Medicinal Plants Recognition Using Deep Learning

Conference Paper · June 2023

DOI: 10.1109/MCNA59361.2023.10185880

4 authors, including:

Yousef Sharrab
Isra University

Dimah Al-Fraihat
University of Warwick

Monther Tarawneh
Tafila Technical University

Medicinal Plants Recognition Using Deep Learning

1st Yousef Sharrab
Dept. of Computer Science
Isra University
Amman, Jordan
sharrab@iu.edu.jo

2nd Dimah Al-Fraihat
Dept. of Software Engineering
Isra University
Amman, Jordan
d.fraihat@iu.edu.jo

3rd Monther Tarawneh
Dept. of Computer Science
Isra University
Amman, Jordan
mtarawneh@iu.edu.jo

4th Ahmad Sharieh
dept. of Computer Science
The University of Jordan
Amman, Jordan
Sharieh@ju.edu.jo

***Abstract*—The use of medicinal plants has been a long-standing practice in traditional medicine worldwide. Accurately identifying medicinal plants is crucial for determining their medicinal properties and potential applications. However, it can be a challenging task due to the complexity of their appearance. Variations in growth stage, lighting, and imaging conditions can make classification challenging, which limits the application of traditional methods for plant identification. This paper proposes a deep learning-based approach that uses a convolutional neural network (CNN) based on the VGG-16 model. With a dataset of 25,686 images, the CNN is capable of learning and representing complex features in images, enabling it to recognize and classify medicinal plants with high accuracy. The proposed approach can efficiently classify plants with different growth stages, lighting conditions, and imaging settings, providing a reliable tool for plant identification. We achieved an impressive recognition rate of 98%, demonstrating the feasibility of using deep learning techniques for accurate plant classification. The proposed approach has enormous potential for providing healthcare professionals and herbal medicine researchers with a reliable tool for identifying herbal plants. the study represents an essential advancement in the use of deep learning techniques for medicinal plant recognition, overcoming the challenges posed by their complex appearance. The proposed approach has far-reaching implications and can significantly impact the field of herbal medicine research, enabling researchers and healthcare professionals to identify and classify medicinal plants more accurately**

***Index Terms*—Deep Learning, Medicinal Plants Recognition, Convolutional Neural Network (CNN), VGG-16, Herbal Medicine, Herbal Plant Identification**

”

I. INTRODUCTION

Herbal medicine has a long history as an integral component of traditional healthcare systems. However, accurately identifying herbal plants can be a challenge, especially for individuals without specialized knowledge in botany and plant systematics [1]. In recent years, there has been growing interest in utilizing computer vision [2] and deep learning techniques to aid in the identification of herbal plants [3]–[6]. These techniques have the potential to provide a fast and accurate method for plant identification, making them valuable tools for healthcare professionals and researchers in the field of herbal medicine. Accurately identifying medicinal plants is crucial for the development of new treatments and drugs. Traditional methods of plant identification, such as manual inspection and morphological analysis, can be time-consuming and prone to errors, while deep learning-based methods have shown promise in various image recognition tasks and have the potential to significantly improve the accuracy and efficiency of herbal plant recognition [7].

This paper proposes a deep learning-based approach to recognizing and identifying herbal medicinal plants, which consists of two main components: a dataset and a model architecture. The dataset used in this study is a collection of images of various plant species, which were collected using mobile phones in natural settings. The model architecture is a VGG-16-based deep learning model consisting of 5 residual building blocks [8].

The main objectives of this research are to evaluate the performance of the proposed deep learning-based system and compare it to traditional machine learning methods. The experiments were conducted on our collected dataset and showed that our model recognition rate was 98.0% with a loss of around 0.04. The VGG-16 architecture achieved a top-5 accuracy of 89.0% on the ImageNet dataset according to the original 2014 paper [9]. This accuracy may not be considered state-of-the-art anymore, as more recent architectures and techniques have surpassed these results. The actual accuracy of a VGG-16 implementation on the ImageNet dataset can vary based on factors such as the training data, implementation, and procedure. The research questions and objectives include: (1) Can a deep learning-based system accurately recognize and identify herbal medicinal plants using a dataset of images?, (2) How does the proposed deep learning-based system compare to traditional machine learning methods for recognizing and identifying herbal medicinal plants?, (3) How can the performance of the deep learning-based system be improved?, and (4) How well can the system perform in natural environments?

II. BACKGROUND

The use of medicinal plants in traditional medicine has been a common practice for centuries. With the advancement of technology, the identification and classification of medicinal plants have become a crucial task in the fields of botany and pharmacology. Deep learning, a subset of machine learning, has been increasingly used for image classification tasks and has shown to be highly efficient in identifying and classifying objects in images. Herbal medicine, also known as phytotherapy, uses plants or plant-derived substances for medicinal

Identify applicable funding agency here. If none, delete this.

purposes. It has been an important part of traditional healthcare systems for centuries and is still widely used in many cultures worldwide. Herbal medicine is considered a safer and more natural alternative to conventional medicine. It is often used to treat many conditions, including chronic diseases, common illnesses, and mental health issues [10]. However, identifying herbal plants can be challenging, especially for individuals without specialized knowledge in botany and plant systematics. Proper identification of plants is crucial for the safe and effective use of herbal medicine, as many plants can be toxic or have dangerous interactions with other medications. Furthermore, many plants have similar appearances, making it difficult to distinguish between them [11].

Another challenge in identifying herbal plants is that many have been traditionally used in multiple cultures and regions and may have different common names and uses depending on the location. This can make it difficult to find accurate information about a plant and its medicinal properties [12]. Additionally, in recent years, the loss of biodiversity and the destruction of natural habitats has led to the extinction of many plant species, which makes it even harder to find the required herbal plants [13]. These challenges highlight the importance of accurately identifying herbal plants and the need for a fast and accurate method for identifying plants, such as using computer vision and deep learning techniques. A fast and accurate method for identifying plants is crucial for the safe and effective use of herbal medicine [14]. As mentioned earlier, many plants have similar appearances, making it difficult to distinguish between them. Furthermore, traditional plant identification methods, such as using physical characteristics or microscopic analysis, can be time-consuming and require specialized knowledge and equipment. This is where deep learning can play a crucial role, as it can provide a fast and accurate method for identifying plants. Deep learning is a subset of machine learning that uses neural networks modelled after the human brain. These neural networks can learn to recognize patterns and make predictions based on large amounts of data [15].

In identifying plants, deep learning can be used to analyze images of plants and classify them based on their visual characteristics. By training a deep learning model on a large dataset of images of various plant species, the model can learn to recognize the unique features of each species and accurately classify new images [16]. Deep learning models can also be used to extract features from the images and then classify the plants. Moreover, using deep learning can also help reduce the time and cost associated with the manual identification of plants. In addition, deep learning methods can process large amounts of data in a short period of time. This makes it a valuable tool for healthcare professionals and researchers in the field of herbal medicine [17].

III. Related Work

There has been an increasing interest in using computer vision and deep learning techniques recently to identify herbal plants. These methods have been shown to be fast and accurate in comparison to traditional methods of plant identification, such as manual inspection and morphological analysis.

One approach to identifying herbal plants is through the use of convolutional neural networks (CNNs). CNNs have been used to classify images of plants based on their leaf shape, color, and texture [3]. In [4], a CNN-based approach was used to classify medicinal plants based on their leaf images, achieving high accuracy compared to traditional machine learning methods. Another study [6] proposed an optimized CNN model for medicinal plant recognition using a dataset of leaf images, achieving high classification accuracy.

Another approach for identifying herbal plants is using deep learning techniques to extract features from images of plants and then using traditional machine learning algorithms for classification. In [1], a feature extraction technique based on deep learning was used to classify medicinal plants, achieving high accuracy compared to traditional methods.

Overall, the above studies have shown the potential of deep learning-based methods in the identification of herbal plants, and have demonstrated their ability to improve accuracy and efficiency compared to traditional methods. However, there is still a need for further research in this field to improve the accuracy and generalization of these methods.

The VGG-16 architecture 1 is a convolutional neural network architecture that was developed by the Visual Geometry Group at the University of Oxford [9]. The VGG-16 model is characterized by its use of very small convolutional filters (3x3) and very deep architectures (16 weight layers) which help to improve the representation of the input image. It is widely used for image classification and other computer vision tasks.

The VGG-16 architecture consists of a sequence of convolutional, max pooling, and fully connected layers. The convolutional layers use a set of filters to extract features from the input image, while the max pooling layers reduce the spatial dimensions of the feature maps. The fully connected layers, also known as the dense layers, are used to perform classification on the features extracted by the convolutional layers [18].

The architecture of VGG-16 can be divided into two main parts: the convolutional part, which is composed of 13 convolutional layers, and the fully connected part, which is composed of 3 fully connected layers. The convolutional layers are divided into 5 blocks, each block having 2 or 3 convolutional layers, and each convolutional layer is followed by a max pooling layer [19]. The fully connected part is connected to the last max pooling layer of the convolutional part, and it's followed by a softmax activation function to produce the final class scores. A full mathematical representation of the VGG-16 architecture is shown in Equations 1 through 16.

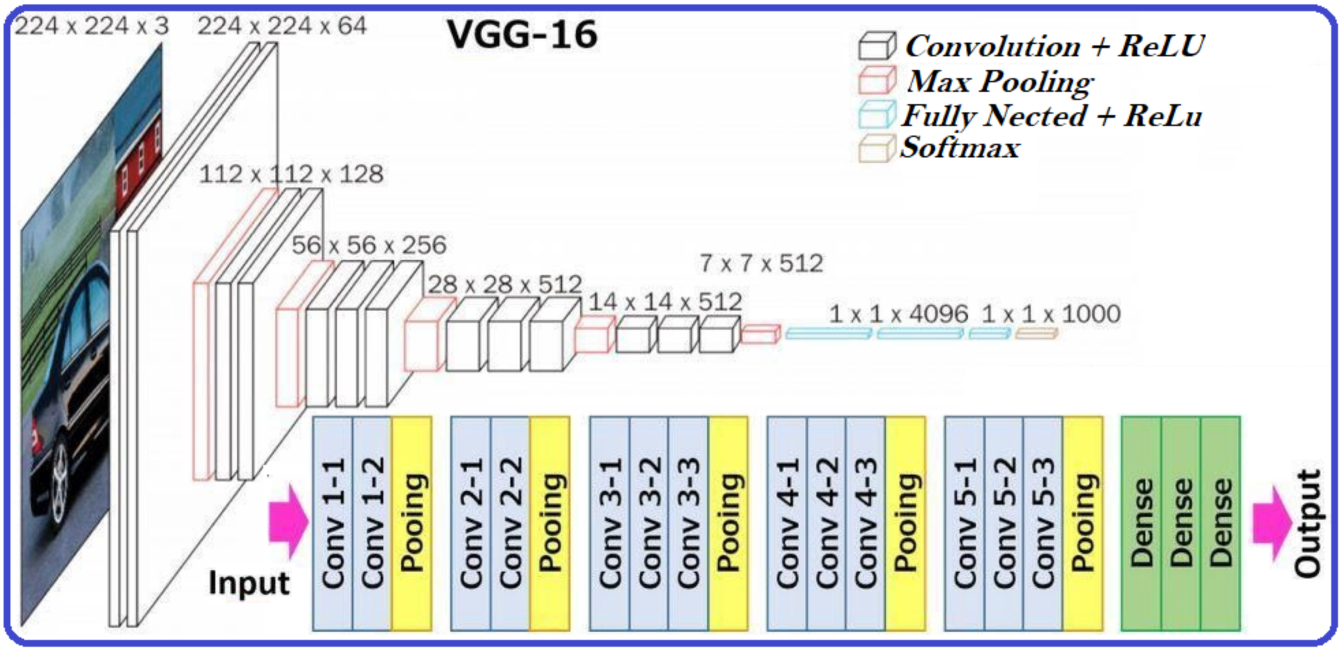

Fig. 1. VGG-16 Model

$$h_1 = \mathrm{ReLU}(\mathrm{conv2d}(x, W_1) + b_1) \tag{1}$$
$$h_2 = \mathrm{maxpool}(h_3) \tag{2}$$
$$h_3 = \mathrm{ReLU}(\mathrm{conv2d}(h_2, W_2) + b_2) \tag{3}$$
$$h_4 = \mathrm{maxpool}(h_3) \tag{4}$$
$$h_5 = \mathrm{ReLU}(\mathrm{conv2d}(h_4, W_5) + b_5) \tag{5}$$
$$h_6 = \mathrm{maxpool}(h_5) \tag{6}$$
$$h_7 = \mathrm{ReLU}(\mathrm{conv2d}(h_6, W_7) + b_7) \tag{7}$$
$$h_8 = \mathrm{maxpool}(h_7) \tag{8}$$
$$h_9 = \mathrm{ReLU}(\mathrm{conv2d}(h_8, W_9) + b_9) \tag{9}$$
$$h_{10} = \mathrm{maxpool}(h_9) \tag{10}$$
$$h_{11} = \mathrm{ReLU}(\mathrm{conv2d}(h_{10}, W_{11}) + b_{11}) \tag{11}$$
$$h_{12} = \mathrm{maxpool}(h_{11}) \tag{12}$$
$$x = \mathrm{flatten}(h_{12}) \tag{13}$$
$$h_{13} = \mathrm{ReLU}(xW_{13} + b_{13}) \tag{14}$$
$$h_{14} = \mathrm{ReLU}(xW_{14} + b_{14}) \tag{15}$$
$$y = \mathrm{softmax}(xW_{15} + b_{15}) \tag{16}$$

where, h_i represents the output feature map of the ith layer, x is the flattened feature map from the previous layer, W_i and b_i are the weight and bias parameters of the ith layer, respectively, and 'flatten' is the operation that flattens a multi-dimensional feature map into map into a single vector. The final layer produces the class scores for the input image, represented by y.

Note that this representation assumes the VGG-16 architecture consists of 5 blocks, each with 2 convolutional layers followed by a max pooling layer.

The explanation of Equations in 1 through 16 are as follows:

- ReLU (Rectified Linear Unit): ReLU is an activation function, which is used to introduce non-linearity in the model. The equation for ReLU is given by: $f(x) = max(0, x)$, where x is the input to the activation function. ReLU is applied element-wise to the output of each layer [20].
- Conv2d: Conv2d is a 2-dimensional convolutional layer that applies filters to the input feature maps to produce the output feature maps. The equation for conv2d can be written as: $h_i = conv2d(h_{i-1}, W_i) + b_i$, where h_{i-1} is the output feature map from the previous layer, W_i is the set of filters for the ith layer and b_i is the bias term for the ith layer [21].
- Maxpool: Maxpool is a pooling layer that reduces the spatial dimensions of the feature maps. The maxpool operation is applied by dividing the feature maps into non-overlapping regions and taking the maximum value from each region. The equation for maxpool can be written as: $h_i = maxpool(h_{i-1})$, where h_{i-1} is the output feature map from the previous layer [22].
- Flatten: Flatten is an operation that flattens a multi-dimensional feature map into a single vector. The flatten operation can be represented as: $x = flatten(h_{12})$, where h_{12} is the output feature map from the previous layer and x is the flattened feature map [23].
- Softmax: Softmax is an activation function that is used in the final layer of the VGG-16 model to produce class scores for the input image. The equation for softmax can be written as: $y = softmax(xW_{15} + b_{15})$, where x is the flattened feature map from the previous layer, W_{15}

and b_{15} are the weight and bias parameters for the final layer, respectively, and y represents the class scores for the input image [24].

One of the main strengths of the VGG-16 architecture is its ability to achieve very good performance on image classification tasks, even when trained on a small dataset [25]. This is due to the use of very small filters and very deep architectures, which help to improve the representation of the input image. Additionally, the architecture is relatively simple and easy to implement, making it a popular choice for a wide range of image classification and computer vision tasks.

The VGG-16 model is a pre-trained model that was trained on a large dataset of images called ImageNet dataset and it is available for download. It can be used as a starting point for other computer vision tasks and this process is known as transfer learning [26], [27]. The ImageNet dataset is a large dataset of labeled images, primarily used for training and evaluating computer vision models. It was first introduced in the paper "ImageNet: A Large-Scale Hierarchical Image Database" at Stanford university under the supervision of Fei-Fei Li, which was published in the Conference on Computer Vision and Pattern Recognition (CVPR) in 2009 [28]. The ImageNet dataset contains over 14 million images and more than 20,000 object categories, it's widely used in the computer vision community for different tasks such as image classification, object detection, and semantic segmentation. It's considered a benchmark for evaluating the performance of deep learning models on image classification tasks.

Transfer learning is a technique in machine learning where a model trained on one task is reused as the starting point for a model on a second task. This is done by taking the pre-trained weights of the model and using them as the initial weights for a new model, which is then trained on the new task. Transfer learning is useful because it can save a significant amount of time and computational resources compared to training a model from scratch. It can also improve the performance of the new model by leveraging the knowledge acquired by the pre-trained model [29].

Transfer learning is particularly useful in deep learning, where it is often difficult and time consuming to train a model from scratch due to the large number of parameters in the model. Instead, pre-trained models such as VGG-16, ResNet, Inception, etc can be used as a starting point and fine-tuned on the new dataset [30] In this paper, the VGG-16 model which has been pre-trained on the Imagenet dataset is being used as the base model, and the new model is being trained on the new dataset, this technique is called fine-tuning.

Although the original VGG model was trained on images that are 224x224 pixels in size, it is common to resize the input images to a smaller size, when using the VGG model for transfer learning [31]. This is because the VGG model has a large number of parameters and layers, which can make it computationally expensive to train, especially with small datasets or large image sizes. Resizing the images to a smaller size reduces the computational cost while still preserving most of the image information [32]. Additionally, this allows the model to be trained faster and makes it more feasible to run on a personal computer or mobile device [33]. In our proposed model, the input image is resized to 150x150.

IV. Methodology

The VGG-16 model is a convolutional neural network model that was trained on the ImageNet dataset, which contains more than 14 million images and 1000 classes. The model is pre-trained and can be used as a feature extractor for other image classification tasks. The architecture of the VGG-16 model consists of multiple convolutional layers, max-pooling layers, and dense layers. The convolutional layers are responsible for learning the features of the image, and the dense layers are responsible for classifying the image based on those features.

The original VGG model, as described in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" by Karen Simonyan and Andrew Zisserman, was trained on images that are 224x224 pixels in size. However, when using the VGG model for transfer learning, it is common to resize the input images to a different size, such as 150x150 in this case. The reason for this is that the VGG model has a large number of parameters and a large number of layers, which can make it computationally expensive to train, especially if the dataset is small or if the images are very large. By resizing the images to a smaller size, you can reduce the computational cost while still preserving most of the information in the images. Additionally, this allows the model to be trained faster, and makes it more feasible to be run on a personal computer or a mobile device.

In this work, the VGG-16 model is used as a feature extractor for the plant classification task. The output from the base VGG-16 model is flattened and fed into a dense layer with 1024 units and a dropout layer with a dropout rate of 0.5. Finally, a dense layer with 29 units and a softmax activation function is added to the model to classify the images into 29 different classes. The new model is then trained on the plant image dataset using the data generators and callbacks defined earlier. The methodology for plant classification model involves the following steps:

- Collecting the dataset which consists of plant images. In order to construct the dataset of plant images, it was necessary to utilize a diverse array of languages for the purpose of conducting internet searches. These languages include English, Spanish, French, German, Italian, Portuguese, Russian, Chinese (Simplified), Arabic, Bengali, Japanese, Turkish, Indonesian, Polish, Korean, Ukrainian, and Thai. The utilization of multiple languages enables us to access a vast array of images for each individual plant within the dataset. This allows for the expansion and enrichment of our image collection, resulting in a greater number of images for each individual plant.
- Preprocessing the dataset, to prepare the dataset for training the CNN model, several preprocessing steps were performed. Firstly, the images were resized to 150 x 150 pixels to match the input size of the VGG-16 model. Secondly, the images were normalized by subtracting the

mean pixel value from each pixel and divided by the standard deviation. Finally, data augmentation techniques such as random flipping, rotation, and zooming were applied to increase the diversity of the training dataset and prevent overfitting.
- Choosing the Model Architecture which is a VGG-16-based deep learning model consisting of 5 residual building blocks [8]. The VGG-16 model, a convolutional neural network (CNN), has been pre-trained on a large image dataset and is used as a feature extractor. The 5 residual building blocks are added to improve the performance of the model by allowing it to learn the residuals between the input and desired output, making it easier for the model to improve its accuracy. This has been achieved in three phases as follows.
- Training the model using the fit_generator method, specifying the generator for the training data, the number of steps per epoch, the generator for the validation data, the number of validation steps, the number of epochs, and the callbacks defined earlier.

The proposed model is stacking a VGG-16 model on top of a other dense layers illustrated in Figure 3, and compiles the model using the Adam optimizer and categorical crossentropy loss function. It defines some callbacks such as EarlyStopping, which stops the training if the validation loss doesn't decrease for a certain number of epochs and ModelCheckpoint, which saves the best model weights to a file. Finally, the training runs for a total of 80 epochs.

V. Dataset

The dataset used in this study consists of 25,686 images of 29 different medicinal plant species. The images were collected from various sources, including botanical gardens, herbaria, and the internet. The dataset was split into training and testing sets, with 80% of the images used for training and the remaining 20% for testing. A sample of medical images in the dataset is shown in Figure 2..

Fig. 2. Sample of the medical plants in the dataset.

VI. implementation

The proposed model consisted of a deep neural network using the VGG-16 architecture combined with extra layers on top. The VGG-16 architecture is being used as the base model, and additional layers are being added on top to further improve the performance of the model 3.

The VGG-16 model is a convolutional neural network (CNN) that is composed of multiple layers including convolutional, pooling, and fully connected layers. The added layers are a flatten layer, a fully connected layer with a ReLU activation and a dropout layer, and a final fully connected layer with softmax activation.

The mathematical representation of the architecture of the model is as follows:

The VGG-16 model:

- A series of convolutional layers with filters of size 3x3 and stride 1.
- A series of max pooling layers with size 2x2 and stride 2.
- The final layers of VGG-16 is composed of 3 fully connected layers with 4096 neurons each.

The added layers:

- Flatten layer: Reshape the output of the previous layer into a 1D tensor
- Fully connected layer: The output of the previous layer is multiplied by a weight matrix W and added with a bias vector b to give the output of this layer.
$$y = Wx + b, \tag{17}$$
where x is the input of the fully connected layer and y is the output of the fully connected layer.
- ReLU activation: The output of the previous layer is passed through a ReLU activation function to introduce non-linearity.
$$y = max(0, x). \tag{18}$$
- Dropout layer: The output of the previous layer is multiplied by a dropout mask, which is a binary mask with a probability of 0.5 of being set to 1.
- Fully connected layer (output layer): The output of the previous layer is multiplied by a weight matrix W and added with a bias vector b to give the output of this layer.
$$y = Wx + b, \tag{19}$$
where x is the input of the fully connected layer and y is the output of the fully connected layer.
- Softmax activation: The output of the previous layer is passed through a softmax activation function to produce a probability distribution over the classes.
$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}, \tag{20}$$
where y_i is the output of the softmax activation for class i, x_i is the input for class i, and n is the number of classes.
- The final output of this model is a probability distribution over the classes, and it is trained to minimize the

```python
# Define the base model
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(IMAGE_SIZE[0], IMAGE_SIZE[1], 3))

# Define the new model
model = Sequential()
model.add(base_model)
model.add(Flatten())
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

# Compile the model
model.compile(optimizer=Adam(lr=1e-5), loss='categorical_crossentropy', metrics=['accuracy'])
```

Fig. 3. Proposed Model Architecture in Python

categorical cross-entropy loss function using the Adam optimizer. The accuracy is used as a metric to evaluate the performance of the model.

VII. Results and Analysis

The proposed approach utilizes a convolutional neural network (CNN), specifically a VGG-16-based model, trained on a dataset of 25,686 images of 29 plant species. The VGG-16 model is pre-trained on ImageNet, a wide range of images, which makes it an excellent starting point for many image recognition tasks.

Figure 4 shows that the model's accuracy increased after training for 30 epochs, with a maximum accuracy of 0.98. This represents a considerable improvement from the starting accuracy. Figure 5 shows that the current loss after a total of 30 epochs is around 0.04, indicating that the model has the ability to learn from the data and enhance its performance over time.

Experimental results show that the proposed method outperforms traditional machine learning techniques in terms of classification accuracy, with a recognition rate of 98%. This approach can be used as a useful tool for healthcare professionals, researchers, and other professionals in the field of herbal medicine to aid in the identification of medicinal plants. It can be extended to other fields such as agriculture and botany, where accurate plant identification can have a significant impact on sustainable natural resource use and conservation efforts.

It is worth mentioning that due to limitations in CPU processing, the model was trained in two separate runs, each with 15 epochs. This underlines the importance of having sufficient computational resources when training complex models. The results presented in Figures 4 and 5 demonstrate the model's potential for further improvement through increased epochs or additional resources.

VIII. Conclusions and Future Work

This study has demonstrated the effectiveness of a deep learning-based method using a convolutional neural network (CNN) to accurately recognize medicinal plants. The proposed

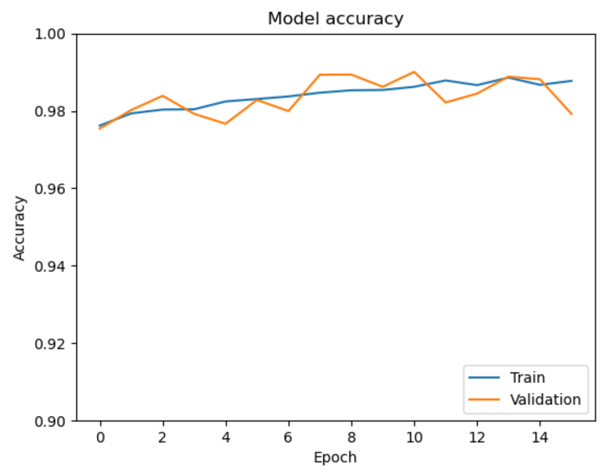

Fig. 4. Model accuracy. The results are obtained after training the model for an additional 15 epochs in addition to previous 15, due to limitations in CPU processing.

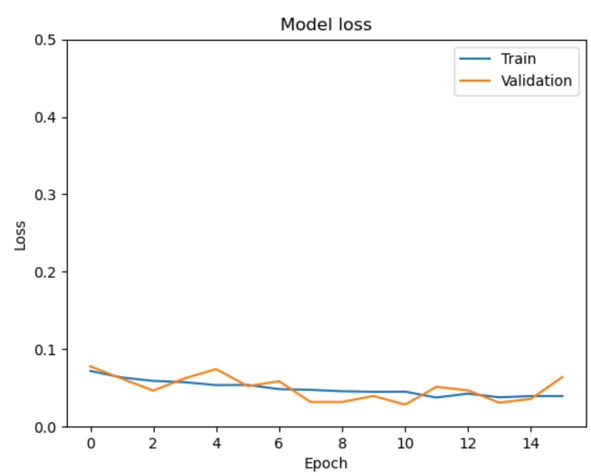

Fig. 5. Model loss. The results are obtained after training the model for an additional 15 epochs in addition to previous 15, due to limitations in CPU processing.

approach, based on the VGG-16 model, achieved a recognition rate of 98%, outperforming traditional machine learning techniques. The study's main contribution is its use of deep learning techniques to classify medicinal plants accurately, despite challenges posed by variations in appearance. The proposed CNN-based approach can learn and represent complex features in images, allowing it to classify medicinal plants with high accuracy. The study's results show that deep learning techniques can be used to classify medicinal plants accurately, providing a reliable tool for healthcare professionals and herbal medicine researchers to identify herbal plants. Future work could involve further integrating the identified plant with a large language model to provide additional information about the scientific name, more images, and explanations of the benefits and side effects of the plant, and instructions on how to use it in both English and the local language. Overall, the study supports the continued development and integration of deep learning in the recognition of herbal plants, which has the potential to significantly impact the field of herbal medicine research.

REFERENCES

[1] S. Malik, K. Kaur, S. Prasad, N. K. Jha, and V. Kumar, "A perspective review on medicinal plant resources for their antimutagenic potentials," *Environmental Science and Pollution Research*, vol. 29, no. 41, pp. 62014–62029, 2022.

[2] Y. O. Sharrab, M. Alsmira, Z. Dwekat, I. Alsmadi, A. Al-Khasawneh, *et al.*, "Performance comparison of several deep learning-based object detection algorithms utilizing thermal images," in *2021 Second International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pp. 16–22, IEEE, 2021.

[3] R. Azadnia, M. M. Al-Amidi, H. Mohammadi, M. A. Cifci, A. Daryab, and E. Cavallo, "An ai based approach for medicinal plant identification using deep cnn based on global average pooling," *Agronomy*, vol. 12, no. 11, p. 2723, 2022.

[4] T. Bai, H. Guan, S. Wang, Y. Wang, and L. Huang, "Traditional chinese medicine entity relation extraction based on cnn with segment attention," *Neural Computing and Applications*, vol. 34, no. 4, pp. 2739–2748, 2022.

[5] M. Alsmirat, Y. Sharrab, M. Tarawneh, S. Al-shboul, and N. Sarhan, "Video coding deep learning-based modeling for long life video streaming over next network generation," *Cluster Computing*, pp. 1–9, 2023.

[6] K. Amulya, K. Deepika, and P. Kamakshi, "An optimized hyper parameter-based cnn approach for predicting medicinal or non-medicinal leaves," *Advances in Engineering Software*, vol. 172, p. 103181, 2022.

[7] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, "Recent advances in image processing techniques for automated leaf pest and disease recognition–a review," *Information processing in agriculture*, vol. 8, no. 1, pp. 27–51, 2021.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[10] C. Li, W.-w. Jia, J.-l. Yang, C. Cheng, and O. E. Olaleye, "Multi-compound and drug-combination pharmacokinetic research on chinese herbal medicines," *Acta Pharmacologica Sinica*, vol. 43, no. 12, pp. 3080–3095, 2022.

[11] R. Jamdade, K. A. Mosa, A. El-Keblawy, K. Al Shaer, E. Al Harthi, M. Al Sallani, M. Al Jasmi, S. Gairola, H. Shabana, and T. Mahmoud, "Dna barcodes for accurate identification of selected medicinal plants (caryophyllales): Toward barcoding flowering plants of the united arab emirates," *Diversity*, vol. 14, no. 4, p. 262, 2022.

[12] M.-A. Crocq, "Historical and cultural aspects of man's relationship with addictive drugs," *Dialogues in clinical neuroscience*, 2022.

[13] M. S. Habibullah, B. H. Din, S.-H. Tan, and H. Zahid, "Impact of climate change on biodiversity loss: global evidence," *Environmental Science and Pollution Research*, vol. 29, no. 1, pp. 1073–1086, 2022.

[14] F. H. Arifah, A. E. Nugroho, A. Rohman, and W. Sujarwo, "A review of medicinal plants for the treatment of diabetes mellitus: The case of indonesia," *South African Journal of Botany*, vol. 149, pp. 537–558, 2022.

[15] A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agricultural Technology*, p. 100083, 2022.

[16] M. S. Kumar, D. Ganesh, A. V. Turukmane, U. Batta, and K. K. Sayyadliyakat, "Deep convolution neural network based solution for detecting plant diseases," *Journal of Pharmaceutical Negative Results*, pp. 464–471, 2022.

[17] D. Li, J. Hu, L. Zhang, L. Li, Q. Yin, J. Shi, H. Guo, Y. Zhang, and P. Zhuang, "Deep learning and machine intelligence: New computational modeling techniques for discovery of the combination rules and pharmacodynamic characteristics of traditional chinese medicine," *European Journal of Pharmacology*, p. 175260, 2022.

[18] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019.

[19] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, "Tensorizing neural networks," *Advances in neural information processing systems*, vol. 28, 2015.

[20] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.

[21] A. Boroumand, S. Ghose, Y. Kim, R. Ausavarungnirun, E. Shiu, R. Thakur, D. Kim, A. Kuusela, A. Knies, P. Ranganathan, *et al.*, "Google workloads for consumer devices: Mitigating data movement bottlenecks," in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 316–331, 2018.

[22] N. K. Chowdhury, M. M. Rahman, and M. A. Kabir, "Pdcovidnet: a parallel-dilated convolutional neural network architecture for detecting covid-19 from chest x-ray images," *Health information science and systems*, vol. 8, no. 1, p. 27, 2020.

[23] J. Jin, A. Dundar, and E. Culurciello, "Flattened convolutional neural networks for feedforward acceleration," *arXiv preprint arXiv:1412.5474*, 2014.

[24] M. A. Mahmoud, P. Guo, A. Fathy, and K. Li, "Srcnn-pil: Side road convolution neural network based on pseudoinverse learning algorithm," *Neural Processing Letters*, vol. 53, pp. 4225–4237, 2021.

[25] J.-H. Luo and J. Wu, "An entropy-based pruning method for cnn compression," *arXiv preprint arXiv:1706.05791*, 2017.

[26] V. Iglovikov and A. Shvets, "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation," *arXiv preprint arXiv:1801.05746*, 2018.

[27] Y. O. Sharrab, I. Alsmadi, and N. J. Sarhan, "Towards the availability of video communication in artificial intelligence-based computer vision systems utilizing a multi-objective function," *Cluster Computing*, vol. 25, no. 1, pp. 231–247, 2022.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, 2009.

[29] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, vol. 173, p. 105393, 2020.

[30] E. Ayan, H. Erbay, and F. Varçın, "Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks," *Computers and Electronics in Agriculture*, vol. 179, p. 105809, 2020.

[31] A. S. Paymode and V. B. Malode, "Transfer learning for multi-crop leaf disease image classification using convolutional neural network vgg," *Artificial Intelligence in Agriculture*, vol. 6, pp. 23–33, 2022.

[32] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, "Learning in the frequency domain," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1740–1749, 2020.

[33] R. Huang, J. Pedoeem, and C. Chen, "Yolo-lite: a real-time object detection algorithm optimized for non-gpu computers," in *2018 IEEE international conference on big data (big data)*, pp. 2503–2510, IEEE, 2018.