A Novel Approach to Identify Medicinal Plant Based on Image Processing

DANDU SAI SATYA VAMSI KRISHNA[1], EDIGA VINOD KUMAR[2], EDIGA MAHENDRA GOUD[3], DR. SHARMASTH VALI Y[4]

[1, 2, 3] Department of Computer Science and Engineering, Presidency University, Bengaluru, India

[4] Assistant Professor, Department of Computer Science and Engineering, Presidency University, Bangalore, India.

Abstract- Never forgetting the beautiful things our ancestors did, like giving birth in the home and treating a variety of complex diseases with the aid of nature. They used to be sufficiently knowledgeable about the leaves and their therapeutic benefits, but this generation is falling behind. Our "Dhanvantri" smartphone app, which gives you the medicinal benefits of the leaf by taking a leaf image as an input, was created to aid the community. The two primary functions of Dhanvantri is to recognise a leaf (the input), classify it according to its label, and then retrieve its medicinal characteristics. By comparing the descriptors of each pixel in the image using various layers of convolutional neural networks, we were able to identify the image. The images were then classified based on their labels. At the conclusion, the tagged image's therapeutic properties are revealed. The trained model is ultimately saved in TensorFlow light format, stored in a database, and invoked using the model reference in Android Studio. To train the model, we used 30 different species of Indian leaves as well as several leaf forms within each species. The outcomes of the Dhanvantri app will aid society in reviving bygone eras.

Indexed Terms- Dhanvantri, therapeutic(medicinal), recognise, classify, descriptors, pixels, convolutional neural network, TensorFlow lite, leaf detection, bygone eras, arive-dantu, parijata, peepal, relu, roxburgh, basale

I. INTRODUCTION

Health is considered to be the top priority for people. Nowadays, if we experience even a mild fever, we immediately rush to the doctor. The doctor will then give us chemical medications, which we will interpret as a complete cure. Chemical medicines are viewed as being less than ideal for human intake, and if used frequently, can even be harmful to health. In order to help the community in identifying medicinal plants more easily, a system is therefore required. In this case, the medicinal plants are concentrated on the introduction of medicinal leaves. We are creating a mobile application for the society so that anyone can use it. Our mobile application will provide you with the leaf's medicinal characteristics after receiving the user's leaf image as input, allowing you to employ the leaf for any health-related problems you may be experiencing. Without compromising it anywhere, we made it happen using a wide range of technologies. We created the application with unschooled in mind as well so that anyone can use it without any prior knowledge.

It is very remarkable to uncover startling facts about the past, such as how complex ailments have been treated at home and made completely curable with the aid of nature. Historically, medicinal plants have been the driving force behind nature and have been responsible for the cure of a variety of diseases. The ancient people had some knowledge of these therapeutic plants at first, but with time and practise they became experts. However, this generation is falling behind in knowledge, making our smartphone app a small contribution to society in trying to restore the good old days.

When it comes to our project, the mobile application that we are creating is mostly divided into two parts: the machine learning component and the android development component. To process the leaf image and train the model using it, various techniques from machine learning and deep learning will be used in the machine learning portion. The Android programming portion will use Android Studio for the UI, layouts, and connection to the backend machine learning model(trained). The difficulty arises when we attempt to integrate the code portions for machine learning and android. Following very quick research and discussion, we learned that there are many ways to link the machine learning code to the Android studio. To select one of them, we used the trial and error method. Additionally, the approach that has little impact on the accuracy of our Machine learning model. We'll talk more about our approach shortly.

II. BACKGROUND LITERATURE

For over 60 years, researchers have been trying to figure out how to make machines understand visual data. Years of study eventually led to the creation of the first computer image scanning technology, which made it possible for computers to digitise and acquire images. when two-dimensional photographs were able to converted into three-dimensional forms by computers in 1963. At the same time, scientists were encouraged by the outcomes of their research in the artificial intelligence field, which is also giving them some hope. Then, using research based on AI and computer vision, scientists began to address issues with human vision-based problems [1] & [2]. Researchers have started focusing on image data for a variety of applications based on that computer vision. From the papers, the 30 different Indonesian species have been studied. Local binary patterns to extract texture and probabilistic neural networks were employed to identify the image. The authors' classification of images of medicinal leaves in the conclusion had a 60% client satisfaction rate [3]. We also heard about a global average pooling-based AI technique for identifying therapeutic plants. They worked on many CNN layers, such as the classification and feature layers. They pre-processed the data by removing the backgrounds from the images before training it. The authors employed GAP (global average pooling), which by limiting the number of parameters, prevents the overfitting issue [4]. Based on machine learning and deep learning, they offer a practical solution to the problem of leaf recognition in this paper. They suggest pre-processing a leaf in order to obtain its finely tuned colour picture, vein image, hand-crafted form, texture features, and Fourier descriptors [5]. In addition to leaf identification, we also come across research that are related to identifying leaf diseases, which are highly interesting and helpful for farmers. In one such study, computer vision was used to understand more about the image, and machine learning was utilised to build a model with greater accuracy. to determine the leaf disease, Extractions include leaf colour, leaf area, leaf damage level, and leaf texture parameters. For classification or detection tasks, random forest classifier has been utilised [6]. We discovered a lot of variation as the study on plant disease detection progressed by comparing it to other research publications. We ought to take greater caution when identifying leaf diseases than when identifying leaves from a medical standpoint. It is important to pay close attention to the area of the leaf that has suffered the most damage so that the machine can learn more about it and make more accurate predictions. One such method, which is pretty interesting offers an accuracy of 98.79% with a standard deviation of 0.57 on 10-fold cross validation [7]. their suggested method automates the identification of plant diseases by performing pre-processing, segmenting the affected leaf area, computing features using the gray-Level Co-occurrence Matrix (GLCM), selecting features, and classifying the features [7]. We got to the conclusion, after conducting extensive study, that the majority of researchers have employed machine learning methods such as support vector machines, k-nearest neighbour, random forest, and other classifiers. Additionally, deep learning algorithms were used to train the neural network for improved performance. When compared to other projects, ours has a significant advantage because we are turning our deep learning model into a mobile application.

III. METHODOLOGY

As you are all aware, our application consists of two components: deep learning and app development. I'll start with the deep learning method, which is the building block of our application. However, I'll just give you a quick look at the architecture and data flow diagram first.

4.1 Proposed System Design

The basic design of the system we are designing is shown in the figure below (fig 1). The design is divided into three components: the front-end view, the back-end deep learning and database, and once again, the client-facing user interface. This starts with a mobile device scan, followed by database checks to see if the image client's provided image is there or not, a backend deep learning model, and the output to the user interface. After doing extensive research on various systems that have already been created, we proposed this one. The difficult part of this is that we are turning our deep learning model into a mobile app. We should be more specific about that because it is a novel challenges in comparison to other limitations. Given that it is an initial architecture, there will be gaps in our understanding of the project, but those discrepancies will close as time goes on.

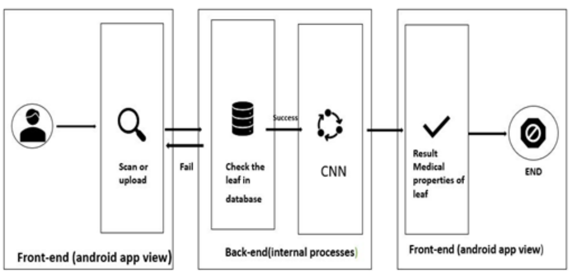

Fig .1 Proposed System design training model

4.2 Data flow Diagram

The project's data flow diagram is shown in the figure (fig 2). How the data is processed throughout the project, from beginning to end. The client first takes the image using the camera or uploads it through the gallery, after which the image is scanned and passed to the Android studio by the user interface. There, the image is transformed into a bitmap. The tensorflow lite model (the deep learning model is recorded in this format) receives the bitmap code from there. Where deep learning is involved, the model recognises the name of the image and assigns its medicinal properties based on training data and accuracy. Through the user interface, the client will see the recognised details. If the client-uploaded image is not on the trained list because the model was only trained on a small amount of images, it will inform the client that the image they are looking for is not available. Check out the image below to get more information on the description that is provided (fig 2).

When we looked for ways to link the Android studio and deep learning model. First, we considered turning the entire deep learning Python code into an API using Postman and calling this API in one of the Python frameworks called Flask. Although the flask is connected to the database and the android studio there, felt these processes were cumbersome and began looking for alternatives. The model is then converted into a lite format in the Tensorflow library so that it can be utilised or called elsewhere.

Bitmap' is a set of bits in a rectangular array of pixels that identifies the colour of each pixel. Having access to a lot of colour information in bitmap files is one benefit of employing bitmap graphics in your designs. TensorFlow Lite employs TensorFlow models that have been downsized and optimised for machine learning (ML) models. so that we can employ machine learning in all of the applications we make

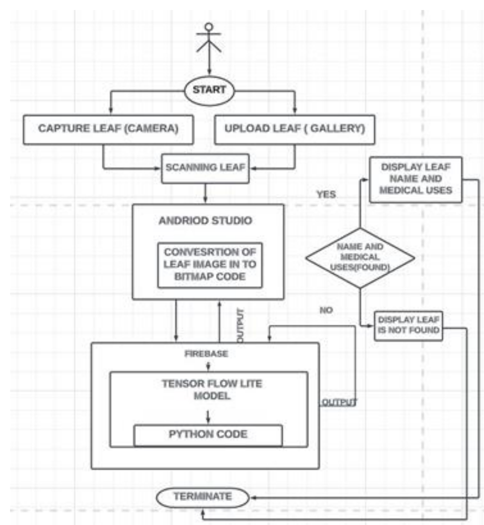

Fig 2. Proposed System design training model

4.3 Data Preparation

When working on projects based on image processing or computer vision, processing the image data is an essential step. From all sources, we have gathered 30 different Indian species. There are around 100 distinct varieties of each species, which implies that there will be a variety of leaf types within each class. We achieve better outcomes as the class's picture count rises. The data has been divided into train and test sets. The model is developed using the train data, and it is then evaluated using the test data.

The species that we collected are: arive-dantu, basale, betel, crepe-jasmine, curry, drumstick, fenugreek, guava, jackfruit, jamun, jasmine, karanda, lemon, mango, Mexican mint, neem, oleander, parijata, peepal, pomegranate, rose apple, roxburgh, sandalwood, and tulsi. There is no need to worry about concerns with clarity, background, dimensions, or colours because the photographs we obtained have already been pre-processed and in a good way. One of the images from our dataset is shown here.

The types of leaves that we introduced to each class are illustrated in the examples below. Figures 3.1 and 3.2 are examples of the arive-dantu leaf kind of images. The data has been loaded and divided into train and test groups. examining the data to gain a better idea of the next steps. basically resizing the images in the dataset by scaling down the data. will show the randomly selected images from each class (fig 3.3)

Fig 3.1 arive-dantu leaf (type 1)

Fig 3.2 (type 2)

Fig 3.3 Random_images

4.4 Proposed Methodology

We will build a deep learning neural network to train the data on because the application and the data we have are both image-based. A neural network with several layers is called a deep learning model. The input layer, which is the top layer, is where input data is received. The output layer can employ the representation of the input data created by the hidden layer, which is the second layer. The outcome predicts the expected outcome. We trained the network with data using the convolutional neural network algorithm. We built a network with a total of four levels: an input layer, two hidden layers, and one output layer.

The leaf picture is covered by the parameter known as the filter (32) in the network's first layer, which is a convolutional 2d layer. The stride parameter is used by the filter to move around the image. The stride was given as (3,3). We utilised relu as the activation function in this layer. The input layer receives the descriptors for the leaf image, including its colour, size, length, and form, based on all of these factors and the method underlying them. The layer is now prepared with the descriptor information from the training set of images. The concealed layer will receive these details.

In our neural network, we have two hidden layers where the non-linear processing of the images is carried out. These layers are where the descriptors of the leaf image were taken. Convolutional neural network's second layer, where maximum pooling will be used. Max pooling is a process that chooses the most from the area of the image that the filter has filtered. The stride is (2,2) here. The image's dimensions have been reduced or normalised using pooling to create the feature map. The output from the hidden layer is again transferred into the input layer by updating or correcting the weights with another technique known as "backpropagation" at this layer. Here the backpropagation used to adjust the weights of the output layer and to send it again to the input layer. this happens for some iterations. In this times iterations the model gets trained with leaf images.

The final output is sent to the output layer following an iterative process in the hidden levels. Softmax is the activation function utilised in this convolutional dense layer. However, in order to produce the required number of classes, we must apply a fully connected layer to the final output.

The created neural network is then put together. Here, one of the parameters in compilation is an optimizer. You need to minimise the loss function and modify the weight of each epoch when training a deep learning model. The loss function in this case is cross-entropy, and the optimizer we utilised is the Adam optimizer. We modelled the data and fit it. and let's check the model's accuracy during training. We display the accuracy and loss while the neural network is being trained (fig 4).

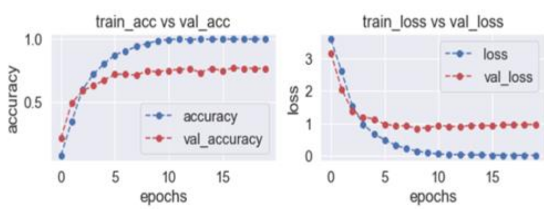

Fig 4. Accuracy during the training

4.5 Evaluation
When the model is tested, the label of the class is predicted, and the model correctly predicts both the label and the medical benefits of that specific class. We are currently assessing the model's performance using test data, and we will compare the outcomes. After putting the model to the test on test data, we obtained a 94% accuracy.

The first phase of the project is now complete. We will now use the tensorflow library to save our model in tensorflow lite format. We can use the model for our next procedures after saving it in that manner. The model is stored in a workable format and will have a simple user interface with the required buttons. In the Android Studio, the code to convert an image to bitmap format was written. We have included the Tensorflow Lite model to the Android Studio after some application development.

IV. RESULT

As the model accuracy was already covered in the evaluation section, the outcome of the mobile application is shown here. The good results we're obtaining are a result of the numerous approaches we attempted for more precision. At first, we considered utilising flask as a framework between Android Studio and our backend Python code. We considered adopting the springboot framework because of a college requirement that you use java programming as part of your project. After a brief investigation, we learned that the workflow, time complexity, and accuracy for the aforementioned usecases will slow down.

Below are pictures of the application's details and user interface. The whole application and its method are described below, along with how the leaf's medical qualities came to be recognised (fig 5.1). The user of this application is limited to two buttons: take a picture and open the gallery. When a user clicks the "take picture" button, the programme requests authorization to access the phone's camera (fig 5.2). if he agrees, the camera starts up, and he is able to show the leaf image. The customer can choose the leaf image he wants to use as input by pressing the "launch gallery" button, which opens the mobile's gallery.

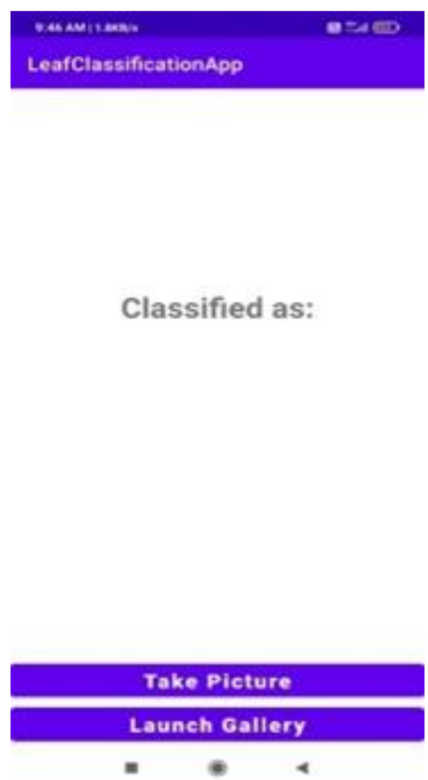

Fig 5.1

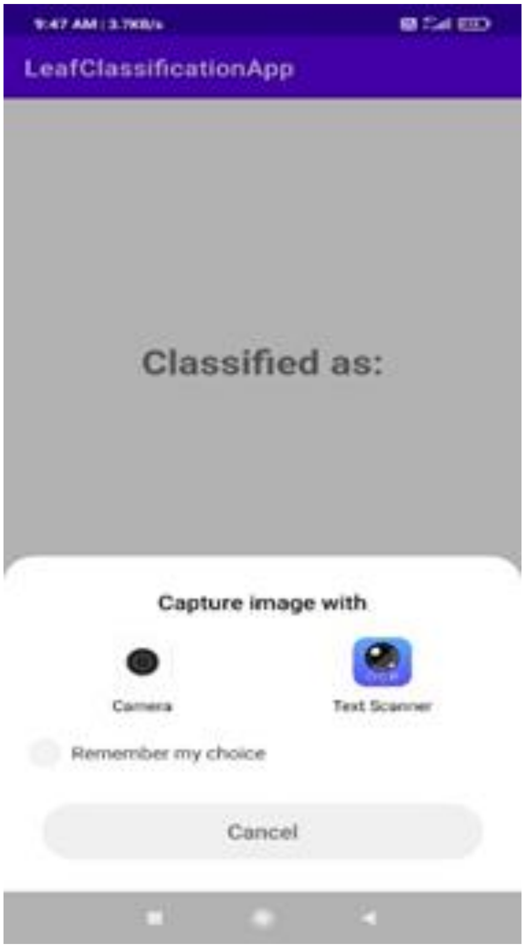

Fig 5.2

Here, we're providing a leaf-through gallery input (fig 5.3). The deep learning that is present in Android Studio is activated by the user interface, which receives input and processes it. The deep learning model functions and recognises the name of the leaf and its medicinal characteristics before sending that information to the Android studio, where the user will see the details. We have given the image of drumstick (fig 5.3) and it recognises as the leaf name as drumstick and also displayed its medicinal properties to the user (fig 5.4). You can check out the details below.

Fig 5.3

Fig 5.4

Here is another that we are providing to the application with the expectation that it will anticipate the needed information. The app correctly identified it as tulsi. Additionally, it demonstrates tulsi's medicinal properties. (fig 5.5 and fig 5.6)

Fig 5.5

Fig 5.6

V. DISCUSSION AND FUTURE SCOPE

Due to their therapeutic benefits, medicinal leaves have been utilised for centuries and are still a significant part of traditional medicine in many countries. As more and more uses and advantages for these herbal remedies are being uncovered by researchers, the future potential of medicinal leaves is bright. The public and researchers can use our programme to help them discover the medicinal characteristics of the leaves. Make an effort to help yourself with medicinal leaves before visiting the hospital for a minor issue. No matter how many generations pass, nature will always be there for us and to treat our illnesses. The generations should be familiar with specific leaves by then, therefore we anticipate that our application will be favourably useful to them in those situations. As the debates continue, it becomes clear that a bright future is possible if we can effectively utilise nature. so that it won't be impacted, and instead we.

CONCLUSION

Identifying medicinal leaves is an important first step in using traditional medicine and natural therapies, to sum up. The correct use of the plant, preventing any injury to the user, and maximising its medicinal advantages are all ensured by accurate identification. The general public is aware that they can obtain treatments from plants and other natural resources, but they lack access to information on the therapeutic benefits. The main motivation for our mobile app was this. Nowadays, everyone carries a smartphone in their hand, allowing them to access their medical needs directly from the device. Our "dhanvantri" programme serves two main functions: first, it identifies the leaf that the user has provided, and second, it displays the leaf's therapeutic benefits. To identify an image, we employ its available attributes, such as colour, shape, and dimension. The name of the image and its therapeutic characteristics were identified using several deep learning algorithms. Tensorflow is a key library that is utilised throughout the entire project, not just for deep learning but also for saving models and using them in Android Studio. We are pleased with the accuracy of 94%.

REFERENCES

[1] D. Forsyth and J. Ponce, Computer vision: A modern approach.Prentice hall, 2011

[2] Bi, Y., Xue, B., Mesejo, P., Cagnoni, S., & Zhang, M. (2023, February). A Survey on Evolutionary Computation for Computer Vision and Image Analysis: Past, Present, and Future Trends. IEEE Transactions on Evolutionary Computation.

[3] Medicinal Plant Identification using Android Application based on leaf image. (2020, December 18). European Journal of Molecular & Clinical Medicine, 7(09), 1496–1506. https://doi.org/10.31838/ejmcm.07.09.161

[4] Azadnia, R., Al-Amidi, M. M., Mohammadi, H., Cifci, M. A., Daryab, A., & Cavallo, E. (2022, November 2). An AI Based Approach for Medicinal Plant Identification Using Deep CNN Based on Global Average Pooling. Agronomy, 12(11), 2723. https://doi.org/10.3390/agronomy12112723

[5] Quach, B. M., Dinh, V. C., Pham, N., Huynh, D., & Nguyen, B. T. (2022, June 10). Leaf recognition using convolutional neural networks based features. Multimedia Tools and Applications, 82(1), 777–801. https://doi.org/10.1007/s11042-022-13199-y

[6] PLANT DISEASE DETECTION USING IMAGE PROCESSING AND MACHINE LEARNING ALGORITHM. (2020, July 3). Journal of Xidian University, 14(7). https://doi.org/10.37896/jxu14.7/012

[7] Ahmad, N., Asif, H. M. S., Saleem, G., Younus, M. U., Anwar, S., & Anjum, M. R. (2021, August 26). Leaf Image-Based Plant Disease Identification Using Color and Texture Features. Wireless Personal Communications, 121(2), 1139–1168. https://doi.org/10.1007/s11277-021-09054-2

[8] Dudi, B. (2019, August 25). Medicinal Plant Recognition based on CNN and Machine Learning. International Journal of Advanced Trends in Computer Science and Engineering, 8(4), 999–1003. https://doi.org/10.30534/ijatcse/2019/03842019

[9] Singh, S., & Bhamrah, M. S. (2015). Leaf identification using feature extraction and neural network. IOSR Journal of Electronics and Communication Engineering, 5, 134-140.

[10] D, F., R, M., N, M., J, R. P., & P, V. (2021, October 29). Identification and Classification of Tomato Leaf Diseases Using Machine Learning Techniques. Webology, 18(05), 1168–1175. https://doi.org/10.14704/web/v18si05/web18297