Vietnam Journal of Computer Science
Vol. 7, No. 2 (2020) 197–208

DOI: 10.1142/S2196888820500116

Plant Identification Using New Architecture Convolutional Neural Networks Combine with Replacing the Red of Color Channel Image by Vein Morphology Leaf

Hiep Xuan Huynh*,‡, Bao Quoc Truong†,§, Kiet Tan Nguyen Thanh*
and Dinh Quoc Truong*,¶

*College of Information & Communication Technology
Can Tho University, Can Tho City, Vietnam

†College of Engineering Technology
Can Tho University, Can Tho City, Vietnam

‡hxhiep@ctu.edu.com
§tqbao@ctu.edu.com
¶tqdinh@ctu.edu.com

Received 2 June 2019
Accepted 30 January 2020
Published 4 March 2020

The determination of plant species from field observation requires substantial botanical expertise, which puts it beyond the reach of most nature enthusiasts. Traditional plant species identification is almost impossible for the general public and challenging even for professionals who deal with botanical problems daily such as conservationists, farmers, foresters, and landscape architects. Even for botanists themselves, species identification is often a difficult task. This paper proposes a model deep learning with a new architecture Convolutional Neural Network (CNN) for leaves classifier based on leaf pre-processing extract vein shape data replaced for the red channel of colors. This replacement improves the accuracy of the model significantly. This model experimented on collector leaves data set Flavia leaf data set and the Swedish leaf data set. The classification results indicate that the proposed CNN model is effective for leaf recognition with the best accuracy greater than 98.22%.

Keywords: Deep learning; convolutional neural networks; leaf classification.

1. Introduction

Image-based methods are considered a promising approach for species identification. A user can take a picture of a plant in the field with the built-in camera of a mobile device and analyze it with an installed recognition application to identify the species or at least to receive a list of possible species if a single match is impossible. By using

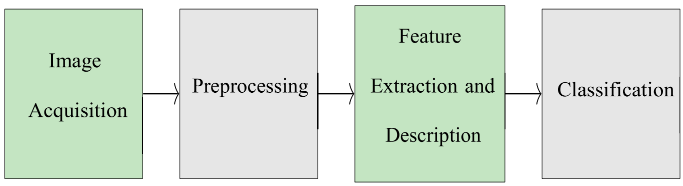

Fig. 1. Generic steps of an image-based plant classification process.

a computer-aided plant identification system, even non-professionals can take part in this process. An image classification process can generally be divided into steps as shown in Fig. 1.

Image acquisition: The purpose of this step is to obtain the image of a whole plant or its organs so that analysis towards classification can be performed. The aim of image preprocessing is enhancing image data so that the undesired distortions are suppressed and image features that are relevant for further processing are emphasized. The preprocessing sub-process receives an image as input and generates a modified image as output, suitable for the next step, the feature extraction. Preprocessing typically includes operations like image denoising, image content enhancement, and segmentation. These can be applied in parallel or individually, and they may be performed several times until the quality of the image is satisfactory. Feature extraction and description: Feature extraction refers to taking measurements, geometric or otherwise, of possibly segmented, meaningful regions in the image. Features are described by a set of numbers that characterize some property of the plant or the plant's organs captured in the images (aka descriptors). Classification: In the classification step, all extracted features are concatenated into a feature vector, which is then being classified.

Tradition image classification is usually based on features, such as SIFT, HOG, SURF, combined with a learning algorithm in these features engineering spaces such as SVM, Neuron, and KNN. The efficiency of all approaches depends heavily on the predefined features. Image features engineering itself is a complex process that requires changes and recalculation for each problem or associated data set.

With the development of neural networks, neural network architecture has been used as an effective solution to extract high-level features from data. Deep Convolutional Neural Network architectures can accurately portray highly abstract properties with condensed data while preserving the most up-to-date characteristics of raw data. This is beneficial for classification or prediction. In recent times, Convolutional Neural Network (CNN) has emerged as an effective framework for describing features and identities in image processing. CNN can learn basic filters automatically and combine them hierarchically to describe underlying concepts to identify patterns. While using CNN one does not need to separately perform features engineering that is time and effort consuming. The generalization of the method makes it a practical and scalable approach to the various application problems of classification and recognition.

Another substantially studied local feature approach is the histogram of oriented gradients (HOG) descriptor.[1] The HOG descriptor, introduced by Lowe,[2] is similar to SIFT, except that it uses an overlapping local contrast normalization across neighboring cells grouped into a block. Since HOG computes histograms of all image cells and there are even overlap cells between neighbor blocks, it contains much redundant information making dimensionality reduction inevitably for further extraction of discriminant features. Therefore, the main focus of studies using HOG lies in dimensionality reduction methods.

SIFT has been proposed and studied for leaf analysis by Chathura and Withanage[3]; Hsiao *et al.*[4]; Lavania and Matey.[5] A challenge that arises for object classification rather than image comparison is the creation of a codebook with trained generic key points. The classification framework by Chathura and Withanage[3] combines SIFT with the Bag of Words (BoW) model. The BoW model is used to reduce the high dimensionality of the data space. Hsiao *et al.*[4] used SIFT in combination with sparse representation (aka sparse coding) and compared their results to the BoW approach. The authors argue that in contrast to the BoW approach, their sparse coding approach has a major advantage as no re-training of the classifiers for newly added leaf image classes is necessary. In Lavania and Matey,[5] SIFT is used to detect corners for classification. Nguyen *et al.*[6] studied speeded up robust features (SURF) for leaf classification, which was first introduced by Bay *et al.*[7] The SURF algorithm follows the same principles and procedure as SIFT. However, details per step are different. The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against image transformations than SIFT.[7] To reduce the dimensionality of extracted features,[6] apply the previously mentioned BoW model and compared their results with those of Pham *et al.*[1] SURF was found to provide better classification results than HOG Pham *et al.*[1]

Ren *et al.*[8] proposed a method for building leaf image descriptors by using multi-scale local binary patterns (LBP). Initially, a multi-scale pyramid is employed to improve leaf data utilization and each training image is divided into several overlapping blocks to extract LBP histograms in each scale. Then, the dimension of LBP features is reduced by a PCA. The authors found that the extracted multi-scale overlapped block LBP descriptor can provide a compact and discriminative leaf representation.

Oide and Ninomiya[9] used neural networks to classify soybean leaves using a Hopfield network and a simple perceptron. Krizhevsky *et al.*[10] used Deep Convolutional Neural Networks for ImageNet and their research results created a new rush for deep learning.

Several publications suggested the use of CNN in leaf classification in recent years. Jassmann *et al.* (2015)[11] developed an application for classifying plants, based on leaf images. The system uses a CNN in a mobile application for mobile phones to categorize the nature of the leaf, trained with the ImageCLEF data set. The proposed architecture consists of a convoluted layer, followed by a composite layer and two fully connected layers applied to the 60×80-pixel input image. Wu *et al.* (2016)[12]

proposed a simplified version of AlexNet for leaf recognition. They used parametric linear units (PReLU) instead of ReLU. Plant disease identification includes the processing of leaf recognition. Sladojevic *et al.*[13] are interested in a new method for developing a disease-identification model based on leaf classification of images, using CNN. The developmental model was able to recognize 13 healthy plant leaf diseases, with the ability to discriminate leaves from the surrounding environment.

The main objectives of this paper are (1) Description of the CNN model plant classification from leaf patterns (2) Improving the accuracy of the model by replacing the red of color channel input image by extract vein morphology leaf and (3) Comparison of results of leaves recognition other methods.

2. Methods

The input data of the model is the image, which normally has three RGB channels. The green and blue channels of colors are better to represent the pattern of leaves. We use green and blue channels of colors and extract vein morphology to replace the red channel of colors.

In Fig. 2, in our scheme implementation, there are two kinds of segmentation data: (V) extract vein morphology by adaptive local threshold algorithm; (VGB) we replace the red of color channel input image by extracting vein morphology leaf. We use two data sets leaf public which are from Swedish and Flavia leaf data set for experiment.

2.1. *Extract leaf vein shape*

To perform the extraction of leaf vein shape, the image segmentation process involves converting the image to grayscale, and then using adaptive thresholding techniques to segment the image and extract the vein leaf image.

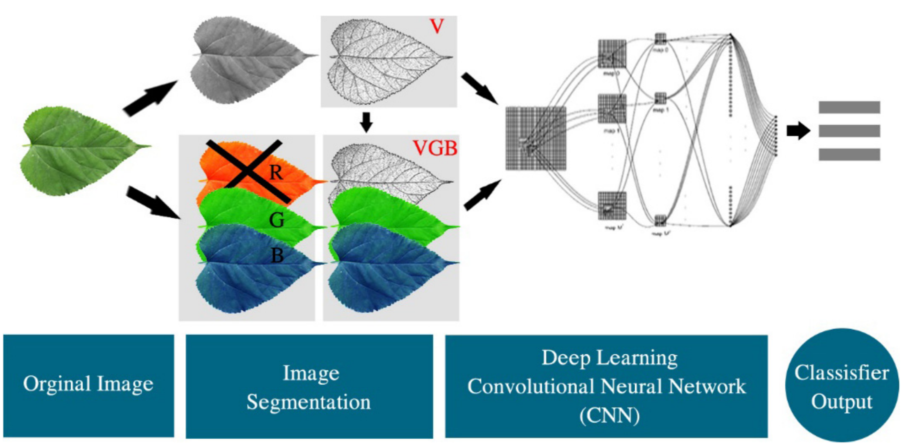

Fig. 2. Scheme of proposed solution.

Fig. 3. Illustrated image with the adaptive local threshold to mean (10), $C = 0.2$.

There are many image processing techniques used to segment the image, many researchers extracted vein morphology from images obtained by the camera and using Gabor filters,[14] Colony filters,[15] thresholds,[16] independent component analysis,[17] we use an adaptive local threshold to segment the image and look for leaf vein shape.

We use adaptive local threshold algorithm that decouples object from background with suggested heterogeneous illumination: bw = adaptive threshold (IM, ws, C) produces a bw binary image with adjacent thresholds is the mean (ws) — C, ws is the neighborhood size, C is the constant, in this study we use $ws = 10$ and $C = 0.2$, these two results in low noise picture most from images in our experiments. Figure 3 shows the result illustrating an adaptive local threshold.

2.2. *Convolutional Neural Network*

2.2.1. *Convolutional Neural Network (CNN)*

CNN is an effective identification method, developed in recent years, that caused widespread attention. Now, CNN has become one of the most efficient methods in the field of pattern classification and recently has been used more widely in the field of image processing[10,18] and it can reach a better performance than the traditional methods[19] through wide verification. CNN consists of one or more pairs of convolutional and max-pooling layers. A convolutional layer applies a set of filters that process small local parts of the input where these filters are replicated along with the whole input space. A max-pooling layer generates a lower-resolution version of the convolutional layer activations by taking the maximum filter activation from different positions within a specified window. This adds translation invariance and tolerance to minor differences of positions of objects parts. Higher layers use more broad filters that work on lower-resolution inputs to process more complex parts of the input. Top fully connected layers finally combine inputs from all positions to do the classification of the overall inputs. This hierarchical organization generates good results in image processing tasks.

2.2.2. *Convolutional Neural Network Structure*

The CNN has these components:

- *Convolution layer*: The convolution operation extracts different features of the input. The first convolution layer extracts low-level features like edges, lines, and corners. Higher-level layers extract higher-level features.

- *Nonlinear layers*: Neural networks, in general, and CNNs, in particular, rely on a nonlinear "trigger" function to signal distinct recognition of likely features on each hidden layer. CNNs may use a variety of specific functions such as rectified linear units (ReLUs) and continuous trigger (nonlinear) functions to efficiently implement this nonlinear triggering. An ReLU implements the function $y = \max(x, 0)$, so the input and output sizes of this layer are the same. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer. In comparison to the other nonlinear functions used in CNNs (e.g., hyperbolic tangent, absolute of hyperbolic tangent, and sigmoid), the advantage of an ReLU is that the network trains many times faster. ReLU functionality is illustrated in Fig. 4.
- The pooling/subsampling layer reduces the resolution of the features. It makes the features robust against noise and distortion.
- Fully connected layers are often used as the final layers of a CNN. These layers mathematically sum a weighting of the previous layer of features, indicating the precise mix of "ingredients" to determine a specific target output result. In case of a fully connected layer, all the elements of all the features of the previous layer get used in the calculation of each element of each output feature.

Training is performed using a "labeled" data set of inputs in a wide assortment of representative input patterns that are tagged with their intended output response. The training uses general-purpose methods to iteratively determine the weights for intermediate and final feature neurons. Figure 5 demonstrates the training process at a block level.

2.3. *The proposed CNN architecture*

CNN architectures vary with the type of images and especially when input image sizes are different. In this paper, we propose CNN architecture for input size 128×128 px.

The proposed architecture is described in Table 1.

The CNN architectures are five layers including: [Conv1 - ReLu - Max pool] $\rightarrow$ [Conv2 - ReLu - Max pool] $\rightarrow$ [Conv3 - ReLu] $\rightarrow$ [Conv4 - FC] $\rightarrow$ Softmax. The final

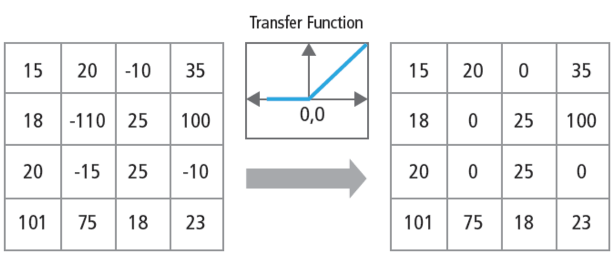

Fig. 4. Pictorial representation of ReLU functionality.

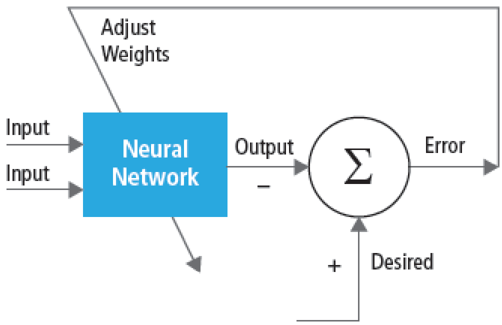

Fig. 5. Training of neural networks.

Table 1. The proposed CNN architecture (images input 128 × 128).

Type	Conv1	Pool	Conv2	Pool2	Conv3	Conv4	
Filter size	8 × 8	7 × 7	5 × 5	6 × 6	2 × 2	3 × 3	
Stride	2	2	2	2	1	1	Softmax
Number of filters	400	400	100	100	50	50	
Output size	61 × 61	28 × 28	13 × 13	4 × 4	3 × 3	1 × 1	

layer has n units corresponding to n and category of leaf data sets. After all layers, a SoftMax loss is placed.

3. Experiment and Results

3.1. *Experiment and data sets*

In order to test the performance of the classification system, we selected two standard sets as follows:

- Swedish leaf data set: The Swedish leaf data set has been captured as part of a joined leaf classification project between the Linkoping University and the Swedish Museum of Natural History.[20] The data set contains images of isolated leaf scans on a plain background of 15 Swedish tree species, with 75 leaves per species (1125 images in total). This data set is considered very challenging due to its high inter-species similarity.
- Flavia data set: This data set contains 1907 leaf images of 32 different species and 50–77 images per species. Those leaves were sampled on the campus of the Nanjing University and the Sun Yat-Sen arboretum, Nanking, China. Most of them are common plants of the Yangtze Delta, China.[21] The leaf images were acquired by scanners or digital cameras on a plain background. The isolated leaf images contain blades only, without petioles.

3.2. *Data augmentation*

One of the reasons of overfitting of the model is insufficient number of training vectors. In such cases, augmentation of the data is an effective solution. This is why we have increased the number of training images by creating three copies of each image after reflection and rotation. Thus, each original image creates three augmented images [Fig. 6].

Data partitioned for the experiment is shown in Table 2.

In this study, the input data of the model is scanned or taken from leaves of the trees, then model the training. The experimental process is as follows:

- Standardized image size: the image resizes to 128×128 px to match the input of the network. To resize images to the desired, first, images are resized such that the larger dimension of them is equal to 128, then the smaller ones are padded with pixels having the value of 0.
- Image Partition: Each category is partitioned as shown in Table 2 for experimental purposes.
- Initialization parameter: Learning rate: set to 0.00005; WeightDecay constant (anti-overfitting) = 0.0005; and Momentum constant = 0.9.

The above parameters are chosen based on experimental results for the proposed model by trial and error method and they give the best results during the experiment. Training time depends on computer resources with GPU or CPU, Matlab software and Matconvnet tool.

3.3. *Experiment results*

For creation and testing of the CNN model, data sets are separated to 70%, 10% and 20% as training, validation and test sets, respectively. Then augmentation data of training set is performed.

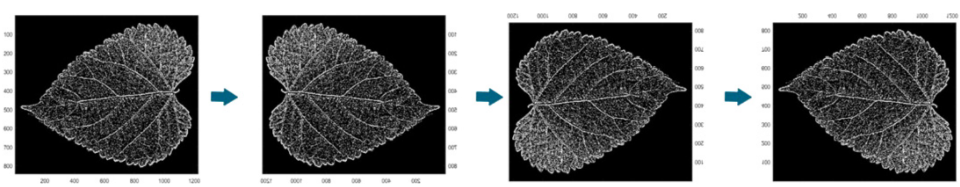

Fig. 6. Data with specific augmented (with reflection augmentation).

Table 2. Partition data.

			Train		Val	Test
Data set	Number layers	Number samples	70%	Augmentation data	10%	20%
Swedish	15	1125	788	3152	212	225
Flavia	32	1907	1335	5340	190	382

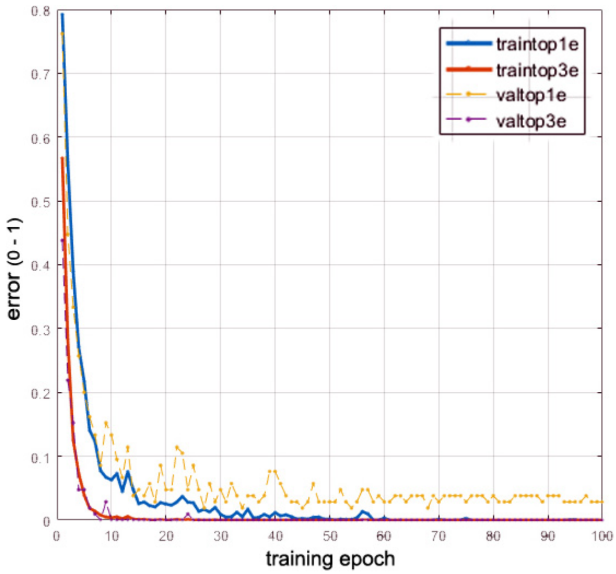

Fig. 7. The curves of validation error of experiments of the Swedish data set with augmentation three channels VGB. Depending on the random initialization value of weight, the error level is gradually reduced, after each training epoch.

With the set of hyperparameters selected, we perform 100 runs with random initial sets of weights of connections [Fig. 7]. We performed early stopping on the number of stochastic gradient descent iterations by monitoring the validation error. The validation errors and classification accuracies were estimated by averaging over 100 runs. The leaf classification accuracies (the average precision of (Number of correct identities/Total leaf of test set) $\times 100\%$) of the proposed CNN on Flavia and Swedish leaf data sets are reported in Table 3. For each experiment on a data set with specific augmented (with reflection augmentation) data sets are presented.

From the experiment results in Table 3, we have the following comments.

The test accuracy of the model is higher on VGB data than on RGB data set. However, storage and training time is longer. With the usage of pre-processing of

Table 3. Experiment results for CNN model (where V is the channel with extracted image of leaf veins).

Rate train/val/test 70%-10%-20%	Number channels	Validation error/accuracy test set CNN (%)
Swedish (15 layers)	3 (RGB)	2.04/97.33
	3 (VGB)	1.92/98.22
	1 (V)	2.42/96.44
Flavia (32 layers)	3 (RGB)	4.05/95.5
	3 (VGB)	5.41/96.63
	1 (V)	5.80/94.52

Table 4. Comparison of results of leaves recognition methods on the Swedish data set.

Descriptor	Feature	Classifier	Accuracy (%)
IDSC	Shape	SVM/k-NN	93.73/94.13
TOA	Shape	k-NN	95.20
TSL	Shape	k-NN	95.73
TSLA	Shape	k-NN	96.53
LBP	Shape	SVM	96.67
I-IDSC	Shape	1-NN	97.07
MARCH	Shape	1-NN	97.33
Our method	**Shape + vein**	**CNN**	**98.22**
DS-LBP	Shape + texture	Fuzzy k-NN	99.25

veins extract, the model accuracy improved. In Table 3, it is observed that classification accuracy of CNNs based on extract of veins of leaves slightly outperforms original leaves-based models. But it must be noted that the extraction of leaves veins slows down the whole process in comparison to the solution without this step. Tables 4 and 5 show that CNN-based feature learning and classification outperforms results of conventional approaches such as e.g., TSL, IDSC or MARCH.

Comparisons of the results were compiled in Wäldchen and Mäder[22] with the method of leaves recognition on the Swedish and Flavia data sets in Tables 4 and 5. Here, the accuracy is (Number of correct identities/Total leaf of test set) $\times 100\%$

From experiment results with Swedish and Flavia data sets, we can confirm that the CNN-based neural network model, which we propose, works very well on classification problem of leaves based on the shape of veins (vein morphology). This result once again confirms the effectiveness and simplicity of the CNN depth geometry model for real-world problems with large data. The recognition process is done by simply building the model and determining the appropriate parameters. Manual identification of important features of the image, a process that takes a lot of time and effort. The effectiveness of the classification process and improved recognition accuracy with usage or proposed CNN are not dependent and do not need manual preprocessing of data. All those aspects form the value of the proposed solution.

Table 5. Comparison results of leaves recognition methods on the Flavia data set.

Descriptor	Feature	Classifier	Accuracy (%)
SMSD, FD, CM	Color + shape	k-NN, DT	91.30
SMSD	Shape	PNN	91.40
SMSD, CM	Shape + color	RF (k-NN, NB, SVM)	93.95
SMSD, $A_{\text{vein}}/A_{\text{leaf}}$	Shape + vein	SVM (k-NN)	94.50
SIFT	Shape	SVM	95.47
SURF	Shape	SVM	95.94
SMSD, FD	Shape	BPNN	96.00
SMSD, CM, GLCM, $A_{\text{vein}}/A_{\text{leaf}}$	Shape + color+ texture + vein	SVM	96.25
Our method	**Shape + vein**	**CNN**	**96.63**

4. Conclusions

Majorly in this work, a successful application of deep learning to the area of agriculture is reported, specifically to plant identification from a leaf. The main result is that we obtained an improved accuracy by using a standard deep learning model. The approach proposed will replace a channel of the original image with an extract of leaf veins and augmentation data to reduce overfitting by reflecting and rotating images.

It is commonly argued that neural networks do not provide any new insight into the problem under study, as they fall into the category of black-box models. However, thanks to a simple visualization technique, it can be shown that images of vein patterns can be helpful for the considered classification task.

References

1. N. H. Pham, T. L. Le, P. Grard and V. N. Nguyen, Computer aided plant identification system, *2013 Int. Conference on Computing, Management and Telecommunications (ComManTel)*, Hochiminh, Vietnam, 2013, pp. 134–139.
2. D. G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vis.* **60**(2) (2004) 91–110.
3. P. H. Chathura and D. Withanage, Computer assisted plant identification system for android, *2015 Moratuwa Engineering Research Conf. (MERCon)*, University of Moratuwa, Sri Lanka, 2015, pp. 148–153.
4. J. K. Hsiao, L. W. Kang, C. L. Chang and C. Y. Lin, Comparative study of leaf image recognition with a novel learning-based approach, *2014 Science and Information Conf. (SAI)*, London, 2014, pp. 389–393.
5. S. Lavania and P. S. Matey, Leaf recognition using contour based edge detection and sift algorithm, *2014 IEEE Int. Conf. on Computational Intelligence and Computing Research (ICCIC)*, Tamilnadu, India, 2014, pp. 1–4.
6. Q. K. Nguyen, T. L. Le and N. H. Pham, Leaf based plant identification system for android using surf features in combination with bag of words model and supervised learning, *2013 Int. Conf. on Advanced Technologies for Communications (ATC)*, Hochiminh, Vietnam, 2013, pp. 404–407.
7. H. Bay, T. Tuytelaars and G. L. Van, Surf: Speeded up robust features, *European Conf. on Computer Vision*, Springer, Berlin, 2006, pp. 404–417.
8. X. M. Ren, X. F. Wang and Y. Zhao, An efficient multi-scale overlapped block LBP approach for leaf image recognition, *Proc. of the 8th Int. Conf. on Intelligent Computing Theories and Applications (ICIC'12)*, Springer, Berlin, 2012, pp. 37–43.
9. M. Oide and S. Ninomiya, Discrimination of soybean laeflet shape by neural networks with image input, *Comput. Electorn. Agric.* **29**(1–2) (2000) 59–72.
10. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, Nevada, USA, 2012, pp. 1097–1105.
11. T. J. Jassmann, R. Tashakkori and R. M. Parry, Leaf classification utilizing a convolutional neural network, *SoutheastCon 2015* IEEE, Fort Lauderdale, FL, USA, 2015, pp. 1–3.
12. Y. H. Wu, L. Shang, Z. K. Huang, G. Wang and X. P. Zhang, Convolutional neural network application on leaf classification, in *Intelligent Computing Theories and Application ICIC 2016*, Lecture Notes in Computer Science, Vol. 9771, eds. D. S. Huang, V. Bevilacqua and P. Premaratne (Springer, 2016), pp. 12–17, https://link.springer.com/chapter/10.1007%2F978-3-319-42291-6_2.

13. S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic, Deep neural networks based recognition of plant diseases by leaf image classification, *Comput. Intell. Neurosci.* **2016**, 3289801.
14. A. Vini Katyal, Leaf vein segmentation using Odd Gabor filters and morphological operations, *International Journal of Advanced Research in Computer Science* **3**(3) (2012) 258–261.
15. S. James and P. R. Cope, The extraction of venation from leaf images by evolved vein classifiers and ant colony algorithms, in *Advanced Concepts for Intelligent Vision Systems*, Series Lecture Notes in Computer Science, Vol. 6474 (2010), pp. 135–144.
16. X. W. Xiaodong Zheng, Leaf vein extraction based on gray-scale morphology, *Int. J. Image, Graph. Signal Process.* **2** (2010) 25–31, http://www.mecs-press.org/.
17. Y. Li and Z. Chi, Leaf vein extraction using independent component analysis, *IEEE Int. Conf. on Systems, Man, and Cybernetics*, Taipei, Taiwan, 2006, pp. 3890–3894.
18. Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature* **521** (2015) 436–444.
19. K. Chatfield, V. S. Lempitsky, A. Vedaldi and A. Zisserman, The devil is in the details: An evaluation of recent feature encoding methods, *British Machine Vision Conf.*, Vol. 2, Dundee, UK, 2011, p. 8.
20. O. J. O. Soderkvist, *Computer Vision Classification of Leaves from Swedish Trees* (Linkoping University, Linkoping, 2001).
21. S. G. Wu, F. S. Bao, E. Y. Xu, Y. X. Wang, Y. F. Chang and Q. L. Xiang, A leaf recognition algorithm for plant classification using a probabilistic neural network, *2007 IEEE Int. Symp. on Signal Processing and Information Technology*, IEEE, Giza, Egypt, 2007, pp. 11–16.
22. J. Wäldchen and P. Mäder, Plant species identification using computer vision techniques: A systematic literature review, *Arch. Comput. Methods Eng.* **25**(2) (2017) 507–543, ISSN: 1134-3060 (Print) 1886-1784 (Online), https://www.researchgate.net/publication/312147459.