SAMPLE CODE

```
import os
import cv2
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.models import load_model

# Set the path to your dataset
dataset_path = '/md.png.csv'

# Set the image dimensions
img_width, img_height = 224, 224

# Set the number of classes (plant species)
num_classes = 10  # Adjust this to the number of plant species in your dataset

# Create a data generator for training and validation
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)

validation_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_generator = train_datagen.flow_from_directory(
    os.path.join(dataset_path, 'train'),
    target_size=(img_width, img_height),
    batch_size=32,
    class_mode='categorical')

validation_generator = validation_datagen.flow_from_directory(
    os.path.join(dataset_path, 'validation'),
    target_size=(img_width, img_height),
    batch_size=32,
    class_mode='categorical')

# Create the CNN model
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(img_width,
img_height, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
```

```
            metrics=['accuracy'])

# Train the model
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // 32,
    epochs=10,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // 32)

# Save the model
model.save('plant_identification_model.h5')

# Load the model (if you want to use it later)
# model = load_model('plant_identification_model.h5')

# Define a function to identify a plant from an image
def identify_plant(image_path):
    img = cv2.imread(image_path)
    img = cv2.resize(img, (img_width, img_height))
    img = img / 255.0
    img = np.expand_dims(img, axis=0)
    predictions = model.predict(img)
    class_id = np.argmax(predictions)
    class_names = ['plant1', 'plant2', 'plant3', 'plant4', 'plant5', 'plant6', 'plant7',
'plant8', 'plant9', 'plant10']  # Adjust this to your plant species names
    return class_names[class_id]
```

```
# Test the function
image_path = '/laurece.jpeg'
print(identify_plant(image_path))  # Output: The name of the plant
```