INTRODUCTION:-

This project aims to develop an AI-based system for accurately identifying medicinal plants using deep learning techniques and deploying the solution using Streamlit. The system takes input images of various medicinal plants and predicts the species based on learned patterns and features. Through this project, we seek to facilitate accurate plant identification for healthcare, conservation efforts, and herbal medicine practices.

The plants play an important role in nature, this project basically deals with plants identification and classification. The identification of medicinal plants holds paramountimportance in various fields, including traditional medicine, pharmaceuticals, and conservation efforts.

There are many methods for classification, such as Support Vector Machine classifier,Classification and Regression tree, Naive bayes analysis, Random Forest Classificationand Convolutional neural networks(CNNs) are employed to train the model on the extracted features. The trained model learns to associate these features with specific medicinal plant species, enabling reliable and efficient identification.

Methodology:

Assemble a diverse dataset of medicinal plant images, capturing various species, growth stages, and environmental conditions**.** Employ **image preprocessing** techniques to enhance image quality.

Extract key features like leaf **shape, color, and texture** critical for distinguishing plant species.

Blob detection:

In the area of image processing, Blob detection is a technique by which system can trace the movements of objects within frame. A blob is a group of pixels identifies as an object. This detection mechanism finds the blob’s position in successive image frames. The blob area must be defined before any detection of blob where Pixels with similar light values or color values are grouped together to find the blob.

Blob analysis:

Blob analysis identifies potential objects and puts a box around them. It finds the area of the blob and from the rectangular fit around each blob, the centroid of the object can be extracted for tracking the object. An additional rule that the ratio of area of blob to the area of rectangle around a blob should be greater than 0.4 ensures that unnecessary objects are not detected.

Project Objectives:

The objective of this paper, the machine learning is to use leaf images and extracted features, including shape, margin, and texture, to accurately identify different species of plants. Leaves, due to their volume, and unique characteristics, are an effective means of differentiating plant species.

System Requirements:

Hardware Requirements:

- Intel Pentium : i3,i5 or above.
- RAM (SD/DDR) : 512MB
- HDD/SSD : 256GB or above
- System bus : 32 bits or above
- RAM : 256MB or above
- Monitor : SVGA COLOR
- Keyboard : 108 keys
- Mouse : 2 buttons

Software Requirements:

For this Identification of medicinal plants project we have used python and it's libraries like keras, tensorflow, numpy, pytorch and etc.

1. Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python language is incredibly easy to use and learn implement. Python codes can be easily written and executed much faster than other programming languages.

2. Keras:

Keras is a user-friendly deep learning framework known for its simplicity and modularity, making it ideal for rapid prototyping and experimentation in neural network development.

3. Pytorch:

It is a fully featured framework for building deep learning models, which is machine learning module that commonly used for applications like image recognition and language processing.

4. Tensorflow:

This open source platform is developed by Google, which helps us to implement best practices for data automation, model tracking and model retraining.

5. Numpy:

Numpy is a python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform and matrices.

6. ImageDataGenerator:

ImageDataGenerator is a utility in Keras that generates batches of augmented image data. It can automatically preprocess images by scaling, resizing, and applying various transformations like rotation, shearing, and flipping, enhancing the training dataset's diversity and improving model performance.

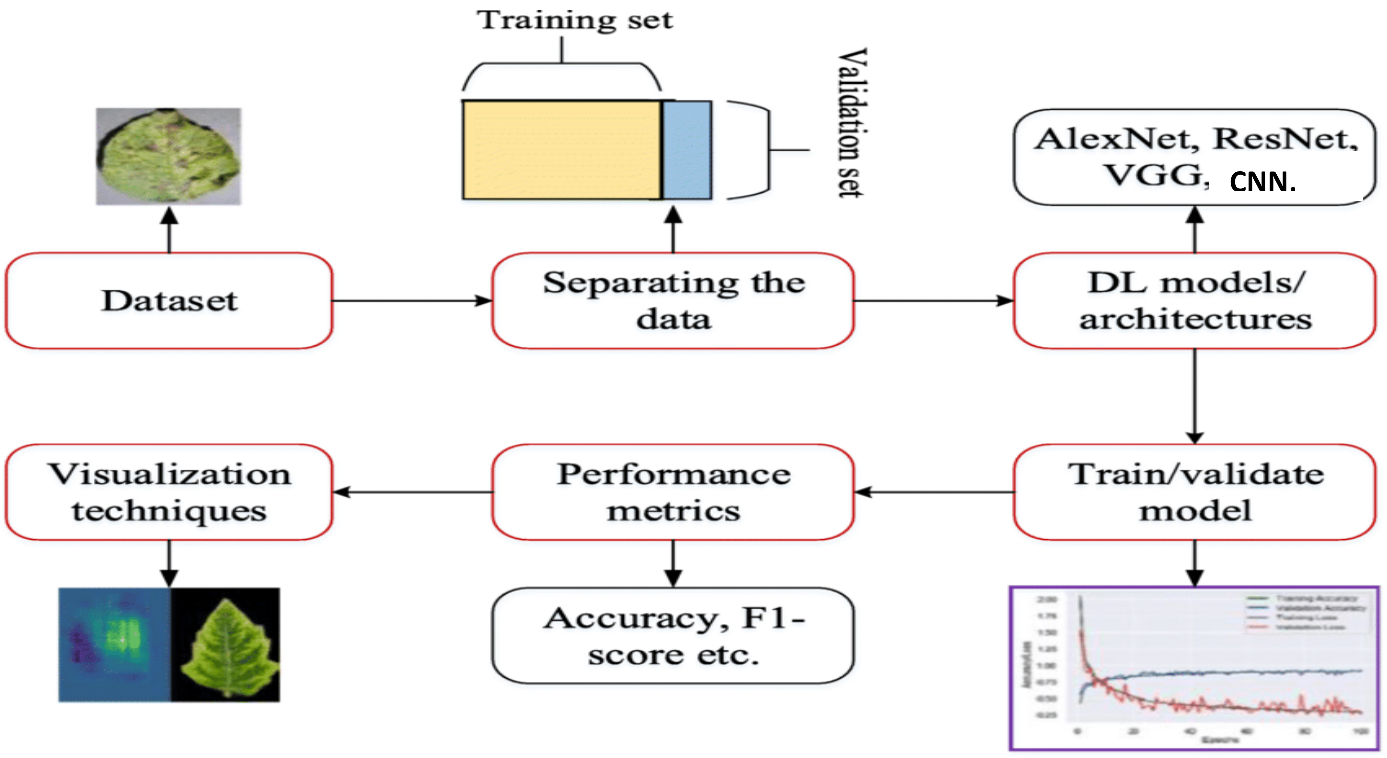
7. CNN(Convolutional Neural Network):

A Convolutional Neural Network (CNN) is a deep learning model designed specifically for processing and classifying images. It uses layers like convolutional, pooling, and fully connected layers to extract features and make predictions.

8. Matplotlib:

Matplotlib is a popular data visualization library in Python that provides a variety of plotting functions. It allows users to create plots such as line charts, scatter plots, histograms, and more, making it easy to visualize data and analyze trends.

Flowchart:



Training set
Validation set
AlexNet, ResNet, VGG, CNN.
Dataset
Separating the data
DL models/ architectures
Visualization techniques
Performance metrics
Train/validate model
Accuracy, F1-score etc.

Code:

#importing libraires
#Data preprocessing
#Build CNN model: a. Initialize the CNN
#b.Convolution
#c.Pooling(Step B and c to be repeated for each layer)
#d. Flattening
#e. Full connection
#f. O/p layer of this CNN
# Training the CNN
# Compile
# Evaluate

#importing libraires
```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator

class_info = {
   0: {"name": "Arive Dantu", "description": "Also known as Elephant Yam, it is used in traditional medicine for digestive disorders and as an anti-inflammatory."},
   1: {"name": "Basale", "description": "Commonly known as Malabar Spinach, it is rich in vitamins and minerals, aiding in digestion and promoting overall health."},
   2: {"name": "Betel", "description": "Betel leaves have antiseptic properties and are often chewed with betel nut for their stimulating and medicinal effects."},
   3: {"name": "Crape Jasmine", "description": "Used in Ayurveda for its anti-inflammatory properties, it is believed to have benefits for skin conditions and pain relief."},
   4: {"name": "Curry", "description": "Curry leaves are rich in antioxidants and are used in traditional medicine for their anti-diabetic and anti-inflammatory properties."},}(Given for all the classes)
```

#Data Preprocessing
```
train_datagen = ImageDataGenerator(
    rescale = 1./255,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True)
```

#preprocessing for training data
```
train_generator = train_datagen.flow_from_directory(
     '/content/drive/MyDrive/Plant Images Dataset/TRAIN',
     target_size=(64,64),
     batch_size=25,
     class_mode='sparse')
```

```
#preprocessing for testing_data
test_datagen = ImageDataGenerator(rescale=1./255)
validation_generator = test_datagen.flow_from_directory(
        '/content/drive/MyDrive/Plant Images Dataset/TEST',
      target_size=(64,64),
      batch_size=32,
      class_mode='sparse')

#Building the CNN
cnn = tf.keras.models.Sequential()

#Convolution1
cnn.add(tf.keras.layers.Conv2D(filters = 32,kernel_size =3,activation = 'relu',input_shape=[64,64,3]))
#pooling1
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))

#Convoluion2
cnn.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3,activation="relu"))
#Pooling2
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))

#Flattening
cnn.add(tf.keras.layers.Flatten())

#Full Connection(Dense Layer)
cnn.add(tf.keras.layers.Dense(units=128,activation='relu'))

  #Output Layer
num_classes = 30
cnn.add(tf.keras.layers.Dense(units=num_classes,activation='softmax'))

# #Compiling the model
cnn.compile(optimizer='adam',metrics=['accuracy'],loss='sparse_categorical_crossentropy')

#Training the model
history = cnn.fit(x=train_generator,validation_data = validation_generator, epochs=100, batch_size = 25)
 print(history)
```

```
#Image loading and testing
from tensorflow.keras.preprocessing import image
import numpy as np

img_size = (64,64)

# Load a sample image for prediction
img_path = "/content/drive/MyDrive/Plant Images Dataset/TEST/Test Lemon/CL-S-011.jpg"
# img_path = input("Upload your image")  # Replace with your image path
img = image.load_img(img_path, target_size=img_size)
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)
img_array /= 255.0  # Normalize the image

# Make predictions
predictions = cnn.predict(img_array)

# Get the predicted class index
predicted_class_index = np.argmax(predictions)

# Use the class mapping from the generator
class_indices_to_names = train_generator.class_indices

# Invert the mapping to get plant names from class indices
class_names_to_indices = dict((v, k) for k, v in class_indices_to_names.items())

predicted_class_info = class_info[predicted_class_index]
predicted_class_name = predicted_class_info["name"]
predicted_class_description = predicted_class_info["description"]

print(f"The predicted plant is: {predicted_class_name}")
print(f"Description: {predicted_class_description}")

cnn.save("model.h5")
```

Conclusion:

The integration of machine learning for the identification of medicinal plants through image processing presents a promising and innovative solution. The developed methodology, comprising data collection, feature extraction, and machine learning model training, demonstrates the potential to automate and enhance the accuracy of plant identification. The project's objectives of creating a user-friendly and accurate identification system align with the growing need for efficient tools in traditional medicine, pharmaceuticals, and biodiversity conservation.