

# med-plant-identify-cnn

December 24, 2023

```
[ ]: import os
import cv2
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator

original_images_dir = r'\orginal_images'

augmented_images_dir = r'\augmented'

datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2
    brightness_range=[0.6,1.3],
    channel_shift_range=20.0,
    fill_mode='constant'
)

image_files = os.listdir(original_images_dir)

for image_file in image_files:

    img = cv2.imread(os.path.join(original_images_dir, image_file))
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    img = np.expand_dims(img, axis=0)

    aug_iter = datagen.flow(img, batch_size=1,
                           save_to_dir=augmented_images_dir, save_prefix='aug', save_format='jpg')
```

```

for i in range(1):
    next(aug_iter)

print("Augmentation complete. Augmented images saved in", augmented_images_dir)

```

## 0.1 LOADING TRAIN AND TEST DATASET USING IMAGE GENERATOR

```

[1]: import pandas as pd
import tensorflow as tf
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator

[46]: direct = r'C:\Users\loges\Downloads\Dataset-medecinal_plants\dataset_train'
batch_size = 32
target_size = (224, 224)

train_datagen = ImageDataGenerator(
    rescale=1.0 / 255,
    validation_split=0.2
)

train_generator = train_datagen.flow_from_directory(
    direct,
    target_size=target_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=True
)

class_len = len(train_generator.class_indices)

print(class_len)
#print(class_name)

```

Found 20042 images belonging to 29 classes.  
29

```
[5]: direct_test = r'C:\Users\loges\Downloads\Dataset-medecinal_plants\dataset_test'
batch_size = 32
```

```

target_size = (224, 224)

test_datagen = ImageDataGenerator(
    rescale=1.0 / 255
)

test_generator = test_datagen.flow_from_directory(
    direct_test,
    target_size=target_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=True
)

class_len = len(test_generator.class_indices)

print(class_len)
#print(class_name)

```

Found 1685 images belonging to 29 classes.  
29

```
[6]: import matplotlib.pyplot as plt

images, labels = next(train_generator)
print(images.shape)

class_labels = list(train_generator.class_indices.keys())

print(class_labels)
def plot_images(images, labels, class_labels, nrows=5, ncols=5):
    plt.figure(figsize=(15, 15))
    for i in range(nrows * ncols):
        plt.subplot(nrows, ncols, i + 1)
        plt.imshow(images[i])
        true_label = class_labels[labels[i].argmax()]
        plt.title(f'True: {true_label}')
        plt.axis('off')
    plt.tight_layout()
    plt.show()
```

```
plot_images(images, labels, class_labels)
```

(32, 224, 224, 3)

['Aloevera', 'Avaram', 'Black-HoneyShrub-karupu\_nelli', 'Eucalyptus',  
'Hibiscus', 'IndianStingingNettle-kangerisondathi', 'IvyGourd-kovaikai',  
'SmallWaterClover-arraikirai', 'Turmeric', 'asthmaplant-toothuvalai',  
'balloonvine-mudakathan', 'butterflypea-sungupuspham', 'capegooseberry-  
pillathakali', 'coconut', 'corainder', 'crownflower-erukam', 'curryleaves',  
'indiancopperleaf-kuppaimeni', 'indianmustard', 'jackfruit', 'jamun', 'jasmine',  
'leamon', 'mango', 'mint', 'neem', 'tobacco', 'tulasi', 'waterapple']



## 0.2 CNN MODEL USING VGG19 AS FEATURE EXTRACTOR TOP LAYER AN COSTUM CNN LAYERS

```
[7]: from tensorflow.keras.models import Sequential,Model
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D,BatchNormalization
from tensorflow.keras.applications import VGG19

base_model = VGG19(weights='imagenet', include_top=False,input_shape=(224, 224, 3))
for layer in base_model.layers:
    layer.trainable = False
```

```
[8]: base_model.summary()
```

Model: "vgg19"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808

block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0

---

Total params: 20,024,384  
Trainable params: 0  
Non-trainable params: 20,024,384

---

```
[21]: fex=Sequential()
fex.add(base_model)
fex.add(Conv2D(512, (3, 3), activation='relu',kernel_initializer='he_uniform'))
#fex.add(Conv2D(256, (3, 3), activation='relu',kernel_initializer='he_uniform'))
fex.add(BatchNormalization())
fex.add(MaxPooling2D((2,2)))
fex.add(Flatten())

x=fex.output
x = Dense(128, activation='relu',kernel_initializer='he_uniform')(x)
x=Dropout(0.3)(x)
x = Dense(64, activation='relu',kernel_initializer='he_uniform')(x)
op = Dense(29, activation='softmax')(x)

model = Model(inputs=fex.input, outputs=op)

model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['accuracy'])
#fex.summary()
model.summary()
```

Model: "model\_7"

---

Layer (type)	Output Shape	Param #
vgg19_input (InputLayer)	[(None, 224, 224, 3)]	0

---

vgg19 (Functional)	(None, 7, 7, 512)	20024384
conv2d_10 (Conv2D)	(None, 5, 5, 512)	2359808
batch_normalization_7 (BatchNormalization)	(None, 5, 5, 512)	2048
max_pooling2d_7 (MaxPooling2D)	(None, 2, 2, 512)	0
flatten_7 (Flatten)	(None, 2048)	0
dense_21 (Dense)	(None, 128)	262272
dropout_7 (Dropout)	(None, 128)	0
dense_22 (Dense)	(None, 64)	8256
dense_23 (Dense)	(None, 29)	1885

---

Total params: 22,658,653  
Trainable params: 2,633,245  
Non-trainable params: 20,025,408

---

```
[49]: history= model.fit(
    train_generator,
    steps_per_epoch=len(train_generator)//32,
    epochs=25
)
```

```
Epoch 1/25
19/19 [=====] - 5s 232ms/step - loss: 0.0746 -
accuracy: 0.9737
Epoch 2/25
19/19 [=====] - 5s 249ms/step - loss: 0.1112 -
accuracy: 0.9622
Epoch 3/25
19/19 [=====] - 5s 240ms/step - loss: 0.1399 -
accuracy: 0.9539
Epoch 4/25
19/19 [=====] - 5s 242ms/step - loss: 0.1309 -
accuracy: 0.9605
Epoch 5/25
19/19 [=====] - 5s 238ms/step - loss: 0.1114 -
```

```
accuracy: 0.9539
Epoch 6/25
19/19 [=====] - 5s 234ms/step - loss: 0.0905 -
accuracy: 0.9655
Epoch 7/25
19/19 [=====] - 5s 235ms/step - loss: 0.0802 -
accuracy: 0.9786
Epoch 8/25
19/19 [=====] - 5s 230ms/step - loss: 0.0928 -
accuracy: 0.9704
Epoch 9/25
19/19 [=====] - 5s 261ms/step - loss: 0.0878 -
accuracy: 0.9770
Epoch 10/25
19/19 [=====] - 5s 257ms/step - loss: 0.0844 -
accuracy: 0.9737
Epoch 11/25
19/19 [=====] - 5s 252ms/step - loss: 0.0664 -
accuracy: 0.9803
Epoch 12/25
19/19 [=====] - 5s 257ms/step - loss: 0.1023 -
accuracy: 0.9655
Epoch 13/25
19/19 [=====] - 5s 254ms/step - loss: 0.1148 -
accuracy: 0.9704
Epoch 14/25
19/19 [=====] - 5s 242ms/step - loss: 0.1412 -
accuracy: 0.9589
Epoch 15/25
19/19 [=====] - 5s 254ms/step - loss: 0.0716 -
accuracy: 0.9770
Epoch 16/25
19/19 [=====] - 5s 253ms/step - loss: 0.0605 -
accuracy: 0.9803
Epoch 17/25
19/19 [=====] - 5s 282ms/step - loss: 0.0909 -
accuracy: 0.9720
Epoch 18/25
19/19 [=====] - 5s 243ms/step - loss: 0.0810 -
accuracy: 0.9786
Epoch 19/25
19/19 [=====] - 5s 250ms/step - loss: 0.0543 -
accuracy: 0.9836
Epoch 20/25
19/19 [=====] - 5s 253ms/step - loss: 0.0969 -
accuracy: 0.9704
Epoch 21/25
19/19 [=====] - 5s 259ms/step - loss: 0.0374 -
```

```
accuracy: 0.9868
Epoch 22/25
19/19 [=====] - 5s 249ms/step - loss: 0.0993 -
accuracy: 0.9655
Epoch 23/25
19/19 [=====] - 5s 241ms/step - loss: 0.1153 -
accuracy: 0.9605
Epoch 24/25
19/19 [=====] - 5s 257ms/step - loss: 0.1100 -
accuracy: 0.9688
Epoch 25/25
19/19 [=====] - 5s 252ms/step - loss: 0.1110 -
accuracy: 0.9655
```

```
[60]: evaluation_result = model.evaluate(
    test_generator,
    steps=len(test_generator)//32,
    verbose=1 )
print("Test loss:", evaluation_result[0])
print("Test accuracy:", evaluation_result[1])
```

```
1/1 [=====] - 0s 456ms/step - loss: 0.2384 - accuracy:
0.9688
Test loss: 0.23844212293624878
Test accuracy: 0.96875
```

## 0.2.1 OUR MODEL HAS TEST ACCURACY OF 96% AND TRAIN ACCURACY OF 97%

```
[61]: model.save('medecinal_plant_identify_cnn.h5')
```

```
[62]: fex.save('feature_extractor_cnn.h5')
```

```
WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet
to be built. `model.compile_metrics` will be empty until you train or evaluate
the model.
```

## 0.3 FUNCTION TO PERFORM PREDECTION OF MEDICENAL PLANTS

```
[63]: import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image
```

```

cn=['Aloeevera', 'Avaram', 'Black-HoneyShrub-karupu_nelli', 'Eucalyptus',  

↳ 'Hibiscus', 'IndianStingingNettle-kangerisondathi', 'IvyGourd-kovaikai',  

↳ 'SmallWaterClover-arraikirai', 'Turmeric', 'asthmaplant-toothuvalai',  

↳ 'balloonvine-mudakathan', 'butterflypea-sungupuspham',  

↳ 'capegooseberry-pillaithakali', 'coconut', 'corainder',  

↳ 'crownflower-erukam', 'curryleaves', 'indiancopperleaf-kuppaimeni',  

↳ 'indianmustard', 'jackfruit', 'jamun', 'jasmine', 'leamon', 'mango', 'mint',  

↳ 'neem', 'tobacco', 'tulasi', 'waterapple']

def predict_plant(path):
    image_path = path
    img = image.load_img(image_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    plt.imshow(img)
    plt.axis('off')
    plt.show()

    img_array = np.expand_dims(img_array, axis=0)
    img_array /= 255.0
    #predictions = model.predict(img_array)
    predictions = model.predict(img_array)
    fp = [":.2f)".format(prob) for prob in predictions[0]]
    predicted_class = np.argmax(predictions)
    predicted_label = cn[predicted_class]

    print("Deep Learning Model")

    #print(f"Predection probab:{predictions}")
    print(f"Predection probab:{fp}")
    print("Predicted class label:", predicted_label)

predict_plant(r'C:\Users\loges\Downloads\imgalo.jpg')

```



```
1/1 [=====] - 0s 184ms/step
Deep Learning Model
Prediction probab:['1.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: Aloevera
```

```
[66]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\tulasi.
    ↴jpg')
```



```
1/1 [=====] - 0s 24ms/step
```

```
Deep Learning Model
```

```
Predetion probab:['0.00', '0.00', '0.01', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.98', '0.00']
```

```
Predicted class label: tulasi
```

```
[88]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\tulsi.
    ↴jpg')
```



```
1/1 [=====] - 0s 19ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'1.00', '0.00']
Predicted class label: tulasi
```

```
[68]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\cor.jpg')
```



```
1/1 [=====] - 0s 19ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '1.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: corainder
```

```
[69]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\cl.jpg')
```



```
1/1 [=====] - 0s 20ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.89',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.11',
'0.00', '0.00']
Predicted class label: curryleaves
```

```
[70]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\hyb1.jpg')
```



```
1/1 [=====] - 0s 19ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.90', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.01', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: Hibiscus
```

```
[72]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\mint.jpg')
```



```
1/1 [=====] - 0s 20ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.05', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.95',
'0.00', '0.00', '0.00', '0.00']
Predicted class label: mint
```

```
[77]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\cf.webp')
```



```
1/1 [=====] - 0s 19ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '1.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: crownflower-erukam
```

```
[78]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\bfp.jpg')
```



```
1/1 [=====] - 0s 20ms/step
Deep Learning Model
Prediction probab:['0.00', '0.01', '0.04', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.01', '0.84', '0.00', '0.00', '0.00', '0.01',
'0.00', '0.00', '0.01', '0.01', '0.00', '0.05', '0.01', '0.00', '0.00', '0.00',
'0.01', '0.00']
Predicted class label: butterflypea-sungupuspham
```

```
[79]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\smci.jpg')
```



```
1/1 [=====] - 0s 18ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'1.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: SmallWaterClover-arraikirai
```

```
[80]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\j.jpg')
```



```
1/1 [=====] - 0s 20ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '1.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: jackfruit
```

```
[81]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\nem.jpg')
```



```
1/1 [=====] - 0s 18ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: neem
```

```
[82]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\lemon.
    ↴webp')
```



```
1/1 [=====] - 0s 18ms/step
Deep Learning Model
Prediction probab:['0.00', '0.07', '0.11', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.03', '0.02', '0.00', '0.00', '0.00', '0.22', '0.00',
'0.00', '0.00', '0.07', '0.00', '0.00', '0.30', '0.16', '0.00', '0.00', '0.00',
'0.00', '0.01']
Predicted class label: leamon
```

```
[83]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\turmeric.
    ↴jpg')
```



```
1/1 [=====] - 0s 19ms/step
```

```
Deep Learning Model
```

```
Predecture probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '1.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
```

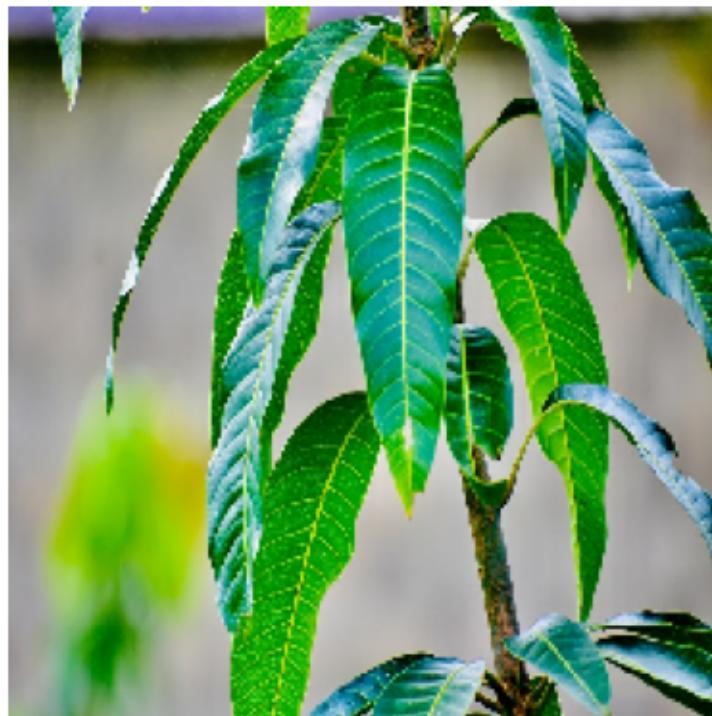
```
Predicted class label: Turmeric
```

```
[84]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\Miyazaki-Mango-plant.
    ↴webp')
```



```
1/1 [=====] - 0s 24ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '1.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: mango
```

```
[85]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\manga.
    ↴jpg')
```



```
1/1 [=====] - 0s 20ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '1.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: mango
```

```
[86]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\le.webp')
```



```
1/1 [=====] - 0s 18ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '1.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: leamon
```

```
[90]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\bv.jpg')
```



```
1/1 [=====] - 0s 23ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '1.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: balloonvine-mudakathan
```

```
[91]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\Eucalyptus.
    ↴jpg')
```



```
1/1 [=====] - 0s 19ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '1.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: Eucalyptus
```

```
[92]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\cpg.jpg')
```



```
1/1 [=====] - 0s 21ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.03', '0.01', '0.74', '0.00', '0.00', '0.00', '0.00',
'0.01', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.01', '0.00',
'0.20', '0.00']
Predicted class label: capegooseberry-pillaithakali
```

```
[93]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\ic1.jpg')
```



```
1/1 [=====] - 0s 18ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.02', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.01', '0.00', '0.00', '0.00', '0.02', '0.00',
'0.71', '0.00', '0.00', '0.00', '0.00', '0.00', '0.04', '0.00', '0.15',
'0.00', '0.03', '0.00']
Predicted class label: indiancopperleaf-kuppaimeni
```

```
[94]: predict_plant(r'C:
    ↴\Users\loges\Downloads\Dataset-medecinal_plants\augumented_images\jas.jpg')
```



```
1/1 [=====] - 0s 26ms/step
Deep Learning Model
Prediction probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.99', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00']
Predicted class label: jasmine
```

```
[99]: predict_plant(r'C:\Users\loges\Downloads\water.jpg')
```



```
1/1 [=====] - 0s 18ms/step
```

```
Deep Learning Model
```

```
Predecture probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',
'0.00', '0.00', '1.00']
```

```
Predicted class label: waterapple
```

```
[100]: predict_plant(r'C:\Users\loges\Downloads\jam.jpg')
```



```
1/1 [=====] - 0s 18ms/step
```

```
Deep Learning Model
```

```
Predecture probab:['0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',  
'0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',  
'0.00', '0.00', '0.00', '1.00', '0.00', '0.00', '0.00', '0.00', '0.00', '0.00',  
'0.00', '0.00']
```

```
Predicted class label: jamun
```