Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 17, No. 1, March 2017, pp. 26-34
http://dx.doi.org/10.5391/IJFIS.2017.17.1.26

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Plant Leaf Recognition Using a Convolution Neural Network

Wang-Su Jeon[1] and Sang-Yong Rhee[2]
[1]Department of IT Convergence Engineering, Kyungnam University, Changwon, Korea
[2]Department of Computer Engineering, Kyungnam University, Changwon, Korea

Ijfis

Abstract

There are hundreds of kinds of trees in the natural ecosystem, and it can be very difficult to distinguish between them. Botanists and those who study plants however, are able to identify the type of tree at a glance by using the characteristics of the leaf. Machine learning is used to automatically classify leaf types. Studied extensively in 2012, this is a rapidly growing field based on deep learning. Deep learning is itself a self-learning technique used on large amounts of data, and recent developments in hardware and big data have made this technique more practical. We propose a method to classify leaves using the CNN model, which is often used when applying deep learning to image processing.

Keywords: Leaf, Classification, Visual system, CNN, GoogleNet

1. Introduction

When leaving a town and entering the suburbs, we may encounter many kinds of trees. We may be able to identify those trees that often grow on urban streets, however most of the trees and plants found in city suburbs will be unknown to the majority of us. There are approximately 100,000 species of trees on earth, which account for about 25% of all plants. Many of the trees are in tropical regions, and because only limited botanical research has been carried out in these areas, it is believed that there are many undiscovered species [1]. It is clear that identifying large numbers of such trees is a complex process.

An example of the complexity of tree identification can be seen with plums and apricots. These are very similar in leaf shape, the shape of the tree, and even in the shape of the young fruit. The flower shape is also very similar, and the tree type can only be identified by determining whether the calyx is attached, or inverted relative to the petal. Additionally, some trees are not easily distinguishable except at particular times; for example, when they bloom or bear fruit. To identify trees like these, considerable information is required, including leaf shape, shape of the leaves that are directly attached to branches, branch shape, shape of the whole tree, tree size, flower shape, flowering time, and fruit.

When using branches of biology such as cell biology, molecular biology, phytochemistry, or morphologic anatomy, it may be possible to distinguish plants without time constraints. However, it is unrealistic for the general public to identify the names of trees or plants using these methods when, for example, they are walking in a woodland.

Since the public will usually distinguish trees by their appearance, studies have been carried out to recognize trees using this method alone. Du et al. [1] have converted a color input image

Received: Feb. 1, 2017
Revised : Mar. 23, 2017
Accepted: Mar. 24, 2017

Correspondence to: Sang-Yong Rhee
(syrhee@kyungnam.ac.kr)

into a binarized image to extract the outline, and the two dimensional features were then extracted using the outline image. These features were grouped using the Move Median Centers (MMC) classifier. This study showed faster execution speeds than those of previous studies, and generated accurate results using the combination of the characteristics. However, the recognition rate was only approximately 90%.

Nam and his colleague [2, 3] used a shape-based search method to distinguish plants. They tried to improve the accuracy by using not only the outline, but also the vein data of the leaves. The outline recognition was improved using the minimum perimeter polygons (MPP) algorithm, and the vein data were represented using extended curvature scale space (CSS) to extract the midrib, intersections and endpoints. A weighted graph was composed using the relationship between the feature points and the distance value, and the similarity was calculated using the value. A total of 1,032 plant leaf images were used from plant inscriptions, however, the exact recognition rate was not given because the research related to database searches rather than to plant identification. However, a result graph showed better search results than for the existing study.

Because recognition is a considerable challenge in the field of computer vision, further development using ImageNet Large Scale Visual Recognition Competition (ILSVRC), has been attempted; however, until 2011, the top five error rates of the most effective recognition technique were 25.8%. With the emergence of AlexNet [4] in 2012, the error rate has dropped sharply to 16.4%, and this dropped further to 2.99% at 2016. This is the result of improved performance when compared to traditional machine learning methods, which classify data after extracting features or preprocessing. In this paper, we study a method for learning and recognizing types of leaves using the convolution neural network (CNN) model, which is a deep learning technology.

The system proposed in this paper is constructed as shown in Figure 1. The method proposes to improve classification performance by using a CNN that extracts and learns feature points.

In Section 2, we examine existing leaf recognition research. In Section 3, we describe GoogleNet, a CNN that imitates human visual systems. Section 4 explains the leaf recognition system, while Section 5 describes the experiment and analyzes the results. Section 6 concludes the paper.

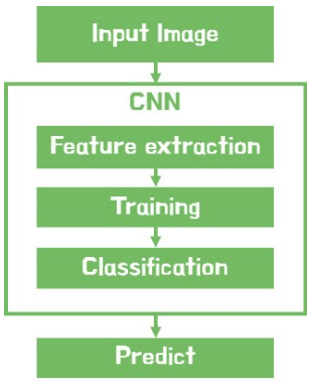

Figure 1. System composition.

2. Method of Leaf Recognition

2.1 Feature Extraction

In previous studies, the leaf color, contour, texture, and shape were used to classify plants. As shown in Figure 2, the color image was transformed into a grayscale image by applying Eq. (1), the grayscale image was then converted to a binary one through binarization, and the contour then extracted. The features are extracted using the characteristics of the contour line [5]. Using these features, the recognition rate was 90% when classified through machine learning. Because the shape of the leaf outlines are similar to each other, the features alone make it difficult to classify the plant.

$$Grau = 0.299 \times I_r + 0.587 \times I_0 + 0.114 \times I. \qquad (1)$$

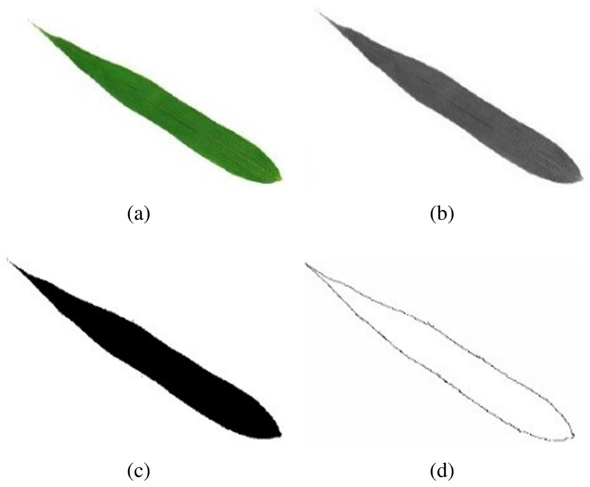

Figure 2. Example of leaf contour extraction.
(a) Input image, (b) gray scale image,
(c) binary image, and (d) contour extraction.

In addition, brightness or shape transformation features may be used with cumulative histogram operations. Typical methods are Histogram of Oriented Gradients (HOG) [6] and Scale-Invariant Feature Transform (SIFT) [7].

Disadvantages of these feature extraction algorithms are firstly that computation levels are high, and secondly that generalization is difficult due to the dependency on specific data.

2.2 Machine Learning

Machine learning is a method that classifies sample data after learning to use feature points. For better performance, generalization of the data is required. Typical machine learning models are AdaBoost [8] and support vector machine (SVM) [9], and the performance of these methods depends on the input feature points. The primary disadvantage of existing machine learning methods is that they cannot extract the optimized feature points, because the learning and classification processes are performed independently.

3. Visual System and Neural Network Structure

3.1 Visual System in Humans

Neural networks mimics the human visual processing neural structure, as shown in Figure 3 [10]. In the retina, the portions of an object that have the strongest difference in the intensity of reflected light are recognized as the edges of the object, and the result is sent to the lateral geniculate nucleus (LGN). The LGN neuron compresses the entire shape around the corners of the object and sends it to the primary visual cortex (V1). The V1 neuron then recognizes the corners, contour, and direction of motion of the object. It also recognizes the difference between the images reflected in the retina of the left and right eyes as distances, and the result is sent to the secondary visual cortex (V2). The V2 neurons recognize the overall shape of the object and the color difference between each part, and send this to the tertiary visual cortex (V3). The V3 neurons recognize the color of the whole object, and the overall shape and color of the object are recognized at the lateral occipital cortex (LOC). As shown in Figure 3, the CNN is the neural network model that implements functions closest to the human visual structure. The first CNN model was designed by Yann LeCun in 1998. Called LeNet [11], it was derived from an experiment using the optic nerve of a cat brain, and showed that the neurons did not react at the same time when a picture was displayed; instead, only some neurons responded.

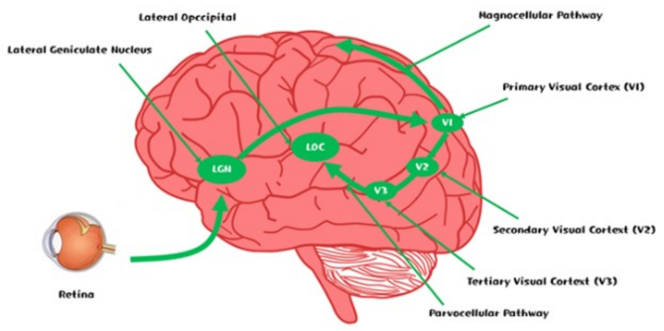
Lateral Occipital
Lateral Geniculate Nucleus
Magnocellular Pathway
Primary Visual Cortex (V1)
V1
LOC
LGN
V2
V3
Secondary Visual Cortext (V2)
Tertiary Visual Cortext (V3)
Retina
Parvocellular Pathway

Figure 3. Human visual system structure.

In a CNN, the convolution and pooling layers replicate the LGN to V3 paths in the visual system structure, and extract feature points from the image. The fully connected layer acts in the same way as the LOC in a human visual system to recognize the image.

As shown in Figure 4, the CNN structure extracts features by performing the convolution operation on the input image, extracts the maximum or average feature values on the pooling layer, and then classifies them in the fully connected layer.

3.2 Structure of GoogleNet

The CNN model used in this study is GoogleNet. With the advent of this model, researchers have developed deepened network structures that do not increase computational complexity. GoogleNet uses inception modules that use multiple convolutions in parallel, to extract various feature points. As shown in Figure 5, in the inception module a 1×1 convolution functions as a cascade. If using a 1×1 and 3×3 convolution, or a 1×1 and 5×5 convolution, it is possible to reduce the number of parameters and deepen the network [12].

To reduce the number of parameters, a team at Oxford University conducted a deep network study and developed the VGGNet model [13]. This model factorizes the convolutional filter, which means that a deep network using a several small layers is constructed.

Factorizing convolution can reduce the parameters by about 30%, by dividing the 5×5 filter into 3×3 and 3×3, as shown in Figure 6; this can also effectively extract feature points.

As shown in Figure 7, the GoogleNet model consists of a deep network with 22 layers of inception modules, with softmax functions used last. The vanishing gradient problem is an issue caused by the deepening of the network, and may lead to slow learning or overfitting. To avoid the overfitting prob-

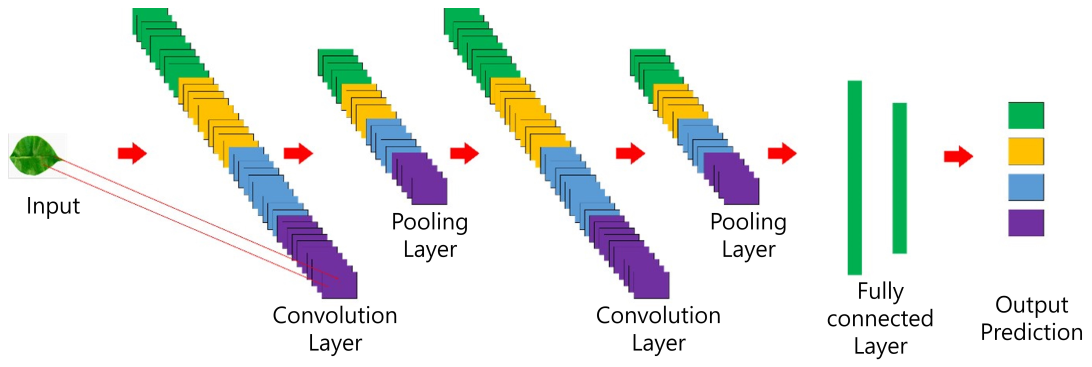

Figure 4. Basic structure of a convolution neural network.

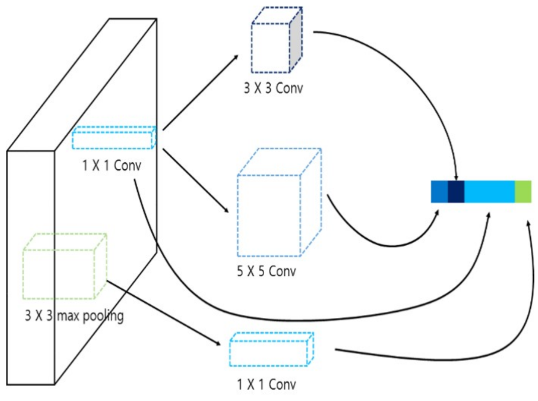

Figure 5. Inception module structure.

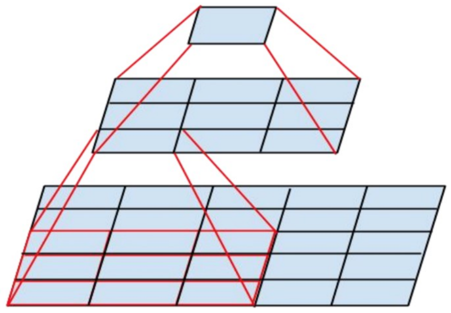

Figure 6. Factorizing convolution used in the VGGNet model.

lem, GoogleNet suggested auxiliary classifiers named ‘super vision’ [14, 15], as shown in Figure 7. The vanishing gradient problem is solved by storing the optimal values and adding the results of the auxiliary classifier using the backpropagation algorithm. This can result in stable learning results. At the end of the learning process, the auxiliary classifiers disappear and are not used at the test stage.

GoogleNet uses batch normalization instead of dropout. As shown in Figure 8, batch normalization is a method by which, by adding the batch normalization layer, the results generated before the layer is modified are used as the input to the activation function. In the batch normalization layer, Eqs. (2) and (3) are used at each layer to obtain the mean and variance [16]. Using the obtained mean and variance, the input is normalized as shown in Eq. (4). The denominator of Eq. (4) is the sum of the variance, and the constant and numerator are normalized by dividing the input value minus the mean.

$$\mu_B = \frac{1}{m_i}\sum_{i=1}^{m} x_i, \tag{2}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i = \mu_B)^2, \tag{3}$$

$$\hat{x}_i = \frac{x_i - \mu_\beta}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{4}$$

$$BN(x_i) = \gamma\left(\frac{x_i - \mu_\beta}{\sqrt{\sigma_B^2 + \epsilon}}\right) + \beta. \tag{5}$$

The nonlinearity can be obtained by multiplying and adding the scale factor and the shift factor to the normalized value, as shown in Eq. (5). Batch normalization solves the overfitting problem by normalizing the inputs to each layer, which allows the learning speed to be fast and achieves regularization.

4. Method of Leaf Recognition Using CNN

4.1 Image Cropping

Image cropping reduces the amount of computation used by the GPU to minimize the foreground portion. Figure 9(a) shows the input image used for learning, Figure 9(b) shows the result of cropping using the input image, and Figure 9(c) shows an image obtained by resizing the cropped image to 229 $\times$ 229 pixel. The adjusted images were used as experimental images.

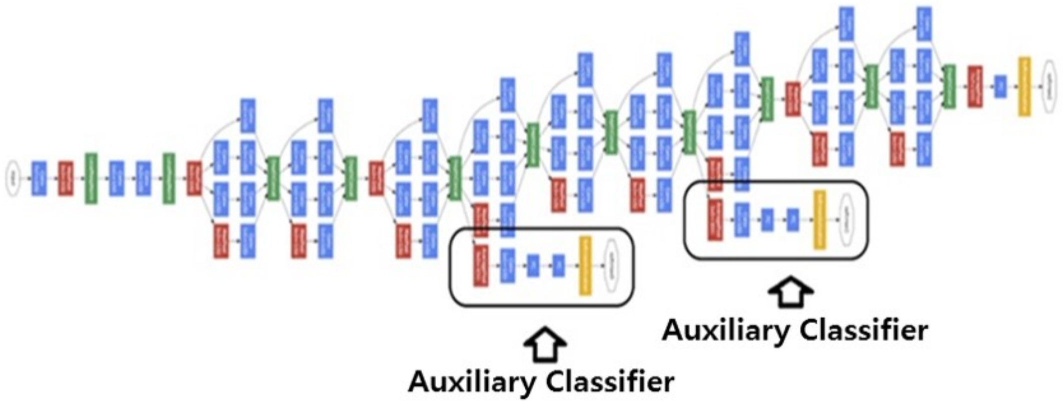

Figure 7. GoogleNet structure and auxiliary classifier units.

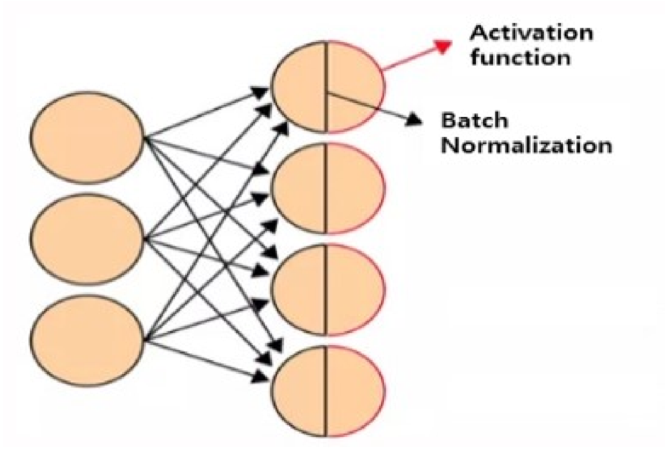

Figure 8. Batch normalization method.

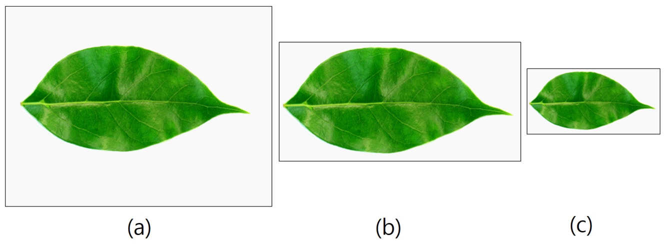

Figure 9. Leaf image cropping and resize example. (a) Input image, (b) cropping image, (c) 229 × 229 image.

4.2 Multi-scale Technique

Multi-scale is a learning process that randomly deforms several sizes by using the minimum and maximum sizes, as shown in Figure 10. By using this method, it is possible to prevent the overfitting phenomena arising as a result of less learning data.

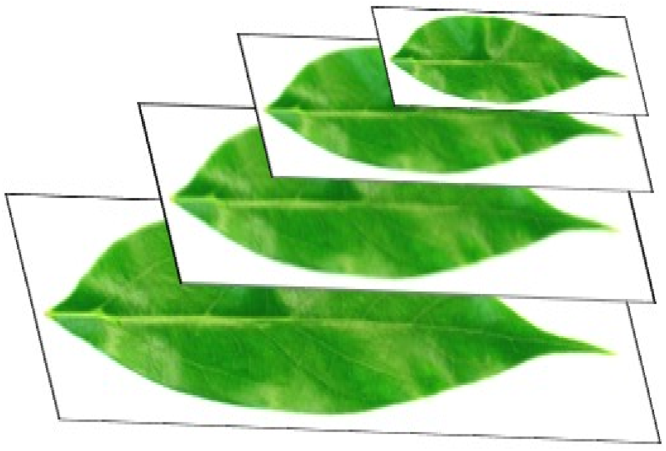

Figure 10. Multi-scale image.

4.3 Learning Using the CNN Model

For leaf recognition, a basic and modified structure of the GoogleNet model are used. The basic structure is as shown in Table 1, and the structure of the inception module used is shown in Figure 11. The inceptive module shown in Figure 11 adopts the factorizing convolution method in the incidence module, and the convolution that is a conventional linear filter has a nonlinear structure. The modified GoogleNet model is based on the structure type shown in Table 2, and on the two additional modules shown in Figure 10(a).

The model in Table 1 does not initially include the inception module. To begin with, the size of the input image is adjusted to 229 × 229, and a 3 × 3 stride 2, and 3 × 3 stride 1, of the convolution is performed. A 3 × 3 padded convolution operation is also performed to reduce data loss before pooling. After pooling, a 3 × 3 stride 1, 3 × 3 stride 2, and 3 × 3 stride

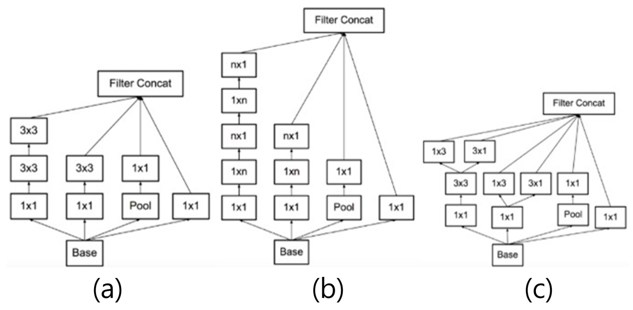

Figure 11. Factorizing convolution applied in the inception module.

Table 1. GoogleNet basic structure [Model 1]

Type	Filter size / stride	Input size
Conv	3×3 / 2	222×229
Conv	3×3 / 1	$149 \times 149 \times 32$
Conv padded	3×3 / 1	$147 \times 147 \times 32$
Ppool	3×3 / 2	$147 \times 147 \times 64$
Conv	3×3 / 1	$73 \times 73 \times 64$
Conv	3×3 / 2	$71 \times 71 \times 80$
Conv	3×3 / 1	$35 \times 35 \times 192$
$3 \times$ Inception	Figure 10(a)	$35 \times 35 \times 288$
$5 \times$ Inception	Figure 10(b)	$17 \times 17 \times 768$
$2 \times$ Inception	Figure 10(c)	$8 \times 8 \times 1280$
Pool	8×8	$8 \times 8 \times 2048$
Linear	Logits	$1 \times 1 \times 2048$
Softmax	Classifier	$1 \times 1 \times 1000$

Conv: convolution.

1 convolution are executed. After passing through the three inception modules shown in Figure 11(a), the five inception modules shown in Figure 11(b), and the two inception modules shown in Figure 11(c), an 8×8 pooling operation is processed. The effect of linear reduction using linear activation can be seen. The softmax classifier is used in the last stage. The model in Table 2 was used in the experiment described in Table 1, with the two additional incidence modules in Figure 11(a).

5. Experiment and Result Analysis

5.1 Experimental Environment

This paper uses the leaf sample data from the Flavia dataset [17], and the common leaf types shown in Figure 12. As shown in Table 3, the eight leaf types are lanceolate, light oval, acupuncture, linear, long oval, elongated, heart, and long leaf. The details of each type are shown in Figure 13. The training images were divided into eight types of 3,767 images, and the test images were chosen by randomly selected 100 images as like Figure 12(b).

We constructed the following experimental environment for learning and testing. The operating system used was Linux CentOS 7.0, and the CPU an Intel i7-6770k. The main memory size was 32 GB, and two parallel processing boards were used with an NVIDIA Maxwell TITAN graphics card. The deep learning framework used was TensorFlow r0.10 .

5.2 Experiment Method

Two CNN models were selected and tested. The chosen two models were GoogleNet and a variant of GoogleNet, and changes in performance were checked when the layers were added. The size of the each image used in the experiment was adjusted from 1600×1200 to 229×229 to fit the model. We also tested color changing or deforming of leaves by creating leaves that were cut or pitted randomly, as is common in nature. The leaf images used in the test are shown in Figures 14 and 15.

Table 2. Modified GoogleNet structure [Model 2]

Type	Filter size / stride	Input size
Conv	3×3 / 2	222×229
Conv	3×3 / 1	$149 \times 149 \times 32$
Conv padded	3×3 / 1	$147 \times 147 \times 32$
Pool	3×3 / 2	$147 \times 147 \times 64$
Conv	3×3 / 1	$73 \times 73 \times 64$
Conv	3×3 / 2	$71 \times 71 \times 80$
Conv	3×3 / 1	$35 \times 35 \times 192$
$5 \times$ Inception	Figure 10(a)	$35 \times 35 \times 288$
$5 \times$ Inception	Figure 10(b)	$17 \times 17 \times 768$
$2 \times$ Inception	Figure 10(c)	$8 \times 8 \times 1280$
Pool	8×8	$8 \times 8 \times 2048$
Linear	Logits	$1 \times 1 \times 2048$
Softmax	Classifier	$1 \times 1 \times 1000$

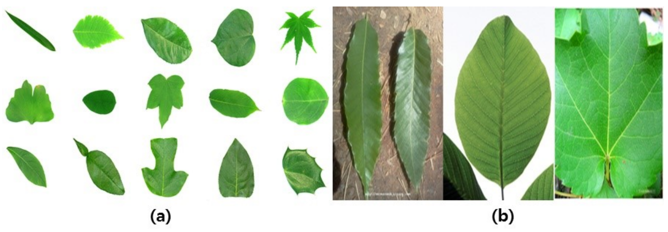

Figure 12. (a) Flavia image dataset and (b) natural leaves.

Table 3. Type and number of leaves

Leaf type	Number of images
Lanceolate	568
Oval	554
Acicular	612
Linear	439
Oblong	374
Reniform, kidney-shaped	580
Cordate, heart-shaped	379
Palmate leaf	361

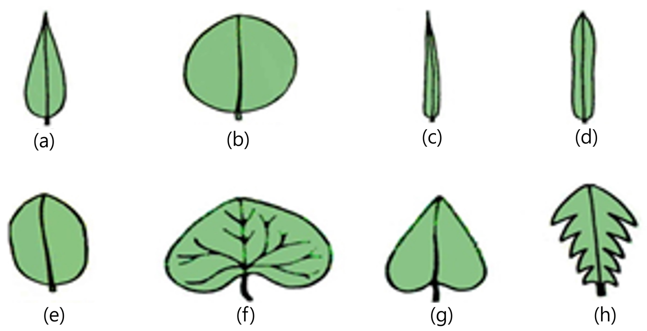

Figure 13. Leaf shapes: (a) lanceolate, (b) oval, (c) acicular, (d) linear, (e) reniform, (f) kidney-shaped, (g) cordate, heart-shaped, and (h) palmate leaf.

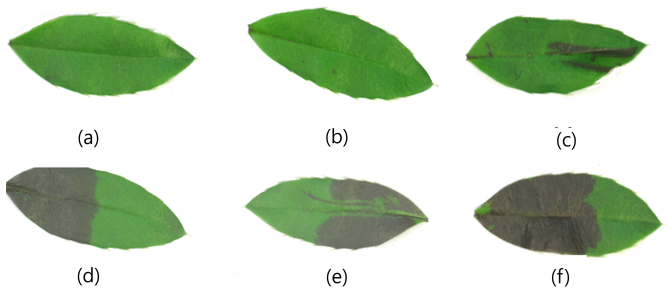

Figure 14. Color change: (a) input image, (b) discoloration 5%, (c) discoloration 10%, (d) discoloration 30%, (e) discoloration 50%, and (f) discoloration 60%.

Figure 14 shows the discoloration ratio of the input leaf images. Figure 15 shows images of damaged leaves.

The images in the Flavia dataset are displayed vertically, horizontally, and at an angle of 45°, which are all angles not necessarily found in nature. We therefore examined all possible leaf directions by rotating them by 90°. Using the methods described above, 10,000 training sessions were conducted and the performance of the two models was compared.

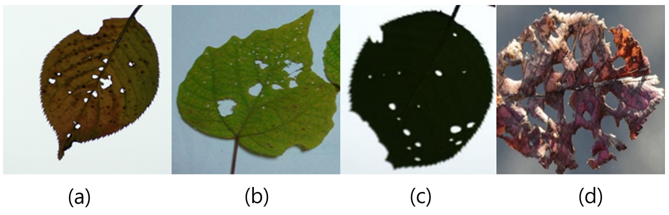

Figure 15. Leaf damage: (a) damage 5%, (b) damage 10%, (c) damage 15%, and (d) damage 30%.

Table 4. Model performance evaluation

	Model 1	Model 2
Image size	229 × 229	229 × 229
Training time	8h 43m	9h 18m
Accuracy	99.6%	99.8%

Table 5. Accuracy rate (%) in relation to discoloration

	Model 1	Model 2
Image in Figure 14(b)	99.5	99.65
Image in Figure 14(c)	99.2	99.3
Image in Figure 14(d)	98.8	99.1
Image in Figure 14(e)	98.5	98.9
Image in Figure 14(f)	98.2	98.6

5.3 Experiment Results

The two models described above were tested, and Model 2 demonstrated advantages over Model 1. The effect of increasing the number of inception modules in Model 2 to slightly increase performance, is shown in Table 2. However, as shown in Table 4, the difference between Model 1 and Model 2 is small. Experimental images were obtained by using the discolored images in Figure 14 and the distorted images in Figure 15, using different angles.

The discolored 100 images were prepared and tested as shown in Figure 14. Testing of the discolored images shows that the recognition rate degrades as the discoloration ratio of the leaves is increased; However, the ratio of degradation was not severe. Table 5 shows that Model 2 is slightly better than Model 1.

Table 6 shows that the recognition rate of Model 2 is slightly better than that of Model 1, even where with the leaf image

Table 6. Accuracy rate (%) in relation to damage

	Model 1	Model 2
Image in Figure 15(a)	97.4	98.4
Image in Figure 15(b)	96.8	98
Image in Figure 15(c)	96.2	97.6
Image in Figure 15(d)	94.4	95

contained 50 holes. According to the above results, the recognition rate of our system was above 94% when using the CNN, even when 30% of the leaf was damaged. Our system therefore improves upon previous studies, which achieved a recognition rate of approximately 90%.

6. Conclusion

In this paper, we proposed a new method to classify leaves using the CNN model, and created two models by adjusting the network depth using GoogleNet. We evaluated the performance of each model according to the discoloration of, or damage to, leaves. The recognition rate achieved was greater than 94%, even when 30% of the leaf was damaged.

In future research we will attempt to recognize leaves attached to branches, in order to develop a visual system that can replicate the method used by humans to identify plant types.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgements

This work was supported by the Ministry of Education (MOE) and the National Research Foundation of Korea (NRF), through the Human Resource Training Project for Regional Innovation (No. 2015-HICLA1035526).

References

[1] I. Friis and H. Balslev, "Plant diversity and complexity patterns: local, regional, and global dimensions," in *Proceedings of an International Symposium, Held at the Royal Danish Academy of Sciences and Letters*, Copenhagen, Denmark, 2003, pp. 25-28.

[2] Y. Nam and E. Hwang, "A representation and matching method for shape-based leaf image retrieval," *Journal of KIISE: Software and Applications*, vol. 32, no. 11, pp. 1013-1021, 2005.

[3] Y. Nam, J. Park, E. Hwang, and D. Kim, "Shape-based leaf image retrieval using venation feature," *Proceedings of 2006 Korea Computer Congress*, vol. 33, no. 1D, pp. 346-348, 2006.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Lake Tahoe, NV, 2012, pp. 1097-1105.

[5] J. X. Du, X. F. Wang, and G. J. Zhang, "Leaf shape based plant species recognition," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 883-893, 2007. http://dx.doi.org/10.1016/j.amc.2006.07.072

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 886-893. http://dx.doi.org/10.1109/CVPR.2005.177

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[8] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," *Machine Learning: Proceedings of the 13th International Conference*, vol. 96, pp. 148-156, 1996.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995. http://dx.doi.org/10.1007/BF00994018

[10] E. Chum, "Drones and artificial intelligence–AI," Available http://www.focus.kr/view.php?key=2016041200101856440

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 2002. http://dx.doi.org/10.1109/5.726791

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, 2015. http://dx.doi.org/10.1109/CVPR.2015.7298594

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. Available https://arxiv.org/cs/1409.1556

[14] L. Wang, C. Y. Lee, Z. Tu, and S. Lazebnik, "Training deeper convolutional networks with deep supervision," 2015. Available https://arxiv.org/cs/1505.02496

[15] C. Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," 2014. Available https://arxiv.org/stat/1409.5185

[16] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015, pp. 448-456.

[17] SOURCEFORGE.NET, "Leaf recognition algorithm for plant classification using probabilistic neural network," Available http://flavia.sourceforge.net/

Wang-Su Jeon received his B.S. degree in Computer Engineering from Kyungnam University, Changwon, Korea, in 2016, and is currently pursuing an M.S. degree in IT Convergence Engineering at Kyungnam University, Changwon, Korea. His present interests include computer vision, pattern recognition and machine learning.
E-mail: jws2218@naver.com

Sang-Yong Rhee received his B.S. and M.S. degrees in Industrial Engineering from Korea University, Seoul, Korea, in 1982 and 1984, respectively, and his Ph.D. degree in Industrial Engineering from Pohang University of Science and Technology, Pohang, Korea. He is currently a professor in the Department of Computer Engineering, Kyungnam University, Changwon, Korea. His research interests include computer vision, augmented reality, neuro-fuzzy, and human-robot interfaces.
E-mail: syrhee@kyungnam.ac.kr