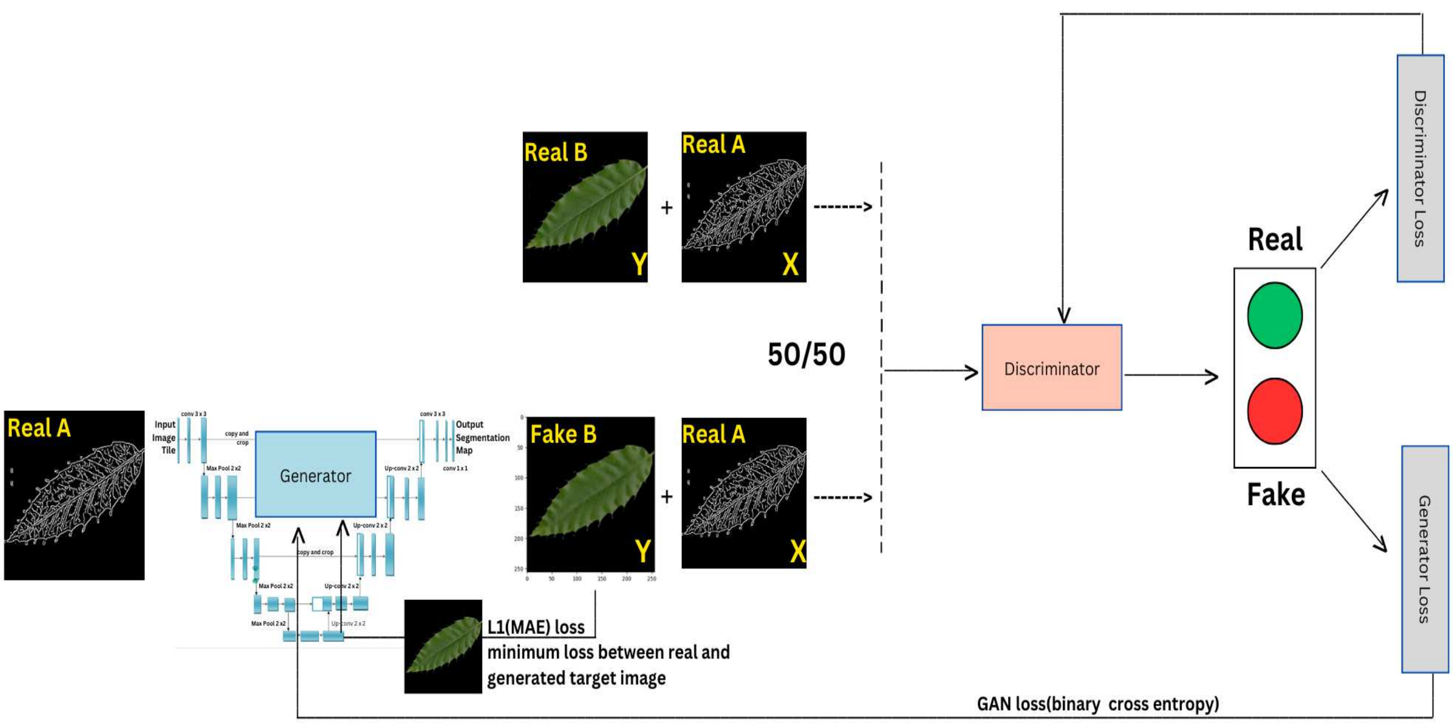

Fig. 4 Pix2Pix Architecture

1) The Generator

The generator network had 15 layers with skip connections and was modeled using U-Net [4]. When the encoder-decoder structure had n layers in total, skip connections were strategically added between layers i and $n\text{-}i$ to prevent the loss of low-level information that would otherwise be lost due to progressive downsampling, also known as the vanishing gradient problem, which could occur when training very deep convolutional networks. Low-level input information is effectively preserved in the generator output using this strategy. By using the learned features, the generator can create more realistic leaf images. When the generator is trained, the L1 (Mean Absolute Error) distance (Equation (3)) between the ground-truth image and generated image is respected in the objective function, as indicated in Equation (4) [5]. As a result, low-level information is reinforced, leading to the generation of leaf pictures that are more lucid than those produced by the traditional cGAN [3]. In addition, the discriminator was modified to incorporate the PatchGAN concept to avoid blurring of the images produced by the generator. This technique aims to categorize discriminator input images into N x area patches [3].

$$L_{L1}(G) = E_{x,y,z}[\quad ||y - G(x,z)||_1] \tag{3}$$

$$G^* = argmin_G max_D L_{cGAN} + \lambda L_{L1}(\text{G}) \tag{4}$$

Conditioned leaf images of size 256 × 256 × 3 (height × width × channel) were fed into the generator module. The feature maps were computed with a 5 × 5 filter from the 1^{st} to 8^{th} convolutional layers within the encoder. This special design was designed to avoid the need for a separate pooling layer, such as max-pooling. The size of the feature map was reduced using special padding and a stride of 2 × 2. The feature map, which is reduced to 1 × 1 × 512 by the 8^{th} convolutional layer, is upsampled through the 1^{st} to 8^{th} transposed convolutional layers within the decoder. When the generator model has n layers overall, skip connections are created between levels i and $n\text{-}i$ to prevent loss of low-level information.

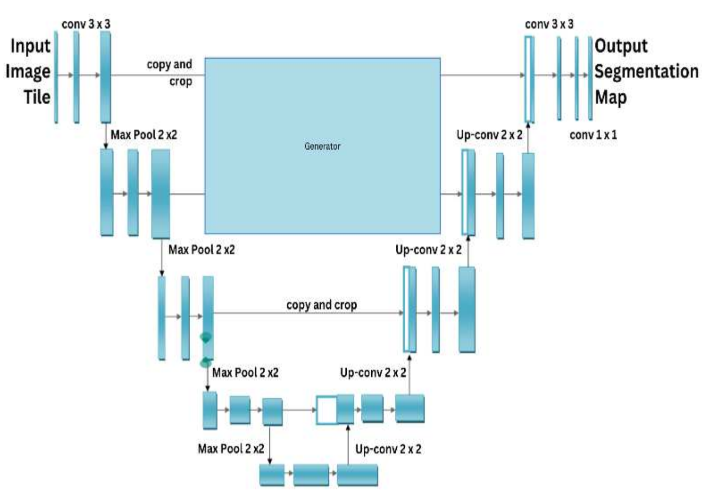

Fig. 5 Generator Architecture of Pix2Pix

We used the conventional method from [2] to optimize our networks, alternating between one gradient descent step on D and one step on G. We train to maximize $log\ D(x, G(x,z))$ instead of training G to minimize $\log(1 - D(x, G(x,z)))$as stated in the original GAN paper [2]. Furthermore, when we optimize D, we split the objective by 2, slowing down D's learning rate compared to G. We employed the Adam solution [6] with a learning rate of 00002 and momentum parameters of β_1 = 05 and β_2 = 0999, using mini batch SGD. We operated the generator net in the same way as in the training phase at the inference time. This is not the standard technique because we apply batch normalization [7] using the test batch's data instead of the training batch's aggregated statistics, and we apply dropout at the test time. This method of batch normalization—known as "instance normalization"—has been shown to work well for picture production jobs when the batch size is set to 1 [8]. Depending on the experiment, we employed batch sizes of one–ten in our investigations. L1 alone produces findings that are reasonable yet blurry. Although it generates visual artifacts in some applications, cGAN alone (setting λ = 0 in Eq. (4)) produces considerably sharper results. These artifacts were reduced by adding the two terms (with λ = 100). Again, U-Net performed better when both the encoder-decoder and U-Net were trained using L1 loss.

We aim to train a Generative Adversarial Network (GAN) using both segmented versions and real images. In contrast to using a U-Net where we typically provide a real image and its corresponding mask for segmentation, in this scenario, we utilize the mask to generate a realistic-looking leaf image. Essentially, we will input a mask (in this case, a canny edge image of a leaf), and the GAN will generate an image that closely resembles a real leaf. This mask can be anything resembling a leaf, such as a sketch drawn with a paintbrush. The process involves training the GAN using the canny edge image of a leaf as the input mask and the corresponding real leaf image as the target. The canny edge image is generated with a low threshold of 10 and a high threshold of 20.

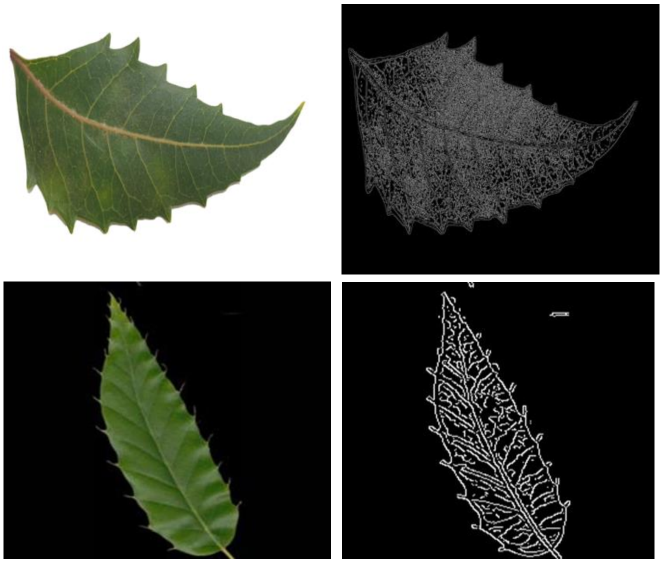

Fig. 6 Real Leaf and Canny Leaf

2) *The Discriminator*

The discriminator network is a convolutional four layers. To distinguish between an actual leaf image and a generated leaf image, the input image was split into tiny patches. This improves the ability of the generator to capture the fine features in the leaf photos. The discriminator module of our cGAN model, which determines whether the leaf images produced by our generator module are real or fraudulent, is shown in Fig 7. As stated in Equation (2), the discriminator receives as input a pair of pictures that are either a geometric center image and an image that uses extra information (actual image), or an image made by the generator and an image that uses extra information (false image). The four convolutional layers caused the input image to shrink to a 32 × 32 × 512 size. Using linear regression and the sigmoid function of the fully connected layer, a reduced feature map, which represents the final value indicating whether the image is real or fake, is created. The 256 × 256 Image GAN delivers results that are visually similar to the 70 × 70 Patch GAN, but somewhat worse in quality according to our FCN-score metric. The 70 × 70 Patch GAN forces outputs that are crisp, even if erroneous, in both spatial and spectral (colorfulness) dimensions.

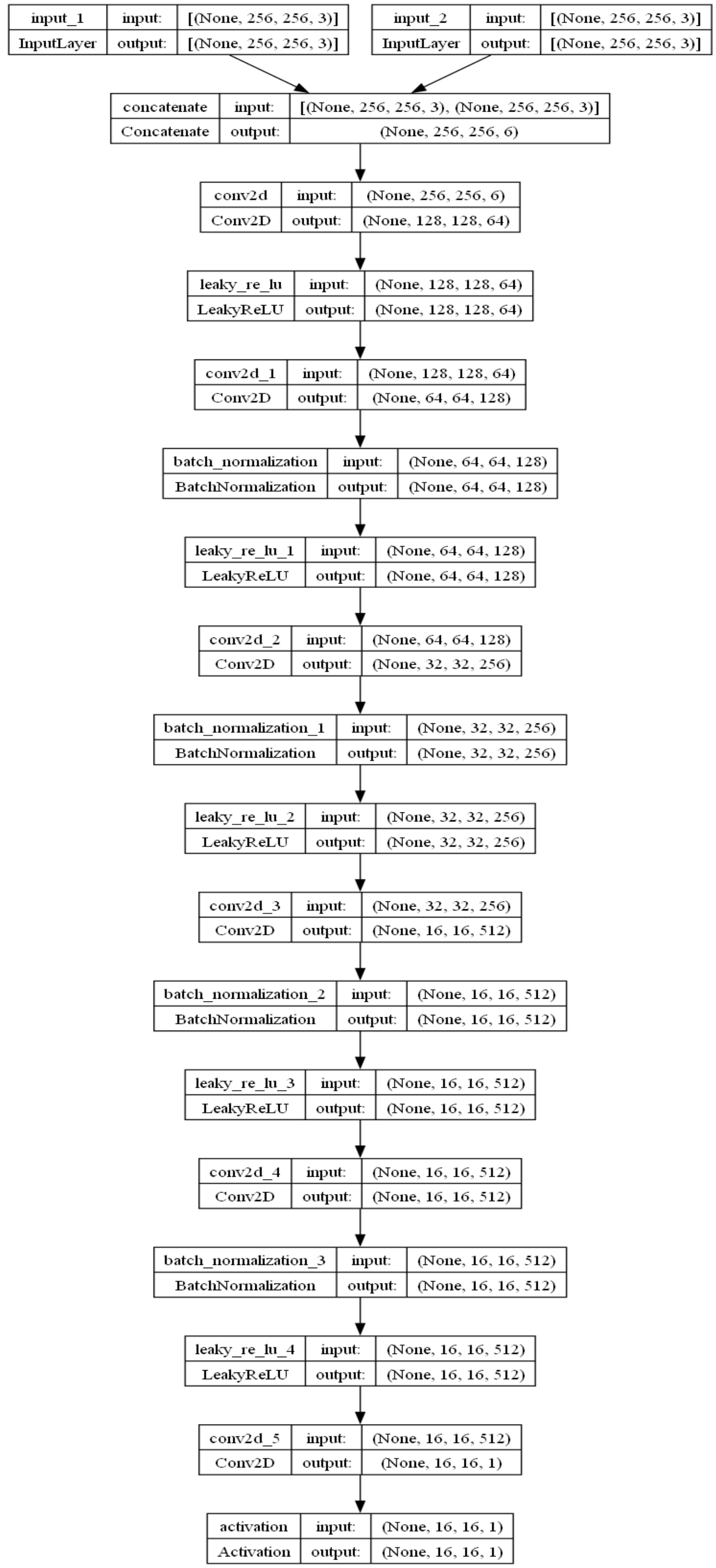

Fig. 7 Discriminator Architecture of Pix2Pix GAN

3) The Combined cGAN Model

The two networks functioned antagonistically as the training progressed; the discriminator became increasingly adept at differentiating the images, and the generator produced increasingly leaf-like images. The loss function is also important for synthesizing more realistic leaf-like images. We employed cross-entropy for the adversarial loss in the generator and discriminator networks. We employed L1 distance loss for the content loss of the generator network to create synthetic images. The generator gains knowledge of mapping, or *{x, z → y},* between the random noise vector z, source picture x, and target image y. The discriminator distinguishes between the actual and fraudulent y|x labels. A block diagram of the cGAN [2] model used to supplement the data is shown in Figure 4(cgan arch). In Section V of this paper, the results are presented in greater detail.

4.3 Data Pre-Processing

Data preprocessing is a crucial technique employed in the development of a Medicinal Plant Identification System to enhance the quality and utility of an image dataset. One key aspect of this process is the application of the rembg function in computer vision, which is a powerful tool for efficiently removing the background from images. This step is instrumental in isolating plant specimens, eliminating irrelevant visual elements, and creating a clean, focused representation for subsequent analysis.

In addition to background removal, the preprocessing pipeline incorporates an image-sharpening technique to enhance feature extraction. By utilizing advanced image-processing algorithms, the system increases the sharpness of each image, making it easier to discern intricate details and extract meaningful features. This sharpening not only aids in the identification of unique botanical characteristics but also contributes to the overall effectiveness of the feature extraction process. Through a combination of background removal and image sharpening, the data preprocessing stage plays a pivotal role in optimizing the quality of the dataset. This refined dataset, devoid of distractions and enriched with sharpened features, ensured that the subsequent stages of the Medicinal Plant Identification System operated on a foundation that promoted accurate and efficient plant recognition based on distinctive leaf features.

Fig. 8 Data Pre-Processing (Background Removal)

4.5 Feature Extraction Using A Deep Learning Model And Classification

For several applications, including segmentation, localization, detection, and classification, convolutional neural networks (CNNs) have become the industry standard [30]. Applications for these designs are widely used in image analysis and other fields where shift-invariance or covariance is necessary. The capacity of CNNs to manage translational invariance, that is, the recognition that an object in an image remains the same even when shifted, is the fundamental idea behind these algorithms. To capture this translational invariance, CNNs use layer-wise affine transformations with convolutional kernels. The following definition of the operation is applied to a 2D convolutional kernel (w) applied to 2D image data (x)

$$S(I, j) = (x * w)(I, j) = \sum_{M}\sum_{N} x(m, n)w(i - m, j - n) \quad (5)$$

Where S (I, j) represents the output at position (I, j), x(m; n) is the value of the input x at position (m, n), w(I– m; j – n) is the parameter of kernel w at position (I– m; j – n), and the summation is performed over all possible positions.

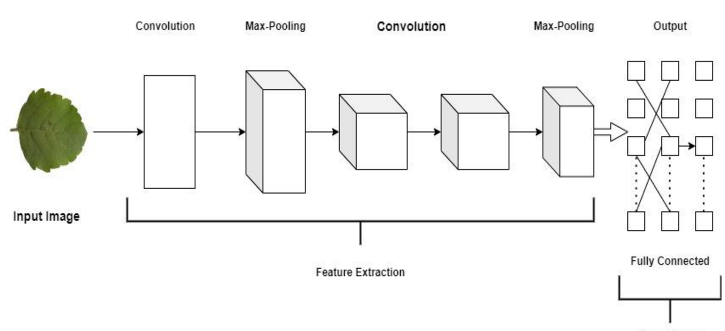

Fig. 9 CNN ARCHITECTURE

1. Convolution Layer

Convolution operations were performed by this layer to extract features from the images. Here, a convolutional kernel produces a convolution plus a bias by sliding along the image with a specific gait. This layer may receive an RGB image as its input or output from another layer. Convolutional kernels share weights to minimize model parameters using the weighted average of pixels in a small area of the input image to create each matching pixel in the output image. The weight of each pixel is determined by a function, and the weights of this layer are all learnable. This procedure can be quantitatively started as

$$X_j^l = f(\sum_{i\in M_j} Y_j^{l-1} K_{ij}^l + b_j^l) \quad (6)$$

Where $Y_j^{l-1}I$is the output of the *ith* feature map in the *l*-1 layer, X_j^l is the input of the *jth* feature map in the *l* layer *Kij* and b_j^l are convolutional kernel and bias in the *l* layer.

Table 3: Configuration of model used.

Name	Parameter
Solver Type	Adam
LR GridSearchCV	3
Batch Size	32
Input Size	224 x 224
Feature extraction buffer size	1000
Train/Test Split	0.80/0.20

2. Pooling Layer

The pooling layers were used to downsample the spatial dimensions of the input volume. For instance, max pooling maintains the maximum value within a specific range, which helps minimize computation and strengthens the network's resistance to changes in scale and orientation. This process is known as downsampling or a pooling operation. This process can be, described as

$$X_j^l = f(\beta_j^l \ .down_s(X_j^{l-1}) + d_j^l) \quad (7)$$

Where X_j^l represents the *jth* feature map in the *l* layer. $\beta_j^l\, j$ and d_j^l are the multiplicative factor and bias respectively. $.down_s$ represents an under-sampling function. Under-sampling can be done in many forms, some of which are average pooling, maximal pooling (max pool), minimal pooling operation, and so on. In our work, we employed max-pooling.

3. Fully Connected Layer

The last fully connected layers of the network map the features learned by the earlier layers onto the output classes Every neuron in the fully connected layer is connected to every other neuron in the feature map of the preceding layer.. The output can be stated as follows.

$$h_{W,b}(x) = f(W^T + b) \quad (8)$$

where $h_{W,b}(x)$ is the output, and W represents the corresponding weights. The inputs to the fully connected layer are many features extracted from the previous layer. Each feature has its unique and important for processing,

4. Loss Function

The difference between the desired and expected outputs was measured using a loss function. The network calculates the loss using the categorical cross-entropy. The categorical cross-entropy function is expressed as follows:

$$\varphi = -\frac{1}{K}\sum_K [y_n \ \log \hat{y}\, n] \quad (9)$$

Where y and $\hat{y}$ represent both the actual and the expected outputs respectively.

5. Activation Layer

It has been established that non-linearity is a crucial component of CNNs and increases their power. An activation function is necessary because CNNs must be able to accept any input between 1 and C1 and simultaneously map it to an output that can range from {0,1} to {1,1} in some situations. Because the activation functions in Deep Learning architectures are designed to produce a nonlinear decision boundary through nonlinear combinations of the weight and inputs, they must exhibit nonlinearity.In CNN training, a nonlinear activation layer is often applied immediately after each convolution layer. In the field of deep learning, several nonlinear functions can be employed to induce nonlinearity. Among them are:

a. TANH

This non-linearity takes the real-valued number to the range [-1, 1], which is mathematically represented as

$$\tanh(x) = 2 \propto (2x) - 1 \qquad (10)$$

b. RELU

This increases the non-linear properties of the model without changing the receptive field of the convolution layer by changing all the negative values to 0. This can be represented as,

$$f(x) = \max(0, x) \qquad (11)$$

c. SIGMOID

The real-valued number is transformed by this non-linearity into the interval [0, 1], or in mathematical terms, it can be expressed as,

$$\sigma(x) = \frac{1}{1 + e^{-x}} \qquad (12)$$

d. SOFTMAX

A binary classifier or multiclass classifier can be used for the classification. The sigmoid activation function is utilized for binary classification, whereas the Softmax function is frequently employed for multiclass classification. The last dense layer uses Softmax activation to determine the likelihood of the predicted classes. The output was determined by selecting the class with the highest probability.

$$S(yi) = \frac{e^{Yi}}{\sum_{J=1}^{K} e^{yj}} \qquad (13)$$

Where e^{Yi} and e^{Yj} denote the probability belonging to the I and j categories respectively; K denotes the number of categories and it is initialized to sixteen in this work.

A. Model Used

1. Xception

It includes the pre-trained Xception model, which is well known for its outstanding results in image recognition assignments. By using careful augmentation and fine-tuning methods, we were able to attain an exceptional accuracy rate of 98%. By utilizing the deep convolutional layers of the Xception architecture, we were able to extract detailed features from photographs of medicinal plants, which allowed for accurate classification.

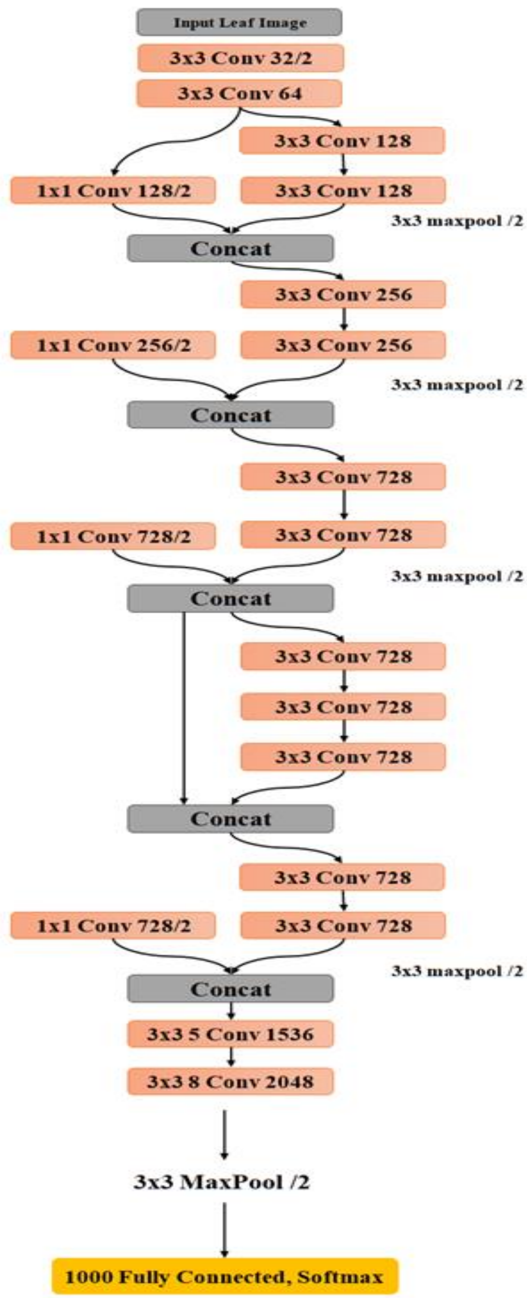

Fig. 12 Xception Architecture

2. ResNet-50

A crucial part was played by the ResNet-50 pre-trained model, which achieved 96% accuracy. By utilizing its deep convolutional layers, ResNet-50 could precisely classify medicinal plant photos by capturing detailed information. Owing to its hierarchical nature, the model was able to extract robust features and recognize tiny visual cues that are essential for precise plant identification. ResNet-50 leveraged its prior knowledge through transfer learning, adjusting its parameters to perform tasks related to the identification of medicinal plants. ResNet-50, a 50-layer deep neural network, showed remarkable proficiency in acquiring intricate patterns, there by augmenting the model's capacity to discriminate between various plant species with pinpoint accuracy.

This remarkable precision highlights the effectiveness of combining domain-specific datasets with sophisticated deep learning architectures, such as ResNet-50, for useful applications in medicinal plant identification.

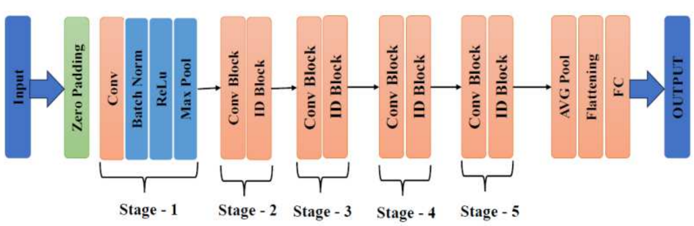

Fig. 11 ResNet-50 Architecture

3. MobileNetV2

With the help of the pre-trained MobileNetV2 model, an astounding 96 percent accuracy was attained. We can discriminate medicinal plants with high precision by utilizing the powerful feature extraction capabilities of Inception-V3 in conjunction with conditional Generative Adversarial Networks (cGANs). The combination of deep learning techniques allows for advanced pattern recognition and the precise classification of various plant types. MobileNetV2 pre-trained weights improve the model generalization and speed up the training process.

Our model exhibited exceptional robustness to changes in plant appearance and climatic circumstances through careful fine-tuning and data augmentation methods. Our system can identify finer details of plants thanks to the cooperation of CNNs and cGANs, which is important for conducting research on therapeutic applications and protecting biodiversity. This ground-breaking research highlights how deep learning can revolutionize the field of plant science.

4. 5. Performance Evaluation

Through a variety of measures, the effectiveness of the MobileNetV2, Xception, and ResNet-50 medicinal plant identification models was assessed. The accuracy of the results was satisfactory, with an average proportion of correctly identified plant species of more than 98 %. At or above 92%, the model's performance in terms of precision, recall, and F1 score, which measures its capacity to reduce false positives and negatives, was excellent. Strong diagnostic capabilities across a variety of plant species were

revealed by the confusion matrix, which offered comprehensive insights into categorization outcomes. Interestingly, the ensemble method that made use of average ensembling further improved the predicted accuracy by combining the advantages of each model. This ensemble approach improves the overall performance while reducing the tendency toward overfitting. Cross-validation methods, including k-fold validation, confirmed the resilience of the ensemble model and guaranteed a constant performance across various data divisions. Taken together, these assessments demonstrated the effectiveness of CNN-based models in the identification of medicinal plants, underscoring their potential for practical use in botany and drug development.

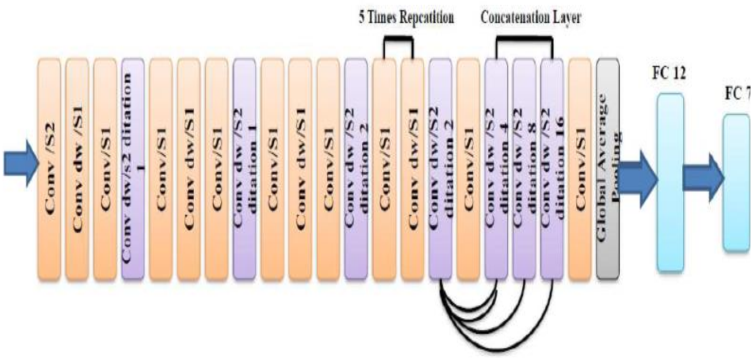

Fig. 10 MobileNetV2 Architecture

Table 4: Tabulated precision, recall and Accuracy for test set on Medi-Plant dataset

Class	Precision	Recall	Accuracy
Arive-Dantu	0.904762	1.000000	0.930693
Avaram	1.000000	1.000000	0.930693
Basale	0.937500	0.937500	0.930693
Bellyache bush	1.000000	0.937500	0.930693
Betel	1.000000	1.000000	0.930693
Big Caltrops	0.800000	0.857143	0.930693
Black Night Shade	0.850000	0.772727	0.930693
Butterfly Pea	0.909091	0.909091	0.930693
Coat buttons	1.000000	1.000000	0.930693
Crown flower	1.000000	0.888889	0.930693
Curry	1.000000	1.000000	0.930693
Drumstick	1.000000	1.000000	0.930693
Fenugreek	1.000000	0.750000	0.930693
Giant Pigweed	1.000000	0.800000	0.930693
Green Chireta	1.000000	0.928571	0.930693
Guava	1.000000	1.000000	0.930693
Hibiscus Rosa-sinensis	1.000000	0.625000	0.930693
Indian Beech	0.909091	0.909091	0.930693
Indian Copperleaf	0.843750	1.000000	0.930693
Indian Jujube	0.545455	0.750000	0.930693
Indian Mustard	1.000000	1.000000	0.930693
Indian Thornapple	0.909091	1.000000	0.930693
Indian pennywort	1.000000	0.700000	0.930693
Jackfruit	1.000000	0.916667	0.930693
Jamun	1.000000	1.000000	0.930693
Jasmine	0.928571	0.928571	0.930693
Karanda	1.000000	1.000000	0.930693
Lemon	0.750000	1.000000	0.930693
Madagascar Periwinkle	1.000000	1.000000	0.930693
Madras Pea Pumpkin	1.000000	1.000000	0.930693
Malabar Catmint	1.000000	1.000000	0.930693
Mango	0.900000	1.000000	0.930693
Mint	1.000000	1.000000	0.930693
MuntingiaCalabura	1.000000	1.000000	0.930693
Neem	0.923077	1.000000	0.930693
Oleander	1.000000	1.000000	0.930693
PanicledFoldwing	0.750000	0.923077	0.930693
Parijata	1.000000	0.700000	0.930693
Peepal Tree	1.000000	1.000000	0.930693
PlectranthusAmboinicus	0.857143	1.000000	0.930693
Pomegranate	1.000000	0.904762	0.930693
Rasna	1.000000	1.000000	0.930693
Rosary Pea	1.000000	1.000000	0.930693
Rose Apple	1.000000	1.000000	0.930693
Roxburgh fig	1.000000	0.857143	0.930693
Sandalwood	0.700000	1.000000	0.930693
Square Stalked Vine	0.857143	1.000000	0.930693
TabernaemontanaDivaricata	0.888889	0.571429	0.930693
Trellis Vine	1.000000	1.000000	0.930693
Tulsi	1.000000	1.000000	0.930693
WEIGHTED AVG	**0.943271**	**0.931343**	**0.930693**
TOTAL SCORE	**0.935102**		

Here, the performance of each model is evaluated, and the evaluation of the feature extraction techniques is also done based on the accuracy score of the model. Each model is evaluated using evaluation metrics and it is defined as follows:

i. **Accuracy** - The ratio of accurately predicted observations to all observations.

ii. **Recall** - The ratio of positively observed observations that were accurately predicted to all observations in the actual class that was in the yes/positive/1 category.

iii. **Precision** - Measure of how often positive observations are successfully anticipated compared to all positive observations that were forecasted.

Accuracy = TP+TN/TP+TN+FP+FN

Precision = TP/TP+FP

Recall = TP/TP+FN

Where TP – True Positive, TN - True Negative, FP - False Positive, FN - False Negative.

5. Results

The creation of a medi-plant dataset, a noteworthy advancement in the field of medicinal plant identification, is presented in this work. The dataset addresses the lack of easily accessible resources for medicinal herbs and includes approximately 3300 leaf photos from different types of Indian plants. Synthetic data were created using a Conditional Generative Adversarial Network. Three pre-trained deep convolutional neural network architectures, MobileNetV2, ResNet-50, and Xception, were used to assess the efficacy of the Medi-plant dataset. After training on all three designs, the suggested Medi-Plant model performed exceptionally well, achieving an astounding accuracy of 98.6%. With a one-second prediction time per image, this tool is remarkably effective in helping users identify plants from photos. Furthermore, the application provides users with pertinent details regarding the herbs detected from the database, and the findings of this work have important ramifications for interested parties and society. The goal of this research was to improve the accuracy and comprehensiveness of medicinal plant identification systems by growing the dataset further. In addition to helping those involved in herbal medicine, this will educate the public about the medicinal properties of many herbs and encourage better lifestyles.

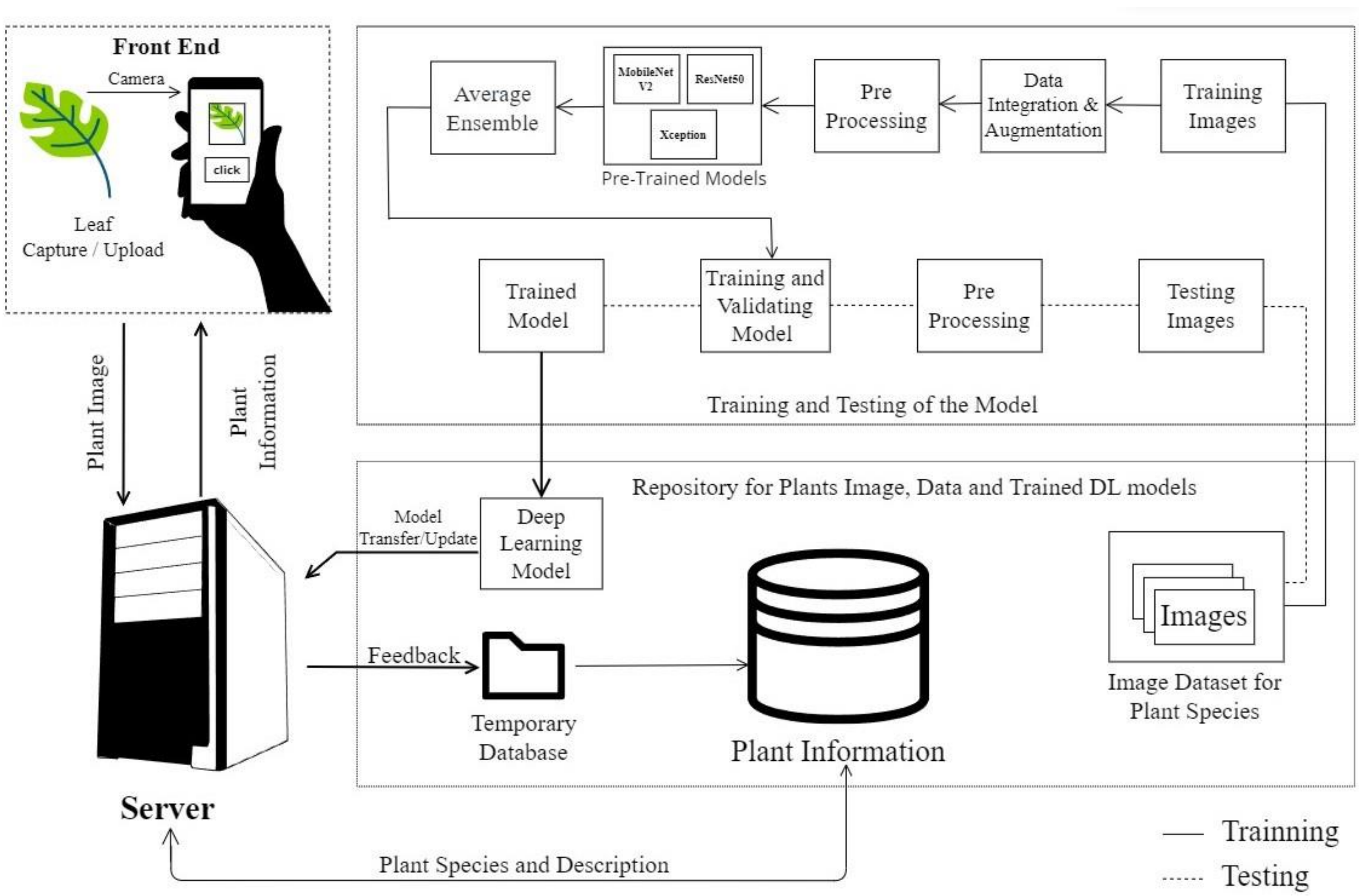

Fig. 13 System design that depicts the flow of the model

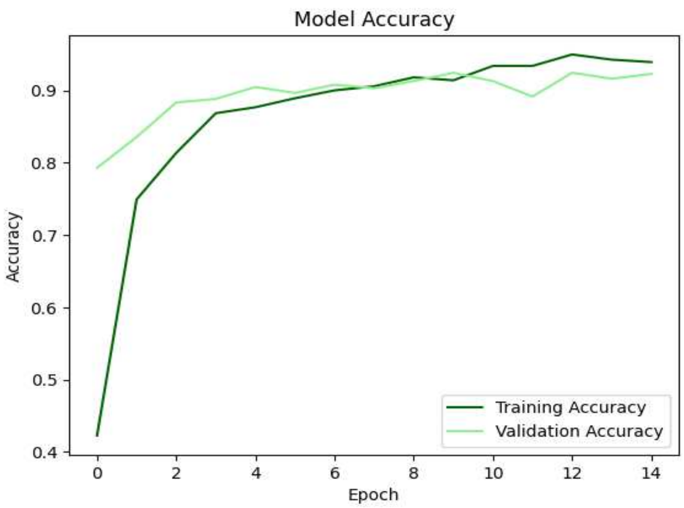

Fig. 14 Model Accuracy for MobileNetV2

The CNN architectures were compared for accuracy, and MobileNetV2 performed admirably, with an impressive accuracy of 93.65 %, indicating its effectiveness in mobile and embedded applications. With a remarkable 97.6% accuracy rate, inception is the top option for challenging picture recognition jobs owing to its strong feature extraction skills.

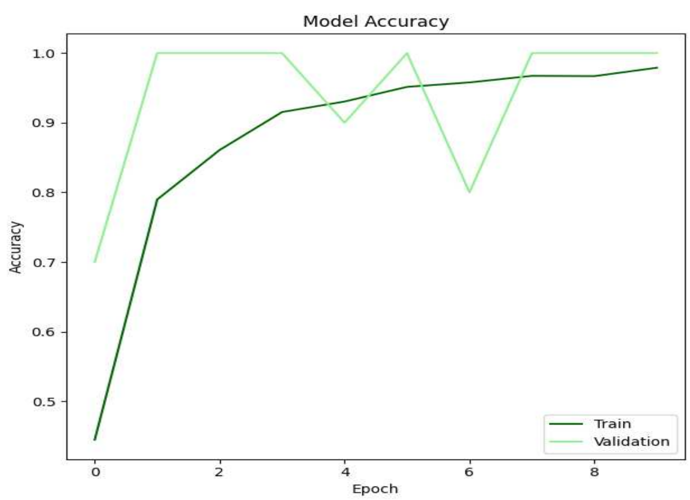

Fig. 15 Model Accuracy for Xception

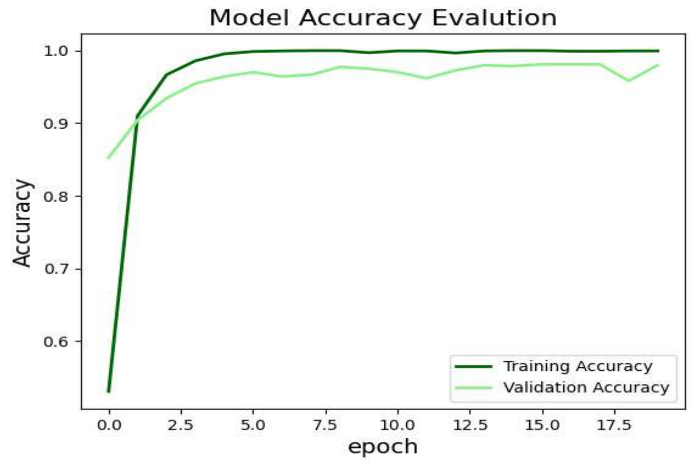

Fig. 16 Model Accuracy for ResNet50

With an accuracy of 95.6%, ResNet50 is a popular choice for research and real-world applications because it offers a good mix of model complexity and performance

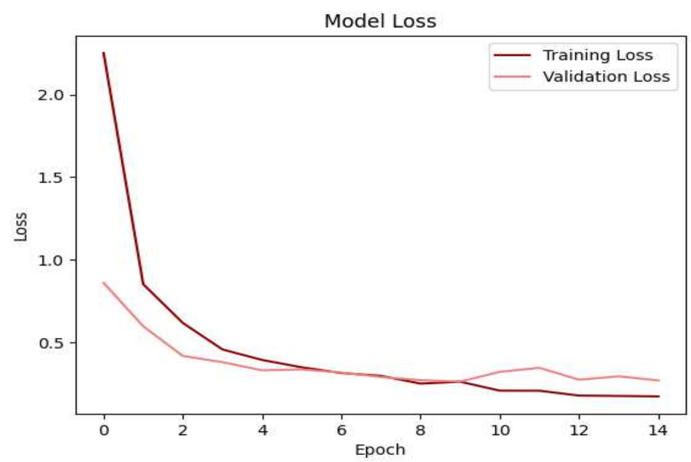

Fig. 17 Model Loss for MobileNetV2

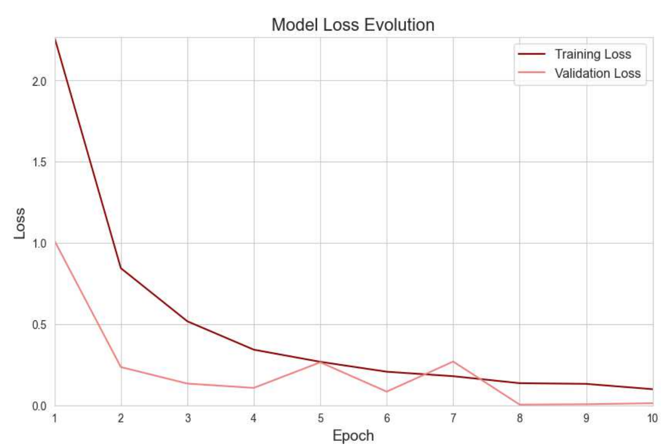

Fig. 18 Model Loss for Xception

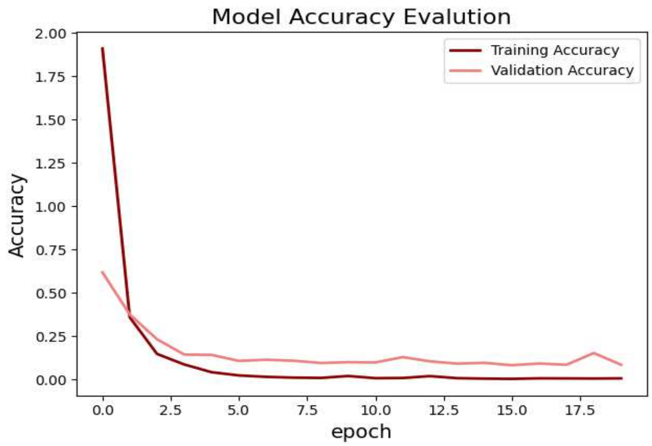

Fig. 19 Model Loss for ResNet-50

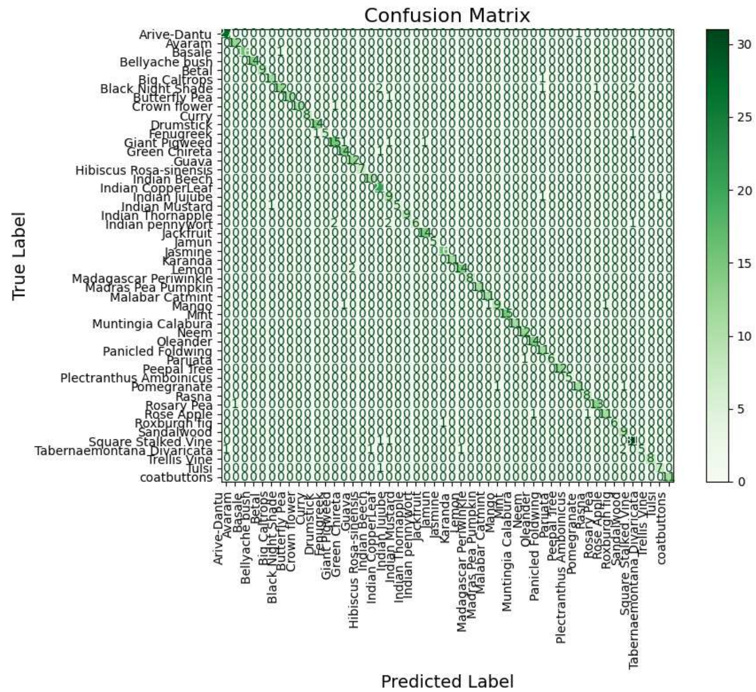

Fig. 20 Confusion Matrix for Average Ensemble Model

A confusion matrix that shows the counts of true positive, true negative, false positive, and false negative predictions is a condensed depiction of a classifier's performance. t offers information on how well a classifier can accurately categorize instances in various classes. The instances in a predicted class are represented by each column of the matrix, whereas the occurrences in an actual class are represented by rows.

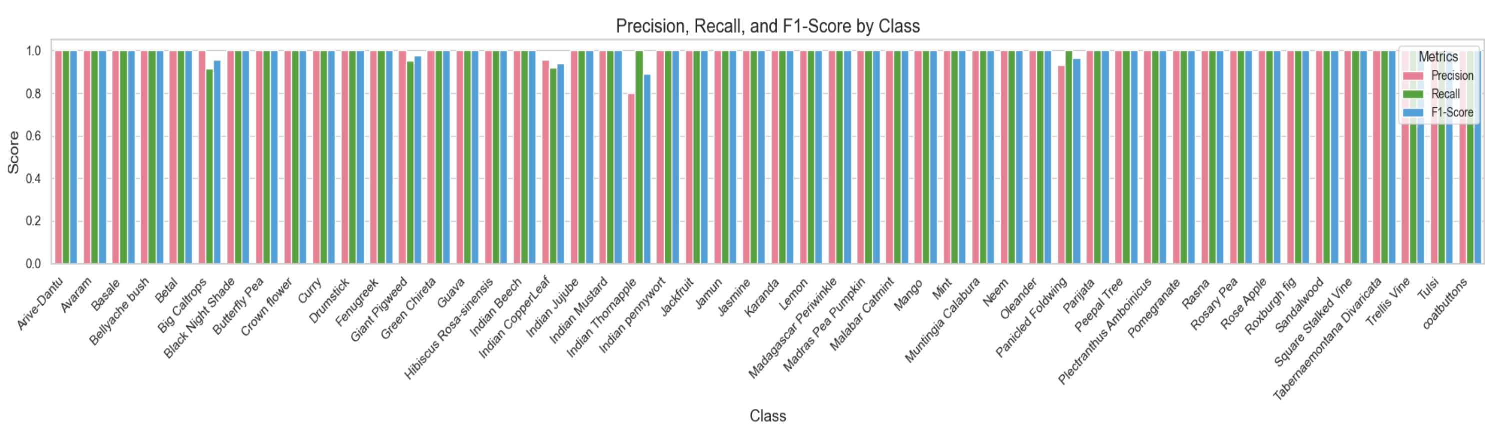

Fig. 21 Precision, Recall and F1-Score of each class for Average Ensemble Model

6. Discussion

A mobile application called 'Medi-Plant' incorporates an ensemble model compatible with both iOS and Android. To detect medicinal herbs, the suggested model (M4) uses artificial neural networks (ANN) for classification and the Xception, MobileNetV2, and ResNet-50 architectures for feature extraction from input photos. The Medi-plant dataset contains 50 distinct herb species that are accessible through the app. Users of the "Med-Plant" function have the option of uploading herb images taken with our device's camera for prediction, or they can take pictures of herbs (ideally against a lighter backdrop). The app's images provide users with a wealth of information on the recognized herb by showing the user-inputted herb image, prediction results, and relevant details taken from the database. By making information about therapeutic herbs easily accessible, it enables people to learn more about them, which may result in a greater use of herbs in agriculture and support the preservation of biodiversity. By making information about therapeutic herbs easily accessible, it enables people to learn more about them, which may result in a greater use of herbs in agriculture and support the preservation of biodiversity.

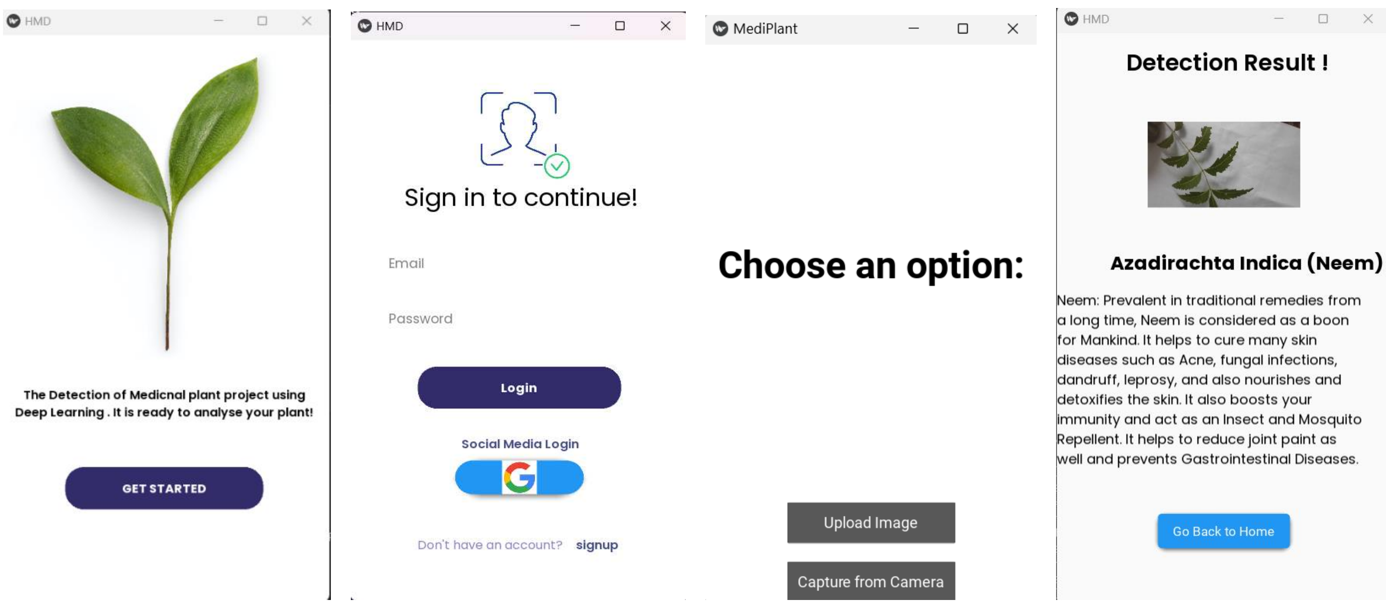

Fig. 22 Screenshot of Medi-Plant Application

6. Conclusions

In summary, this work emphasizes the importance of biodiversity conservation, especially for protecting endangered species of medicinal plants. A major advancement in the ease of identification and application of medicinal plants has been accomplished with the creation of an automatic identification system that uses neural network techniques in computer vision and deep learning. Although there are not many easily accessible datasets for medicinal plants, Conditional Generative Adversarial Networks have made it possible to create synthetic data, which has allowed the establishment of a medium-plant dataset. Testing against pretrained deep convolutional neural network architectures, such as MobileNetV2, ResNet-50, and Xception, has proven the efficacy of the medium plant dataset. With an astounding accuracy of 98.6%, the suggested medium-plant model, which combines the best features of the three models, demonstrated greater performance. As a result, the Medi-Plant Identification application was created, allowing for the quick and precise identification of therapeutic herbs with a prediction time of one second per picture.

7. List of Abbreviations

CNN - Convolutional Neural Network

DCNN - Deep Convolutional Neural Network
DL - Deep Learning
GAN - Generative Adversarial Network
cGAN - Conditional Generative Adversarial Network
Pix2Pix - Image-to-Image Translation

8. Declarations

- *Ethics approval and consent to participate*

  Not applicable
- *Consent for publication*

  Not applicable
- *Availability of data and material*
  1. Ahila Priyadharshini R., Arivazhagan S., Arun M. (2021) Ayurvedic Medicinal Plants Identification: A Comparative Study on Feature Extraction Methods. In: Singh S.K., Roy P., Raman B., Nagabhushan P. (eds) Computer Vision and Image Processing. CVIP 2020. Communications in Computer and Information Science, vol 1377. Springer, Singapore. https://doi.org/10.1007/978-981-16-1092-9_23
  2. Ahila Priyadharshini, R., Arivazhagan, S., Arun, M. (2023). A Curated Dataset for Spinach Species Identification. In: Gupta, D., Bhurchandi, K., Murala, S., Raman, B., Kumar, S. (eds) Computer Vision and Image Processing. CVIP 2022. Communications in Computer and Information Science, vol 1777. Springer, Cham. https://doi.org/10.1007/978-3-031-31417-9_17
  3. Ritesh Ranjan. (2021) segmented medicinal leaf images.https://www.kaggle.com/datasets/riteshranjansaroj/segmented-medicinal-leaf-images
- *Competing interests*

  The author has no conflicts of interest to declare that are relevant to the contents of the articles.
- *Funding*

  No funding body had any role in the design of the study, collection, analysis, and interpretation of data, or in writing the manuscript.
- *Authors' contributions*

  We carried out the analysis and interpretation of the plant leaf dataset to identify a specific leaf image and examined the finalized model. We enriched the data using pix2pix GAN to create synthesis images. We created the mobile application and construct the classification models with the description of plants, including information on their therapeutic uses.
- *Acknowledgements*

  The authors would like to thank the Management and Principal of Mepco Schlenk Engineering College (Autonomous), Sivakasi for providing us the state of art facilities to carry out this proposed research work in the Mepco Research Centre in collaboration with Anna University Chennai, Tamil Nadu, India.
- *Authors' information*

Mrs. P. Swathika, Assistant Professor(Senior Grade)
Department of Artificial Intelligence and Data Science,
Mepco Schlenk Engineering College (Autonomous), Sivakasi, 626005, Tamil Nadu, India
swathikap@mepcoeng.ac.in;

Ajithar A, Hishaamudin Z, Nitheeswaran E
Student
Department of Artificial Intelligence and Data Science,
Mepco Schlenk Engineering College (Autonomous), Sivakasi, 626005, Tamil Nadu, India
ajitharmahes@gmail.com;
hishaaofficial1107@gmail.com;
nitheeswaran56@gmail.com;

Appendix

The Medi-Plant collected and respective details such as leaf image, common name, scientific name, number of samples, and medicinal properties.

Leaf Image	*Common Name*	*Botanical Name*	*Samples*	*Medicinal Properties*
	Arive-Dantu	Amaranthus Viridis	122	Curing asthma and bronchitis.
	Avaram	Senna auriculata	66	Fevers, diabetes, diseases of the urinary system, and constipation.
	Basale	Basella Alba	103	Anemia, Cancer in the stomach, and osteoporosis.
	Bellyache bush	Jatropha gossypifolial	81	Antihypertensive, anti-inflammatory, antiophidian.
	Betel	Piper betel	48	Antioxidant, anti-inflammatory.
	Big Caltrops	Kallstroemia maxima	63	Anti-cancer and antioxidant capacities
	Butterfly Pea	Clitoriaternatea	108	Memory enhancer, nootropic, and antistress.

	Black Night Shade	Solanum nigrum	51	stomach irritation, spasms, pain, and nervousness
	Coat buttons	Tridax procumbens	58	Treat bronchial catarrh, diarrhea, dysentery, and liver diseases
	Crown Flower	Calotropis gigantea	66	Skin, digestive, respiratory, circulatory, and neurological disorders
	Curry	Murraya Koenigii	60	Piles, Inflammation, Itching, Fresh cuts, Dysentery, Bruises, Edema.
	Drumstick	Moringa Oleifera	77	Tuberculosis, fatty liver.
	Fenugreek	Trigonella Foenum-graecum	36	Antidiabetic, antihyperlipidemic, antiobesity, anticancer, anti-inflammatory.
	Guava	Psidium Guajava	103	High cholesterol, heart disease, and cancer.
	Green Chireta	Andrographis paniculata	65	Common cold, diarrhea, fever due to several infective causes, and jaundice.
	Giant Pigweed	Trianthemaport ulacastrum	65	Traditionally used as an analgesic, stomachic, laxative.
	Hibiscus Rosa	Rosa-Sinensis	43	Antioxidant, anti-inflammatory, anti-microbial, anti-ulceration
	Indian_ Beech	Pongamia pinata	81	Diabetes and its complications
	Indian_ Mustard	Brassica Juncea	124	Digestion, Inflammation, Immune system.
	Indian Jujube	Ziziphus mauritiana	66	Anti-cancer and liver damage treatment
	Basale	Basella Alba	103	Anemia, Cancer in the stomach, and osteoporosis.
	Indian pennywort	Centella Asiatica	34	Blood purifier and high blood pressure
	Indian Thornapple	Datura metal	64	Analgesic, anthelmintic
	Indian Copperleaf	Acalypha indica	57	anthelmintic, anti-inflammation
	Jackfruit	Artocarpus Heterophyllus	56	Vitamin C, potassium, dietary fiber, and other vitamins and minerals.
	Jamun	Syzygium Cumin	39	Diabetes, hyperlipidemia, hypertension, obesity.
	Jasmine	Jasminum	71	Hepatitis, cirrhosis, Dysentery
	Karanda	Carissa Carandas	74	Urinary disorders and diabetic ulcers.
	Lemon	Citrus Limon	57	Digestive disorders and aid digestion.
	Mango	Mangifera Indica	62	Vitamins and minerals, including vitamin C, vitamin A, potassium, and fiber.
	Muntingia Calabura	Muntingia calabura	56	Headaches, prostate problems, reduce gastric ulcers.
	Madras Pea Pumpkin	Cucumis maderaspatanus L.	53	Treatment of cuts, wounds, inflammation, scabies, fever, chest infections, sinusitis.
	Madagascar Periwinkle	Catharanthus roseus	55	Antidiabetic, bactericide and antihypertensive
	Malabar Catmint	Anisomelesmal abarica	65	Anti-allergic, anti-anaphylactic anti-inflammatory.
	Mint	Mentha	97	Antioxidants, menthol, and phytonutrients that help enzymes digest food.
	Neem	Azadirachta Indica	60	Anti-inflammatory, antihyperglycaemic
	Oleander	Nerium Oleander	62	Malaria, ringworm, indigestion.
	Parijata	Nyctanthes Arbor-tristis	40	Intermittent fevers, arthritis, and obstinate sciatica.
	Peepal Tree	Ficus Religiosa	63	Ear pain, toothache, haematuria.
	Panicled Foldwing	Dicliptera paniculata.	68	Antidote for snake poison

	Plectranthus Amboinicus	Coleus amboinicus	48	Antimicrobial, Anti-inflammatory, Anti-tumour.
	Pomegranate	Punica Granatum	79	It improves wound healing and is beneficial to the reproductive system.
	Rasna	Alpinia Galanga	50	Reducing joint pain in arthritis, better kidney health.
	Rose apple	Syzygium Jambos	56	Antioxidant content, contains diuretic.
	Roxburgh	Ficus auriculata	50	Cuts, wounds, diarrhea, and dysentery.
	Rosary Pea	Abrus precatorius	66	Asthma, cough, tuberculosis, bronchitis, and chest pain.
	Sandalwood	Santalum Album	58	Headache, stomach ache, and urinary.
	Square Stalked Vine	Cissus quadrangularis	158	Obesity, cancer, high cholesterol.
	Tabernaemon tanaDivaricat a	Tabernaemonta nadivaricata	56	Anti-inflammatory, anti-microbial, anti-tumor, anti-oxidant,
	Trellis Vine	Pergulariadaemi a	57	Headache, stomach ache, treat skin irritations and burns.
	Tulsi	Ocimum Tenuiflorum	52	Antibacterial, antifungal, and antiviral.

References

[1] K. J. Willis, “State of the world’s plants 2017. Report,” Roy. Botanic Gardens, U.K., Tech. Rep., 2017.

[2] B. Saad, H. Azaizeh, G. Abu-Hijleh, and O. Said, “Safety of traditional Arab herbal medicine,” Evidence Complementary Alternative Med., vol. 3, no. 4, pp. 433439, 2006.

[3] Ambarwari, A., Adrian, Q. J., & Herdiyeni, Y. (2020). "Analysis of the effect of data scaling on the performance of the machine learning algorithm for plant identification.” Journal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 4(1), 117-122.

[4] Azeez, Y. R., & Rajapakse, C. (2019, March). "An application of transfer learning techniques in identifying herbal plants in sri lanka.” In 2019 International Research Conference on Smart Computing and Systems Engineering (SCSE) (pp. 172-178). IEEE.

[5] Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., ... & Hassabis, D. (2020). "Improved protein structure prediction using potentials from deep learning." Nature, 577(7792), 706-710.

[6] Vidya Sree, K., Sandeep Kumar Reddy, G., Dileep Varma, R., Mihira, P., & Remya, S. (2023). "A One-Stop Service Provider for Farmers Using Machine Learning." In Sentiment Analysis and Deep Learning: Proceedings of ICSADL 2022 (pp. 805-823). Singapore: Springer Nature Singapore.

[7] Liu, P., Cottrill, A. L., Kozawa, D., Koman, V. B., Parviz, D., Liu, A. T., ... & Strano, M. S. (2018). "Emerging trends in 2D nanotechnology that are redefining our understanding of “Nanocomposites”." Nano Today, 21, 18-40.

[8] Patel, I., & Patel, S. (2019). "Flower identification and classification using computer vision and machine learning techniques." International Journal of Engineering and Advanced Technology (IJEAT), 8(6), 277-285.

[9] Habiba, S. U., Islam, M. K., & Ahsan, S. M. M. (2019, July). "Bangladeshi plant recognition using deep learning-based leaf classification." In 2019 International conference on computer, communication, chemical, materials and electronic Engineering (IC4ME2) (pp. 1-4). IEEE.

[10] Panicker, R. O., Soman, B., & Sabu, M. K. (2019). Tuberculosis detection from conventional sputum smear microscopic images using machine learning techniques. Hybrid Computational Intelligence, 63-80.

[11] Wei, J., Li, Z., Xue, W., Sun, L., Fan, T., Liu, L., ... & Cribb, M. (2021). "The ChinaHighPM10 dataset: generation, validation, and spatiotemporal variations from 2015 to 2019 across China." Environment International, 146, 106290.

[12] Tomar, D., & Agarwal, S. (2017). Multiple instance learning based on twin support vector machine. In Advances in Computer and Computational Sciences: Proceedings of ICCCCS 2016, Volume 1 (pp. 497-507). Springer Singapore.

[13] A. Kadir, L. E. Nugroho, A. Susanto, and P. I. Santosa, ‘‘Leaf classification using shape, color, and texture features,’’ Int. J. Compute. Trends Technol., vol. abs/1401.4447, pp. 226–229, Nov. 2013.

[14] J. S. Cope, D. Corney, J. Y. Clark, P. Remagnino, and P. Wilkin, ‘‘Plant species identification using digital

morphometrics: A review," Expert Syst. Appl., vol. 39, no. 8, pp. 7562–7573, 2012, doi: 10.1016/j.eswa.2012.01.073.

[15] N. Ahmed, U. G. Khan, and S. Asif, "An automatic leaf-based plant identification system," Sci. Int., vol. 28, no. 1, pp. 1–4, Jan. 2016.

[16] T. Jin, X. Hou, P. Li, and F. Zhou, "A novel method of automatic plant species identification using sparse representation of leaf tooth features," PLoS ONE, vol. 10, no. 10, Oct. 2015, Art. no. e0139482, doi: 10.1371/journal.pone.0139482.

[17] J. C. Neto, G. E. Meyer, D. D. Jones, and A. K. Samal, "Plant species identification using elliptic Fourier leaf shape analysis," Comput. Electron. Agricult., vol. 50, ,Feb.2006, doi: 10.1016/j.compag.2005.09.004.

[18] J. Chaki and R. Parekh, "Plant leaf recognition using shape-based features and neural network classifiers," Int. J. Adv. Comput. Sci. Appl., vol. 2, no. 10, pp. 1–12, 2011, doi: 10.14569/ijacsa.2011.021007.

[19] J.-X. Du, X.-F. Wang, and G.-J. Zhang, "Leaf shape based plant species recognition," Appl. Math. Comput., vol. 185, no. 2, pp. 883–893, Feb. 2007, doi: 10.1016/j.amc.2006.07.072.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105, doi: 10.1061/(ASCE)GT.1943-5606.0001284.

[21] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2015, pp. 452–456, doi: 10.1109/ICIP.2015.7350839.

[22] C. Zhang, P. Zhou, C. Li, and L. Liu, "A convolutional neural network for leaves recognition using data augmentation," in Proc. IEEE Int. Conf. Compute. Inf. Technol., Oct. 2015, pp. 2143–2150, doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.318

[23] Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," Commune. ACM, vol. 63, no. 11, pp. 139–144, 2020, doi: 10.1145/3422622.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Compute. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[25] L. Sharon, Automating Data Augmentation: Practice, Theory and New Direction. Stanford, CA, USA: Stanford Univ., 2020. Accessed: May 25, 2021. [Online]. Available: https://ai.stanford.edu/blog/dataaugmentation

[26] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, arXiv:1411.1784

[27] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5967–5976, doi: 10.1109/CVPR.2017.632

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent., 2015, pp. 234–241.

[29] M. B. Lee, Y. H. Kim, and K. R. Park, "Conditional generative adversarial network-based data augmentation for enhancement of iris recognition accuracy," IEEE Access, vol. 7, pp. 122134–122152, 2019, doi: 10.1109/ACCESS.2019.2937809.

[30] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert, "Attention gated networks: Learning to leverage salient regions in medical images," Med. Image Anal., vol. 53, pp. 197–207, Apr. 2019, doi: 10.1016/J.MEDIA.2019.01.012.