Leaf App: Leaf recognition with deep convolutional neural networks

T L I Sugata and C K Yang

Department of Information Management, National Taiwan University of Science and Technology, Taipei 106, Taiwan

E-mail: triluhur04@gmail.com

Abstract. In this paper, a very deep convolutional neural network is used to do leaf recognition. In order to predict location of leaves, some pre-processing technique is adopted to extract regions in the image before doing classification. To improve the accuracy, we enlarge the dataset by data augmentation, i.e., doing several transformations such as horizontal reflection, contrast enhancement and rotations. Experimental results show that by using deep convolutional neural network with data augmentation, our system can achieve accuracy close to the state-of-the-art systems.

1. Introduction

Plants are very important in our lives because they provide us oxygen which is needed by all living things. In order to identify variety of plants much work has been done, including in the field of computer vision. Extraction of leaf shapes, geometric features (diameter, physiological length, physiological width, etc.) and morphological features (aspect ratio, perimeter ratio, rectangularity, etc.) [1], [2] are usually what involved to automatically identify a leaf.

Convolutional neural network (convnet or CNN) has recently become popular because it can achieve state-of-the-art result in image classification by feeding the network with a huge number of images. Alex Krizhevsky et al. [3] used five convolutional, three fully-connected layers and utilized drop out in ImageNet [5] dataset to achieve an error rate of 16.4%. Zeiler et al [4] proposed a technique to visualize convnet by using a deconvolutional network (deconvnet). They visualize AlexNet [3] to examine the convnet and do several improvements to achieve better results.

In this paper, we make the following contributions. First, we proposed a mobile leaf recognition system using a well-known deep convolutional neural network called VGG-16 architecture to achieve a better result without extracting leaf features such as shape, diameter, aspect ratio, etc. from images. Second, we extract the region of images to let the system recognize multiple and overlapped leaves together with their bounding box. Figure 1 shows the overall process of the system.

The rest of the paper is organized as follows. In section 2, we review some of the previous work. In section 3, we describe the pre-processing technique we have done before doing the classification. Section 4 describes the network architecture. Section 5 presents the experimental result. Section 6 will be the discussion and conclusion of the paper.

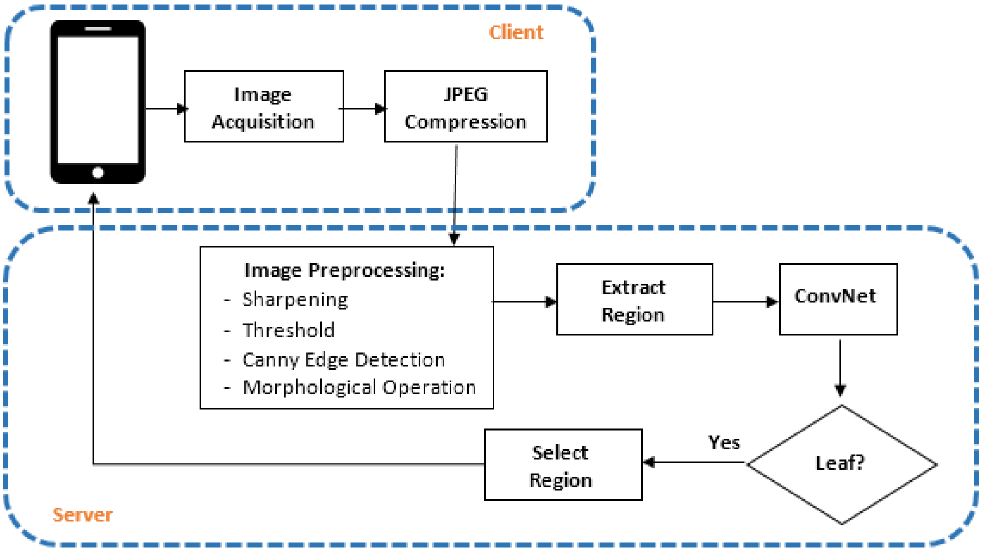

Figure 1. The overall process flow.

2. Related Works

Leaf Recognition. In recent years, many researches have been conducted on leaf recognition. Kumar et.al [11] proposed mobile app for recognize 184 kind of trees by extracting the curvature features. Wang et al. [12] use Pulse-coupled neural network (PCNN) to extract leaf features. They achieve accuracy above 90% in three different datasets. While Hu et al. [13] applies Multiscale Distance Matrix to get geometric structure of the leaf shape.

Deep Learning for Leaf Recognition. Recently, several papers start to utilize CNN to do the leaf recognition [9][10]. Data augmentation was added [9] in order to improve the accuracy. Lee et al. [10] used the AlexNet [3] model to identify leaf and visualize its feature to analyze which features are important for the leaf identification.

3. Image Pre-Processing

This process is aiming at extracting regions in the image. Because the input image from a mobile phone may not contain a single leaf. So applying some simple image preprocessing techniques on the image is needed, especially in the condition where a leaf is overlapped with each other.

3.1. Image Sharpening

For edge detection is it necessary to sharpen the image, so that edges can be fully retrieved. Image sharpening is achieved by convolving the following kernel K with the input image:

$$\mathrm{K} = \frac{1}{8}\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & 2 & 8 & 2 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix} \tag{1}$$

3.2. Thresholding

To generate a binary image, a black pixel is used to indicate the foreground and white pixel the background, and the Otsu method [6] is applied to an image that is made grayscale beforehand. Otsu method will automatically select threshold for image segmentation by utilizing the 0-th and the first-order cumulative moment of the gray-level histogram.

3.3. *Edge Detection*

The well-known edge detector Canny edge detection is used [7]. This algorithm is widely adopted because it can have good detection results. The canny algorithm itself runs by the following steps:

a. Smoothen an image using a Gaussian filter to remove the noise
b. Calculate the image gradient
c. Apply non-maximal suppression to filter out response
d. Apply double thresholds to remove edge pixels with a weak gradient
e. Use hysteresis to track the edge which is not extracted from noise

3.4. *Morphological Operation*

Morphological operators are applied to images to remove the noise inside the image. Mathematical morphology operators consist of four basic operations which are opening, closing, dilation and erosion. The following are opening (2) and closing (3) operators:

$$\square\square\square\circ\circ I_{\circ}\ H = (\square I \ominus H\square)\oplus H\square \tag{2}$$

$$\square I \bullet H\square = (\square I\oplus H\square) \ominus H\square \tag{3}$$

And the following are the dilation (4) and erosion (5) operators:

$$I \oplus H = \max_{(i,j)\in H}\{I(u+i,v+j)+H(i,j)\} \tag{4}$$

$$I \ominus H = \min_{(i,j)\in H}\{I(u+i,v+j)+H(i,j)\} \tag{5}$$

3.5. *Image Segmentation*

For extracting individual leaves in images that contain overlapping leaves or multiple leaves, using a thresholding image is not enough because deep convnet will detect it as a “single-leaf” or “non-leaf” image. So the following steps shown in Figure 2 are used.

a. Get the threshold map using Otsu method
b. Get the edge map using Canny edge
c. Sum up the threshold map and edge map
d. Invert the map
e. Remove noise/small holes by using contour hierarchy

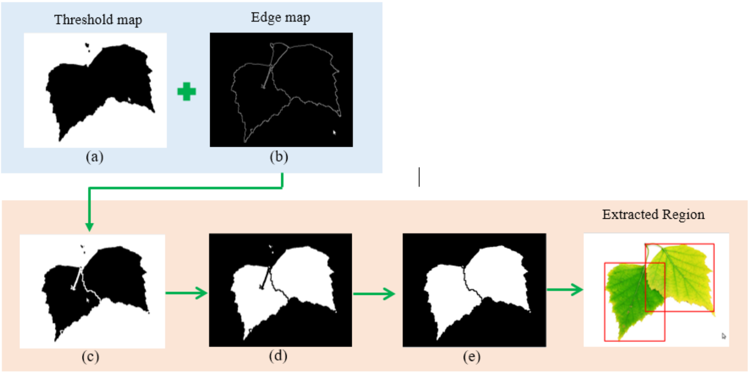

Figure 2. Overview of the segmentation approach.

4. Network Architecture

The deep convolutional neural network model used in this paper is VGG16 [8]. VGG16 as shown in **Fig. 3** consists of 16 weight layers 13 convolutional layers and 3 fully connected layers with 138million total number of parameters and takes 224 x 224 pixels RGB image as the input.

Fine-tuning. In deep convnet, we simply perform a fine tuning in the pre-trained model from ImageNet-1K dataset into Flavia dataset to classify 32 leaf classes. The fine tuning is chosen because to train the dataset from scratch will be time-consuming and the weights learned from ImageNet-1K pre-train model can be transferred to VGG16 with Flavia dataset.

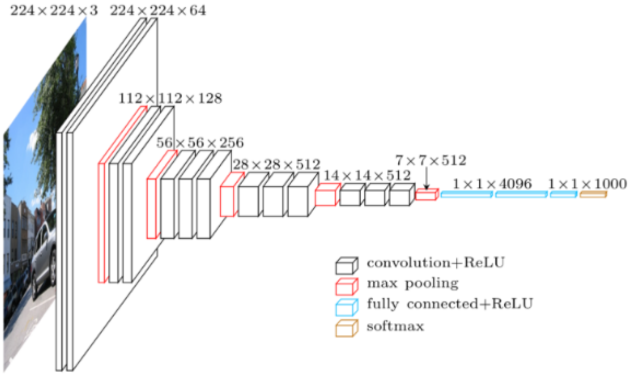

Figure 3. VGG16 architecture [16]

5. Experimental Result

Experiments were conducted on NVIDIA GeForce GTX 1080 8GB GPU using an open source framework developed by Berkeley Vision called Caffe [9] run as a server. The model training is set with an initial learning rate of 0.001, batch size of 25 examples, weight decay of 0.0005, momentum 0.9 and we use a stochastic gradient descent (SGD) as the solver type.

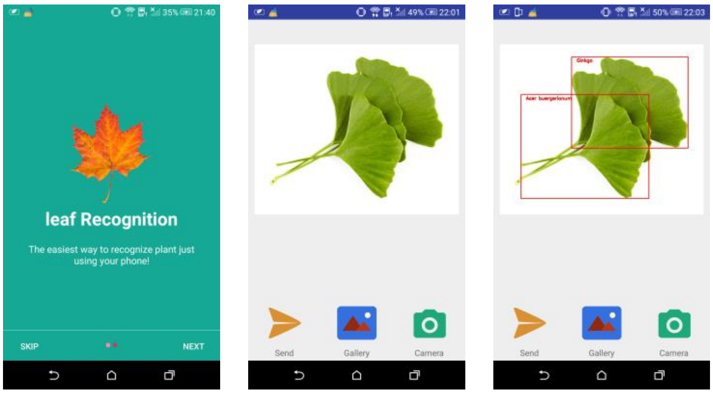

Figure 4. User interface of leaf app.

5.1. Dataset

We train the network using Flavia dataset [2], which is a popular dataset for leaf recognition. It is publicly available and consists of 32 plants classes. Each image contains a single leaf and the number of images in the dataset is 1907. To make it work with previous model, we rescaled all image dimension to 224x224.

5.2. Data Augmentation

Since we are using a deep network, we need to have a huge number of training images to get the better results and also to reduce the overfitting effect. Because the Flavia dataset only contains 1907 images, we enlarge the dataset by doing several transformations: horizontal reflection, contrast enhancement and rotations (by rotating the image with the interval 10 degrees).

5.3. Comparative Results

Table 1 shows the accuracy comparison when different datasets are used. Images are selected randomly for testing. Our system shows that by using the initial images in the Flavia dataset which contains 1907 images, it can already achieve the accuracy of 99.7%. While using a bigger dataset (initial image is expanded twenty times through data augmentation) [9], the accuracy improves to 99.9%, as shown in Figure 5.

Table 1. Accuracy in dataset.

Test set	Accuracy (%)
Initial dataset	99.7
Initial + data augmentation	99.9

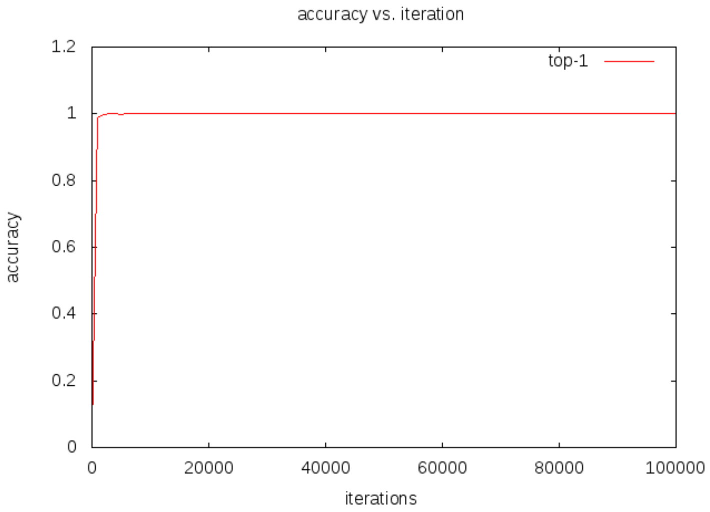

Figure 5. Accuracy according to iterations.

In Table 2, we can see that by using deep convnet, our system can achieve accuracy 99.9%. It means that deep convnet can outperform the results from other schemes. By using convnet, leaf recognition can be done automatically. Extracting the leaf shape, diameter, aspect ratio, etc. is no longer needed.

Table 2. Comparison with other schemes

Scheme	Accuracy (%)
PNN in [2]	90
PCNN in [12]	96.67
MLNN in [15]	94
CNN in [9]	94.6
Deep CNN in [10]	99.6
VGG+data augmentation	99.9

6. Discussion and Conclusion

This paper presents deep convolutional neural network for leaf recognition. We develop a mobile application, where a user can capture an image from the camera or load an image from the phone's gallery, send it to the server and the server will return the class which has the highest score along with its locations. From the experimental results, it can be shown that our system can achieve a better result when compared with the other methods. However, to be able to extract the region and locate the leaf image it is highly dependent on the image conditions (e.g., shadow, illumination, etc.). For future works, we will develop a system that can recognize leaf image under more general conditions.

Acknowledgment

This work was supported in part by the Ministry of Science and Technology of Taiwan under the grants: MOST 104-2221-E-011-083-MY2, MOST 105-2218-E-011-005, MOST 105-2218-E-001-001 and MOST 106-3114-E-011-003.

References

[1] Zhao C, Chan S S F, Cham W-K and Chu L M 2015 Plant identification using leaf shapes—A pattern counting approach *Pattern Recogn.* pp 3203–15

[2] Wu S G, Bao F S, Xu E Y, Wang Y-X, Chang Y-F and Xiang Q-L 2007 A leaf recognition algorithm for plant classification using probabilistic neural network *IEEE Int. Symp. on Signal Processing and Information Technology* pp 11–16

[3] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks *NIPS* **25** pp 1097–1105

[4] Zeiler M D and Fergus R 2014 Visualizing and understanding convolutional networks *Computer Vision – ECCV 2014*, *Lect. Notes Comp. Sc.* **8689** eds Fleet D, Pajdla T, Schiele B and Tuytelaars T (Cham: Springer) pp 818–833

[5] Russakovsky O, Deng J and Su H *et al* 2015 ImageNet Large Scale Visual Recognition Challenge *Int. J. Comput. Vision* **115** pp 211–252

[6] Otsu N 1979 A threshold selection method from gray level histogram *IEEE T. Syst. Man Cyb.* **9** pp 62–66

[7] Canny J 1986 A computational approach to edge detection *IEEE T. Pattern Anal.* **PAMI-8** pp 679–698

[8] Simonyan K and Zisserman A 2015 Very deep convolutional networks for large-scale image recognition *Int. Conf. on Learning Representations*

[9] Zhang C, Zhou P, Li C and Liu L 2015 A convolutional neural network for leaves recognition using data augmentation *IEEE Int. Conf. on Computer and Information Technology*

[10] Lee S H, Chan C S, Wilkin P and Remagnino P 2015 Deep-plant: Plant identification with convolutional neural networks *IEEE Int. Conf. on Image Processing (ICIP)* pp 452–456

[11] Kumar N, Belhumeur P, Biswas A, Jacobs D W, Kress W J, Lopez I C and Soares J V B 2012 Leafsnap: A computer vision system for automatic plant species identification *ECCV* pp 502–516

[12] Wang Z, Sun X, Zhang Y, Ying Z and Ma Y 2016 Leaf recognition based on PCNN *Neural Comput. & Applic.* **27** pp 899–908

[13] Hu R, Jia W, Ling H and Huang D 2012 Multiscale distance matrix for fast plant leaf recognition *IEEE T. Image Process.* **21** pp 4667–72

[14] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S and Darrell T 2014 Caffe: Convolutional architecture for fast feature embedding *arXiv preprint arXiv: 1408.5093*

[15] Du J, Huang D, Wang X and Gu X 2005 Shape recognition based on radial basis probabilistic neural network and application to plant species identification *Advances in Neural Networks – ISNN 2005*, *Lect. Notes Comp. Sc.* **3497** eds Wang J, Liao X F and Yi Z (Berlin, Heidelberg: Springer) pp 281–285

[16] https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/