Leaf classification by combining hand-crafted and deep features

2 authors:

Rafflesia Khan
University of Limerick

Rameswar Debnath
Khulna University

BCS Journal of Computer and Information Technology
Vol. 01, 2019, ISSN: 2664-4592 (Print), 2664-4606 (Online)

Leaf Classification by Combining Hand-Crafted and Deep Features

Rafflesia Khan and Rameswar Debnath
Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh.
E-mail (s): rafflesiakhan.nw@gmail.com, rdebnath@cseku.ac.bd

Abstract: Automated leaf identification is considered as a challenging problem because of rich morphological similarities among different leaf species. Traditional machine learning-based models along with hand-crafted features fail to differentiate each variation of leaf property individually. Besides, if the training datasets for leaf images are small in size, poor deep features are obtained in deep learning-based models which limits the models from performing accurately. In this paper, we propose a leaf identification model based on Convolutional Neural Network (CNN) where hand-crafted features are combined with deep features for model training. We apply five hand-crafted feature selection methods such as Harris corner detector, RGB to LAB color transformation, wavelet transformation, adaptive thresholding and Gabor filter. These feature selection methods give various morphological information about leaves of different species. These hand-crafted features increase training data with more information that can improve the performance of CNN model. We experiment our proposed model on different benchmark dataset and see that proposed model shows consistency and superiority compared to the existing solutions for leaf identification. From the result, we see that our proposed model achieves accuracy 99.64% on Flavia dataset with 32 leaf species and 98.13% on MalayaKew dataset with 44 leaf species.

Keywords: Plant Leaf Identification, Hand-Crafted Features, Deep Learning Features, Convolutional Neural Network.

1. Introduction

Plants are one of the most essential parts of nature and human lives. Plants foster the ecosystem by supplying oxygen and food for both mankind and animal at the same time clearing-out carbon dioxide from the environment. Along with that, plants play a significant role in keeping temperature stable, maintaining ecological balance, preventing soil erosion, water pollution, air braking, etc. Apart from these, plants are used in furniture industry, chemical industry, cosmetic industry, botanical medicine (e.g., ayurvedic medicine) and also for biodiversity conservation. Proper plant identification is essential for industries such as drug, chemical, cosmetics, etc., agricultural productivity and also for identifying rear as well as endangered plants. Generally, plants are identified by observing their morphological characteristics such as structure of stems, roots, leaves, flowers, fruits, etc. followed by the analysis and consultation of a guide of information or a known database [1].

Every year industries, agricultural institutions, ecological protection agencies spend billions of dollars alongside experienced botanists give hedge effort for plant identification. But now a days, with the help of computer vision, image processing and artificial intelligence we can automatically distinguish a large number of plant species within a very short time. Each different spices of plant have unique leaf, therefore, leaf recognition and classification plays an important role in plant classification. Some leaves have medicine value while other leaves of different class of plant who can look similar but does not have the same medicinal value. In the herbal industry, recognizing correct leaf is very important. For example, Cannabidiol is a phytocannabinoid discovered in 1940 which is projected to have 20 billion markets in pharmaceutical as well as beauty product by 2020. Cannabidiol looks similar to marijuana. An efficient leaf classification algorithm can reduce the risk of using wrong leaf which may cause huge risk.

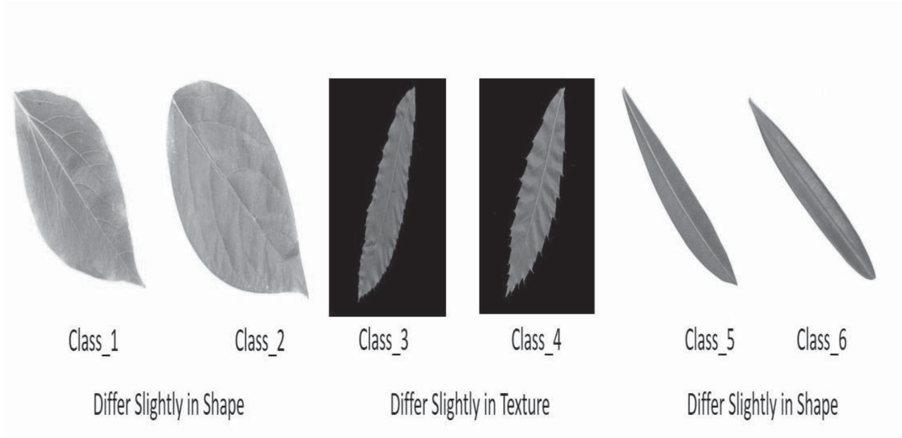


Fig. 1. Almost similar leaf of different species

Leaf identification is one of the major fields of computer vision and object recognition research area because of the many application areas of leaf. Most of the plant leaves are of the same color with almost similar shape and/or texture as shown in **Fig. 1** which makes the identification process complex. **Fig. 1** shows an image of almost similar leaves of different species where identifying each leaf differently is quite difficult. Thus, nondifferentiable similarities among leaves confuse classifiers. So, there exists a necessity of proper leaf classification model which can differentiate every nondifferentiable similarity to identify very similar leaves differently and serve their application areas.

However, a successful development of leaf identification system can also explore to other classification tasks such as genre detection of different animal, fruit, flower or other object.

Different researchers are using different methodologies for identifying leaves as well as plants with less complexity and more accurately. In [1], authors use shape and texture feature with SVM (Support Vector Machine) classifier for leaf classification. Other solutions employ shape-color-texture [2], color-texture [3], moment invariant and texture [4] to identify plant-leaf. Some recent models use deep features [5] where another model combines HOG, KNN, ANN for better recognition rate [6]. Most of the existing models either use hand-crafted features [1-4] or use deep learning features [5-6]. Although hand-crafted features provide good result, these often get affected by image acquisition process, noise in image, angle, brightness, scale, rotation of image etc. On the other hand, deep learning features provide better performance in identification but can suffer with over-fitting problem or dataset bias problem due to lack of necessary training data [7,8].

Considering the existing challenges, we propose a plant leaf recognition model where we combine hand-crafted and deep features for ensuring better performance over existing methods. The main contributions of this paper are as follows:

- We apply five significant hand-crafted feature selection methods such as Harris corner detector, RGB to LAB color transformation, wavelet transformation, adaptive thresholding and Gabor filter that are capable of differentiating every closely related leaf of different species.
- We propose a new deep feature extraction model architecture that learns CNN or deep feature combined with hand-crafted feature for leaf identification.
- We present comparative analysis on the performance of our model with existing models using two benchmark datasets (e.g., Flavia [9] and MalayaKew Dataset [10]).

The rest of the paper is organized as follows: Section 2 presents some literature review on plant leaf identification problem. Working principle of our proposed model is described in Section 3. In Section

4, the proposed model is validated on different datasets and a comparative analysis is made. Section 5 concludes the paper with some future instructions.

2. Related Work

There have been many approaches proposed by researchers for effective plant leaf identification by image processing, feature extraction and pattern recognition techniques.

Kumar, T. P. et al. [1] proposed a model for leaf classification that uses shape and texture feature and they conclude that Gabor filter gives better texture description over HOG. The maximum accuracy of model [1] was 97.64%. Haque F, Haque, et al. [11] derived seven features from geometric parameters of leaf shape. But their working process is slow and not applied on big dataset. In a comparative analysis [12] Arafat, Syed Yasser, et al. explained that HOG has an accuracy of 97%, C-Sift has the accuracy 98% and for MSER the accuracy is 90% for Flavia dataset.

Wang, Zhaobin et al. [13] used SVM classifier with the entropy sequence obtained by plus-coupled neural network (PCNN) as key feature and other ancillary features. Lee, Sue Han, et al. [14] proposed a new hybrid global-local feature extraction model which integrates information of two CNN models using different train data formats. Zhang, Chaoyun, et al. [15] presented a seven-layer CNN model with data augmentation and oversampling where they produce 100 predictions for each training image and finally gets 94.5% accuracy. In [16], Jeon et al. constructed a 10-layer CNN model with data augmentation and achieved overall 87.91% accuracy.

In the next section, we will describe our proposed method that combines hand-crafted features with deep features.

3. Proposed Methodology

There exist several plant-leaf identification methods that use hand-crafted features for classification. Other existing models create a CNN network that automatically extracts the most essential features and progressively creates higher-level features out of commonly found groups of pixels in the images, which even boost its performance. The features produced by CNN network is called deep features.

In case of leaf recognition, the identification process becomes harder as most of the leaf species have similar morphological characteristics i.e., shape, texture, size, venation, etc., are almost identical. Classifiers cannot recognize properly due to nondifferentiable similarities among leaves. Thus, a good model needs to be capable of finding each and every subtle feature differently.

CNN models find patterns in an image considering pixel-by-pixel similarity. But unpreprocessed general leaf image contains noise and also do not provide enough morphological details to CNN classifier. In this paper, we consider combining hand-crafted features containing more morphological details with deep features into a single model. We apply five efficient hand-crafted features selection methods to get more morphological features. Each type of features creates an image of the same size of the input image and becomes an individual input data for training.

Generally, CNN scales up to a network to millions of parameters. It needs massive labelled data of each class to support the learning process for a good performance [17]. Considering this reason, most of the existing models enlarge the dataset by rotating every image in different angle [14, 16]. Unlike the existing methods, we apply different hand-crafted feature selection methods that give different morphological features and enlarge the dataset too.

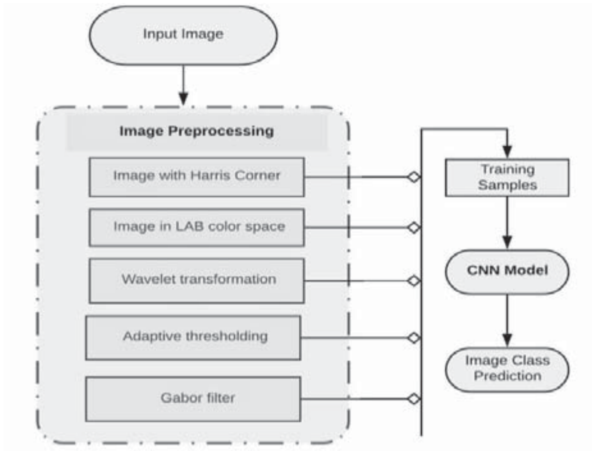


Fig. 2. Proposed model flow chart

Fig. 2 shows a flow chart of our proposed model. We divide our working methodology into two major part: (1) Image preprocessing (i.e., hand-crafted features learning), (2) Convolution Neural Network (i.e., deep features learning and classification).

3.1 Image Preprocessing

We apply Harris corner detector, RGB to LAB color transformation, wavelet transformation, adaptive thresholding and Gabor filter methods. Then each transformed data is converted into an image. Each of these five images contains more different and efficient details of their corresponding input image. The methods are described later. **Fig. 3** shows the five conversion images of a sample input image. These five images along with the original input image that is six images for each input image is fed into CNN.

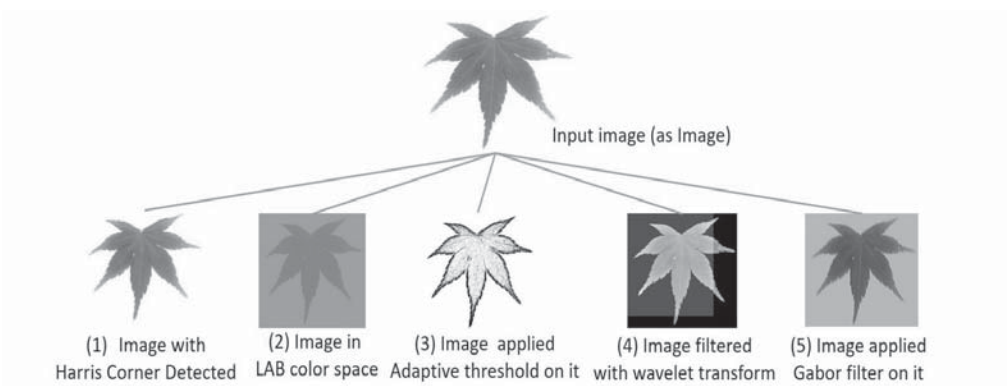


Fig. 3. Five conversion images from an input image

3.1.1 Harris Corner Detector

Most of the time the shape of a leaf is considered as an important feature [18]. Also, most often the shape of a leaf image varies from species to species which helps any classifier to separate every different species more individually and more accurately. Our main objective is to feed the CNN model more clues about the morphological characteristics or features of leaf image by providing more information about leaf in every image pixel. We apply Harris Corner Detector [19] for describing the outer edge of a leaf as shown in **Fig. 3 (1)**. In Harris Corner Detector a window is moved around an image and every large variation is detected. Harris Corner Detector detects difference in intensity for window displacement of E(p, q) in x and y direction of an image following **Eq. 1**, where w(x, y) represents window function, I(x+u, y+v) represents shifted intensity and I(x, y) represents source image intensity [20].

$$\mathrm{E(p, q)} = \sum_{x,y} \mathrm{w(x, y)[I(x+u, y+v) - I(x, y)]^2} \qquad (1)$$

Then based on the variation score, the corners of an image are figured out [20]. Eventually, by finding corners in a leaf image, we find the contour of a leaf in an image. So, image with Harris Corner provides more information, about a leaf shape to image pixels and thus contributes to better CNN model creation.

3.1.2 RGB to LAB Color Transformation

Another image with hand-crafted feature is an image created by transforming the input image channel from RGB to LAB color space as shown in **Fig. 3 (2)**. In LAB color space, colors of an image are defined independently of their nature of creation or the device they are displayed on [21]. So, this property of LAB facilities our recognition model to learn deep feature independent of any platform or device.

3.1.3 Wavelet Transform

Recently wavelet transform is used for de-noising and compression of images and signals. Digital images are viewed or processed at multiple resolutions using the Discrete Wavelet Transform (DWT). It provides powerful insight into an image's spatial and frequency characteristics [22]. Applying DWT on an image, we get the approximation of horizontal, vertical and diagonal detail coefficients of an image and then we combine these coefficients back. This forward and inverse DWT in 2D image gives us an image with more information about its horizontal, vertical and diagonal detail coefficients as shown in **Fig. 3 (3)**. For our model, we have used Haar wavelets with a threshold value of level 1 that helps to reduce noise as well as redundant information of an image.

3.1.4 Adaptive Thresholding

Adaptive Thresholding [23] uses an algorithm that calculates the threshold for a small region of the image so that we can get different thresholds for different regions of the same image as shown in **Fig. 3 (4)**. It gives us better results for images with varying light conditions [24]. Lighting on image always manipulates object classification models to misinterpret a brighter portion of an image area as an efficient feature in those brighter pixels of image. Thus, Adaptive thresholding helps to make every pixel contain even information. Applying this method, we get a binary image where every pixel of the image contains much efficient and evenly distributed features. We convert the thresholded one channel image into a three-channel image for further processing.

3.1.5 Gabor Filter

Gabor filter gives the highest response at every edge and it also points where texture changes [25]. Eventually, it gives us an image with more information about the object's edges and texture in it as shown in **Fig. 3 (5)**. Gabor Filter can be expressed as the following **Eq. 2**.

$$g(x, y, \gamma, \theta, \varphi, \sigma, \lambda) = \exp(-\frac{\acute{x}+\gamma^2\acute{y}^2}{2\sigma^2})\exp(i(2\pi\frac{\acute{x}}{\lambda}+\varphi)) \qquad (2)$$

Here x, y represents the pixel value, γ, θ, φ, σ, λ are different parameters. We have used kernel size 31, σ (i.e., standard deviation of the Gaussian function) value 4.0, λ (i.e., wavelength of the sinusoidal factor) value 10.0, γ (i.e., spatial aspect ratio) value 0.50 and φ (i.e., phase offset) value 0.0 which iterates over the range of values that each pixel in the Gabor kernel can hold.

3.2 Convolution Neural Network (CNN)

We develop a new CNN model architecture with the combination of Convolution layer, Maxpooling layer, Dropout Layer and Fully connected layer. **Fig. 4** shows our CNN model structure. At first, we apply data augmentation process by using keas ImageDataGenerator [27]. The augmented data is used in the CNN training. In the training process, the Convolutional layer performs some mathematical operations and produces a feature map. We then apply the activation function (Relu) on the output to introduce nonlinearities into the model. Batch normalization normalizes the features in input distribution in the learning process. The Maxpooling layer followed by CAB is used to get an input representation by reducing its dimensions. Dropout layer then reduces redundant values from Maxpooling output. A fully connected layer is used to flatten the high-level features that are learned by combining all the layers and convolutional layer. After that, a fully connected layer performs classification on the features extracted by the preceding convolutional layers and down sampled by the pooling layers. Fully connected layers pass the flattened output to the output layer where we have used an activation to predict the input class label. Finally, the class label with the highest prediction value is considered as the actual class of input image leaf.

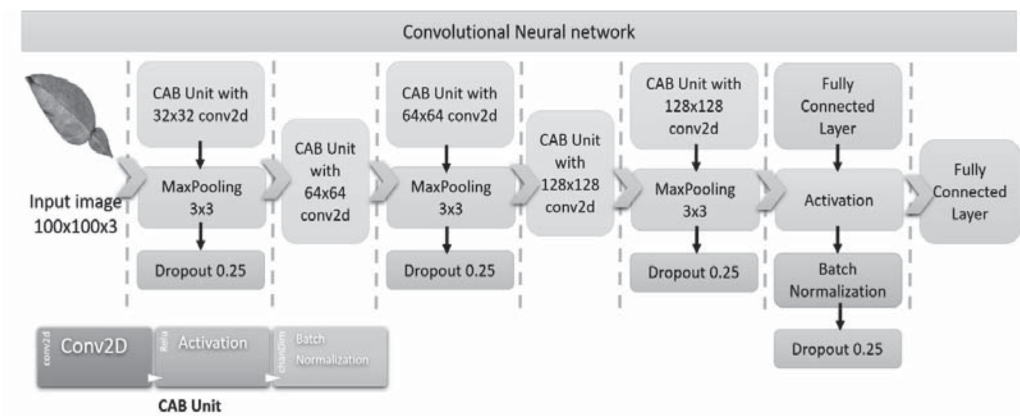


Fig. 4. CNN model of our proposed model

In our CNN model, input image of size 100 x 100 x 3 is passed through a stack of layers. At first, the input image is passed to a 32 x 32 CAB (i.e., convolution, activation and batch normalization unit) followed by 3 x 3 maxpooling of 0.25% dropout. Then a 64 x 64 CAB unit is followed by another 64 x 64 CAB unit with 3 x 3 maxpooling of 0.25% dropout. Next a 128 x 128 CAB unit is followed by another 128 x 128 CAB unit with 3 x 3 maxpooling of 0.25% dropout. After performing all these convolution, pooling and dropout, the output is fed into fully connected layers with 1024 neurons. In this layer activation is performed using Relu with 0.25% dropout followed by batch normalization. The last layer is the fully connected layer predicts the number of classes.

As we train our CNN model with leaf images containing more morphological features that best describe leaves, our CNN model learns much efficient deep feature going through pixel-by-pixel. Eventually, this develops much significant classifier.

4. Results and Comparison

In order to verify the performance of our proposed model, we have applied our model on different datasets. In this section, we present some of our experimental results along with some comparative analysis.

We evaluated our proposed model on both Flavia [9] and MalayaKew Dataset [10]. Dataset [9] contains 32 different leaf species with an approximate 56 images for each class. Dataset [10] contains 44 different leaf species with an approximate 52 train and 12 test images for each class in D1 set (whole leaf images). **Table 1.** shows an overall result of our model on different performance measures (e.g., Accuracy, Precision, Recall, F1 Score). These results are achieved by following the main working process of our proposed model described in section 3. Here in all experiments, classification accuracy is computed to infer the robustness of the system following **Eq. 3.** Where TR = total number of true predicted species, TN = total number of tested images.

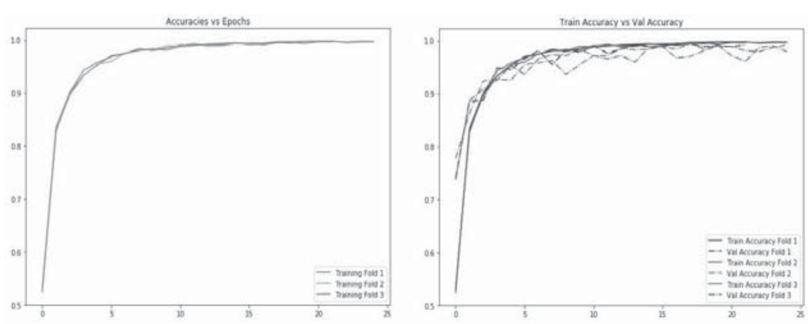
$$Accuracy = TR / TN \qquad (3)$$

Our model archives the results showed in **Table 1.** by using 52 train and 12 test images from D1 set of [10] and 50 train and 16 randomly chosen test images of [9]. We create the conversion images mentioned in Section 3 for only train images.

Table 1: The performance of our proposed model on different performance measures on dataset [9] and dataset [10].

Performance Measure	Dataset [9]	Dataset [10]
Accuracy	0.996	0.9851
Precision(micro avg)	1	0.98
Recall (micro avg)	1	0.98
F1 Score (micro avg)	1	0.98

For better evaluation of our model, we perform K-fold Cross-Validation which is a resampling procedure used to evaluate machine learning models [28]. **Fig. 5** shows the graphical view achieved by applying k-fold cross-validation on [9]. Here we have used, number of folds=3, epochs =100, batch size=128. **Fig. 5(a)** shows training accuracy of 3 different folds where at each fold 10% of training samples (e.g., total 1907 images of 32 class) was randomly chosen as test samples. **Fig. 5(b)** shows training vs validation accuracy of 3 different folds where we remove 12 training samples to be used as test samples of each class.



(a) Training accuracy over number of iteration using 3(k)-folds and randomly chosen 10% of training samples as test samples.

(b) Training accuracy Vs Validation accuracy over number of iteration using 3(k)-folds and 12 test samples different from training samples.

Fig. 5. Accuracy over a number of iterations using k-fold cross-validation

We also perform failure analysis and observed that in dataset [10] most of the misclassification is observed in Class 2, Class 29 and Class 38. From our investigation as illustrated in **Fig. 6,** we found that the leaves of these three classes are very similar in color and shape which make individual identification process much difficult.

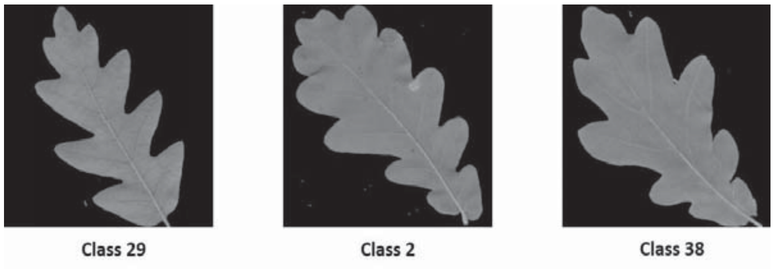


Fig. 6. Image samples from the most similar leaf classes (Class 29, Class 2 and Class 38)

For comparative analysis, we have compared our model's identification capability with some existing works. As within our studied literature, existing models evaluate their proposed model on different datasets, here we divide the comparison into two parts. Part 1: shows a comparison of model evaluation on dataset [9] and Part 2: shows a comparison of model evaluation on dataset [10].

4.2.1 Comparative analysis using dataset [9]

We compare the test accuracy of our model with some existing models. We perform a comparative analysis of our model's performance with the existing models who use dataset [9] for model evaluation. **Table 2** shows a comparative analysis of our model's performance with state of the arts in case of test accuracy**.**

Table 2: The performance of our proposed model on dataset [9].

Model	Features	Classifier	Applied on	Accuracy
Our model	Hand-Crafted Features + Deep Features	CNN	32 Species, Total 50 Training, 10 Test Samples of each species from total 1500 Images.	**0.9964**
Zhang, Chaoyun, et al. [15]	Deep Features	CNN	32 Species, Total 50 Training, 10 Test Samples of each species from total 1500 Images	0.9468
Jeon, Wang-Su et al. [16]	Deep Features	CNN	32 Species, Total 50 Training, 10 Test Samples of each species from total 1500 Images	0.8792
Wang, Zhaobin et al. [13]	Entropy Sequence + Ancillary Features	SVM	32 Species, Total 1600 Training Sample	0.9667
Haque, Farhana et al. [11]	Deep Features	CNN	10 Species, Total 50 Training Sample	0.94
Sharma, Prerna, et al. [6]	90 Features Using HOG	ANN	18 Species, Total 1900 Training Sample	0.97

As, unlike other considered models, model [15] provides a graphical view of test accuracy and loss function value over iterations, we compare test accuracy and loss function value over iterations of our model with model [15].
For this comparison, we divide dataset [9] into two parts, 1586 images for training and 320 images for testing, same as [15]. **Fig. 7** shows a comparative graphical view of our model with [15]. **Fig. 7(a)** shows a comparative graph of test accuracy over a number of iterations. **Fig. 7(b)** shows a

comparative graph of training loss function over a number of iterations. **Fig. 7** clearly explains that our model outperforms [15] in both the case of test accuracy and training loss function. Also, our model's test accuracy reaches up to 0.99% and training loss function decreases down to 0.021 (approx.) within less than 300 iterations which is more efficient compared to 12000 of model [15].

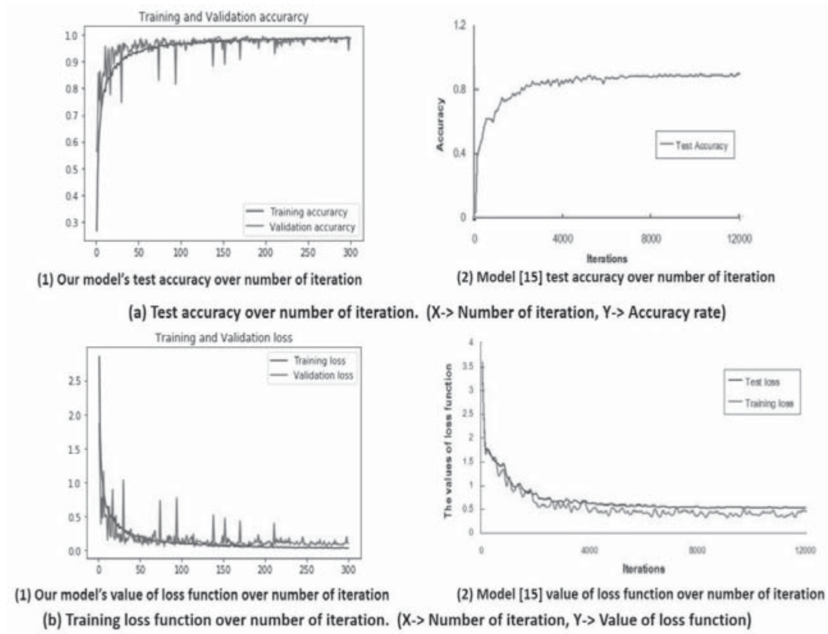


Fig. 7. Comparative graphical view of our model with [15]

4.2.1 Comparative analysis using dataset [10]

We perform a comparative analysis of our model's performance with some existing models who use dataset [10] for model evaluation. Here we performed the analysis using images from dataset [10] MalayaKew's whole leaf images (e.g., D1). Like other models [5] and [14], from the leaf images of 44 species of [10], 528 leaf images (i.e. 12 for each class) are selected as test set and 2288 leaf images (i.g. 52 for each class) are selected as training set. Table 3 shows the test accuracy of our model compared with existing models.

Table 3: The performance of our proposed model on D1 set (whole leaf image) of dataset [10]

Model	Features	Classifier	Accuracy
Our model	Hand-crafted Features + Deep Features	CNN	**0.981**
Lee, Sue Han, et al [5, 14]	Deep Features	CNN	0.977

Our proposed model uses efficient hand-crafted features which helps the CNN model to learn insightful deep features about different leaf species. Finally, our model, with these hand-crafted features combined with deep features achieves accuracy 99.64% on dataset [9] and 98.13% on dataset [10].

5. Conclusion

In this paper, we proposed a novel CNN based plant leaf identification model where we create five hand-crafted featured conversion images from each input image. After that we feed those images to our CNN network, followed by data augmentation. Produced features describe different leaf species more accurately and substantially makes our proposed CNN model's learning much efficient and enhances the performance of classification. Compared with other existing leaf classification systems our proposed model provides better performance. The highest accuracy of our model is 99.64% on Flavia dataset [9] over 32 species of leaf and 98.13% on MalayaKew dataset [10] with 44 leaf species. In future, we look forward to test our model on different object and improve the performance.

References

[1] Kumar, Thallapally Pradeep, M. Veera Prasad Reddy, and Prabin Kumar Bora. "Leaf identification using shape and texture features", *Proceedings of international conference on computer vision and image processing*. Springer, Singapore, 2017.

[2] A. Kadir, L. E. Nugroho, A. Susanto, and P. I. Santosa, "Leaf classification using shape, color, and texture features", preprint arXiv:1401.4447, 2013.

[3] T. Beghin, J. S. Cope, P. Remagnino, and S. Barman, "Shape and texture based plant leaf classification", *Advanced Concepts for Intelligent Vision Systems*, pp. 345–353, 2010.

[4] Kalyoncu, Cem, and Ö. Toygar. "Geometric leaf classification", *Computer Vision and Image Understanding* 133, pp. 102-109, 2015.

[5] Lee, S. Han, C. S. Chan, P. Wilkin and P. Remagnino. "Deep-plant: Plant identification with convolutional neural network", *IEEE International Conference on Image Processing (ICIP)*, pp. 452-456, 2015.

[6] Sharma, Prerna, A. Aggarwal, A. Gupta, and A. Garg. "Leaf identification using HOG, KNN, and neural networks", *International Conference on Innovative Computing and Communications*, pp. 83-91. Springer, Singapore, 2019.

[7] Overfitting and underfitting with machine learning algorithms. Machine Learning Mastery, Available link: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/, Last date of access (April 21, 2019).

[8] A. Torralba and A. A. Efros, "Unbiased look at dataset bias", in CVPR, vol. 1, no. 2, pp. 1521–1528, 2011.

[9] Wu, S. Gang, F. S. Bao, E. Y. Xu, Y. Wang, Y. Chang, and Q. Xiang. "A leaf recognition algorithm for plant classification using probabilistic neural network", *IEEE international symposium on signal processing and information technology*, pp. 11-16, 2007.

[10] Lee, S. Han, C. S. Chan, S. J. Mayo, and P. Remagnino. "How deep learning extracts and learns leaf features for plant classification." *Pattern Recognition*, vol. 71, pp. 1-13, 2017. Dataset: http://web.fsktm.um.edu.my/~cschan/downloads_MKLeaf_dataset.html.

[11] Haque, Farhana, and S. Haque. "Plant recognition system using leaf shape features and minimum Euclidean distance", *ICTACT Journal on Image & Video Processing*, vol. 9, no. 2, 2018.

[12] Arafat, S. Yasser, M. I. Saghir, M. Ishtiaq and U. Bashir. "Comparison of techniques for leaf classification", *Sixth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, pp. 136-141, 2016.

[13] Wang, Zhaobin, X. Sun, Y. Zhang, Z. Ying, and Y. Ma. "Leaf recognition based on PCNN", *Neural Computing and Applications* vol. 27, no. 4, pp. 899-908, 2016.

[14] Lee, S. Han, C. S. Chan, S. J. Mayo and P. Remagnino. "How deep learning extracts and learns leaf features for plant classification", *Pattern Recognition*, vol. 71, pp. 1-13, 2017.

[15] Z. Chaoyun, P. Zhou, C. Li and L. Liu. "A convolutional neural network for leaves recognition using data augmentation", *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 2143-2150, 2015.

[16] Jeon, W.Su, and S.Y. Rhee. "Plant leaf recognition using a convolution neural network", *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 1,, pp. 26-34, 2017.
[17] A. Pulkit, R. Girshick, and J. Malik. “Analyzing the performance of multilayer neural networks for object recognition”, *European conference on computer vision*, pp. 329-344. Springer, Cham, 2014.
[18] A. Sabu and K. Sreekumar, "Literature review of image features and classifiers used in leaf based plant recognition through image analysis approach", *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, pp. 145-149, 2017.
[19] Harris, G. Christopher, and M. Stephens. “A combined corner and edge detector”, *Alvey vision conference*, vol. 15, no. 50, pp. 10-5244. 1988.
[20] Sinha, Utkarsh. “Harris corner detector”, AI Shack, Online Available link: https://aishack.in/tutorials/harris-corner-detector/, Last date of access (May 2, 2019).
[21] Rathore, V. Singh, M. S. Kumar and A. Verma. "Colour based image segmentation using L* a* b* colour space based on genetic algorithm", *IJETAE* 2.6, pp. 156-162, 2012.
[22] G. Shruthi, and R. Krishna. "Image reconstruction using discrete wavelet transform", IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) 2.4, pp. 14-20, 2013.
[23] P. O. A. Thresholding, “Adaptive thresholdings”, 2003.
[24] S. F. BogoToBogo K Hong Ph.D.Golden Gate Ave, “Image thresholding and segmentation.” Available link: https://www.bogotobogo.com/python/OpenCV_Python/python _opencv3_Image _Global_Thresh olding_Adaptive_Thresholding_Otsus_Binarization_Segmentations.php, Last date of access (May 2, 2019).
[25] Murthy and Krishna. “Gabor filters: A practical overview”, Computer Vision Tutorials. Available link: https://cvtuts.wordpress.com/2014/04/27/gabor-filters-a-practical-overview/, Last date of access (May 10, 2019).
[26] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
[27] Image Preprocessing – Image Data Generator Class - Keras Documentation, Available link: https://keras.io/preprocessing/image/, Last date of access (May 10, 2019).
[28] A gentle introduction to *k*-Fold cross-validation. *Machine Learning Mastery*, Available link: https://machinelearningmastery.com/k-fold-cross-validation/, Last date of access (May 8, 2019).

Authors Biography

Rafflesia Khan is a student of M.Sc. at Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh. She has completed her Bachelor’s degree from Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh in 2017. Her research areas of interest are image processing, visual object detection & recognition, machine learning, pattern recognition, and internet of things security.

Rameswar Debnath is a Professor of Computer Science and Engineering Discipline at Khulna University, Khulna, Bangladesh. He has completed his Bachelor degree from Computer Science and Engineering discipline, Khulna University, Khulna, Bangladesh in 1997. He has received his Masters in Engineering degree and Ph.D. degree in Computer Science and Engineering from the University of Electro-Communications, Tokyo, Japan in 2002 and 2005 respectively. His research areas of interest are image processing, statistical machine learning and its applications to pattern recognition, visual object detection and natural language processing.