International Journal on Emerging Technologies 11(3): 904-910(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

# Recognition of Plant Species using Deep Convolutional Feature Extraction

***Nguyen Van Hieu*[1] *and Ngo Le Huy Hien*[2]**
[1]*Doctor, Department of Information and Technology,*
*The University of Danang - University of Science and Technology, Danang, Vietnam.*
[2]*Undergraduate Researcher, Department of Computer Science and Engineering,*
*The University of Danang - VN-UK Institute for Research and Executive Education, Danang, Vietnam.*

*(Corresponding author: Ngo Le Huy Hien)*
*(Received 13 March 2020, Revised 14 May 2020, Accepted 16 May 2020)*
*(Published by Research Trend, Website: www.researchtrend.net)*

**ABSTRACT: There are more than 391,000 plant species currently known to global science, and it is challenging to distinguish among them. The identification of plant species requires in-depth surveyors and botanists who possess a tremendous amount of knowledge on native plant species. Therefore, plant recognition has become an interdisciplinary concentration in both botanical taxonomy and machine learning for a faster identification process. In this paper, a convolutional neural network system has been proposed to perform feature extraction using different deep learning models in large-scale plant classification methods. The plant image dataset was collected from the PlantCLEF2003 dataset, which consists of 51,273 images from 609 plant species. Four deep convolutional feature extraction methods, including Resnet50V2, Inception Resnet V2, MobilenetV2, and VGG16, are used to extract features from the images. A comparative evaluation of four deep learning models using two classification methods, Support Vector Machine (SVN) and k-nearest neighbor (KNN), is presented. With the highest accuracy of 95.6%, MobilenetV2 performed better than the other deep learning models for plant recognition in both SVM and KNN classification methods. Moreover, the SVM classifier has outperformed the KNN in terms of accuracy in the plant image recognition system. The outcomes are promising for further applications and future work gears towards experiments on a larger dataset with high-performance computing facilities to propose a higher accuracy system of plant image identification in natural environments.**

**Keywords:** Convolutional Neural Network, Deep features,Deep learning, K-nearest neighbors, Plant identification, Support Vector Machine.

**Abbreviations:** CNN, convolutional neural network; KDES, kernel descriptor; PCA, principal component analysis; PNN, probabilistic neural network; SVM, support vector machine.

## I. INTRODUCTION

There is a high demand for automated plant recognition systems that assist users without in-depth knowledge and specialized skills in botany and plant systematics to identify and lookup for plant species through natural photographs [37]. Machine learning aided plant recognition systems are promising solutions towards bridging the botanical taxonomic gap, which has been considerably developed in both botany and computer communities [12]. By machine learning technology advances, sophisticated models have been proposed, which serve for the plant image recognition and retrieval [5, 9, 11]. And in recent years, improving the performance of the systems draw massive attention from global researchers and engineers in the field of machine learning [37].

Many research authors have conducted studies on the development of tools for the identification of plants and accomplished specific outcomes over the last ten years. In the beginning periods, leaves are most commonly used for plant identification among the researchers, by utilizing low-level features, including shape, color, and texture [1-4]. Wu *et al.* [5] have implemented one of the most authoritative works in the field of plant classification. Their system constructs twelve morphological features derived from five basic geometric features and then reduces the dimension by Principal Component Analysis (PCA) in order to send fewer inputs to a probabilistic neural network (PNN). This system achieved 90.3% of average accuracy on the Flavia dataset, which is their own creation. Using the same dataset, many researchers have followed and developed different techniques on plant identification and obtained certain results [6-8]. The best result published so far on this Flavia dataset gave an accuracy of 98.5%, which was developed by Le *et al.* (2014) [9]. They applied a new feature extraction technique with the kernel descriptor (KDES) to build a fully automated plant identification system. The proposed system also achieved a high accuracy of 98.3% by testing on a dataset that consists of 55 Vietnamese medicinal plants with the Support Vector Machine (SVM) classifier [9]. Using a fusion of fuzzy local binary pattern and fuzzy color histogram and a PNN classifier, Herdiyeni and Wahyuni (2012) have tested their system on a dataset of 2448 leaf images (270 *240 pixels) collected from herbal plants in Indonesian forests to accomplish a classification accuracy of 74.5% [10]. In [11], Arai *et al.* (2013) attained 95.8% accuracy by using the SVM classifier and the discrete wavelet transform to extract translation-invariant features from a dataset of 8 different ornamental plants in Indonesia, with the size of each image was 256*256 pixels [11]. In [12], Yu *et al.* have acquired the BJFU100 dataset by mobile phone in the natural environment, which consists of 10,000 images from 100 ornamental plant species growing in Beijing Forestry University campus. A 26-layer deep learning model with 8 residual building blocks is created for uncontrolled plant identification. They have proposed a model with a recognition rate of 91.78% on the BJFU100 dataset.

Using Plant CLEF, which is the dataset of the Pl@net project [13], a number of researchers have theoretically and practically proposed systems for the identification of plants [14-15]. Josef *et al.* [16] have implemented experiments to test three network architectures, including Inception v3, ResNet50, and DenseNet 201, on a clean dataset of 256,288 samples of 10,000 different plant species. With an accuracy of above 90%, they realized that DenseNet performed better than the rest models. By using PlantCLEF 2015 and PlantCLEF 2017 dataset, Danzi *et al.* [17] proposed a loss function that encodes the hierarchical relationships of the taxonomic tree into the deep learning objective function. This is a promising model for classification tasks with multilevel labels.

Continuing to the development of deep learning breakthroughs in image recognition, the project used a part of the PlantCLEF2003 dataset, which is collected from numerous global volunteers, containing 51,273 environmental images of 609 plant species. The authors aimed to propose high-efficiency models for plant image recognition in large-scale plant classification methods. In this research, four pre-trained deep learning models for plant image feature extraction were implemented, including Resnet50V2, Inception ResnetV2, MobilenetV2, and VGG16, together with Support Vector Machine (SVM) and K-nearest neighbors (KNN) classification methods. A comparative evaluation of four deep learning models using two classification methods is presented. The proposed models achieve a recognition rate of 95.6% in the best case on our collected dataset. The result is a state-of-the-art solution for choosing a deep learning method suitable for classification methods, contributing to the development of plant image recognition systems.

## II. PLANT IMAGE RECOGNITION

*A. Deep Convolutional Feature Extraction Models*

*1. Resnet50V2*

Residual Neural Networks (ResNets) [18] consist of many stacked “Residual Units”. The general form of each unit can be expressed as:

$$\mathbf{y}_l = h(\mathbf{x}_l) + A(\mathbf{x}_l, W_l), \ \mathbf{x}_{l+1} = f(\mathbf{y}_l)$$

where $\mathbf{x}_l$ and $\mathbf{x}_{l+1}$ are input and output of the $l$-th unit, and A is a residual function. In [18], $h(\mathbf{x}_l) = \mathbf{x}_l$ is an identity mapping and $f$ is a ReLU [19] function.

ResNets that have more than 100-layer deep have indicated cutting-edge accuracy for some difficult recognition tasks [20]. With an error rate of 3.57%, Resnets is the winner in both ImageNet ILSVRC [21] and MS COCO [22] competitions in 2015. The ResNets core idea is to find out the additive residual function A with regard to $h(\mathbf{x}_l)$, and using an identity mapping $h(\mathbf{x}_l) = \mathbf{x}_l$ as a key choice. This is realized by appending a "shortcut" (identity skip connection).

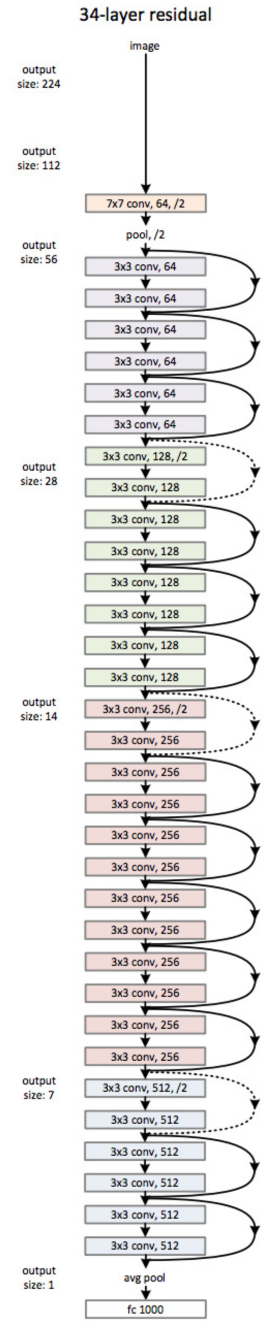


**Fig. 1.** A 34-parameter-layer residual network (3.6 billion FLOPs) [19].

Overall, the design of a 34-layer residual network is demonstrated in Fig. 1. The dotted skip connections represent multiplying the identity mapping by the Ws linear projection term to align the dimensions of the inputs [23].

ResNet v2 is the second version of ResNet, which was released by the second paper on ResNet. The dominant improvement of Resnet V2 is the arrangement of the layers in the residual block, as indicated in Fig. 2. The important changes in ResNet v2 are: using a stack of 1 × 1 - 3 × 3 - 1 × 1 BN-ReLU-Conv2D, the Batch normalization, and ReLU activation that comes before 2D convolution [19].

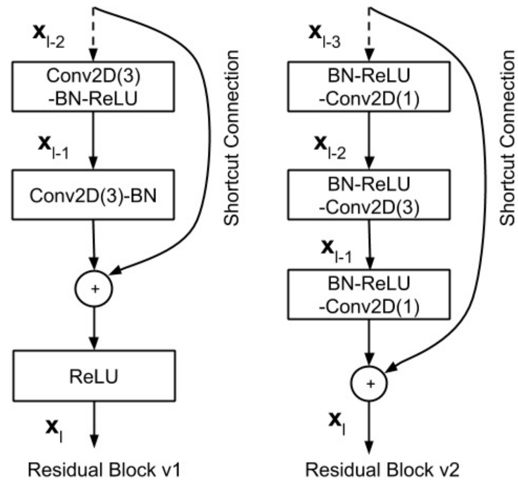


**Fig. 2.** A comparison between ResNet v1 and ResNet v2 on residual blocks [23].

*2. InceptionResnetV2*

Inception-ResNet is a hybrid of the Inception net and Residual net. Inception-ResNet-v2 [24] merges both the concepts of Inception-v3 and ResNet architectures. Inception-ResNet-v2 consists of the same stem as Inception-v3 and the same residual blocks as ResNet model. However, inside every residual block, filter concatenation is carried out, and their filter size varies for these residual blocks.

The inception-ResNet-v2 model makes use of residual connections to inception modules after the implementation of the stem. This permits Inception-ResNet-v2 to attain higher accuracies in a shorter time frame, whereas it has a similar computational expensive as in the Inception-v4. For residual connections to work, residual connections replaced pooling layers in pure Inception modules. Like Inception-v4, the Inception-ResNet-v2 model has the same structures for the classification part, which consists of an average pooling layer, a dropout layer, and a fully-connected layer that returns the softmax probabilities over predicted output classes [24].

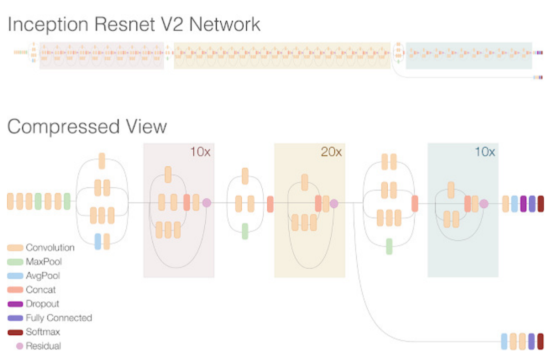


**Fig. 3.** Inception-ResNet-v2 schematic diagram [25].

*3. MobilenetV2*

An expanded network of the Inception-ResNet-v2 is illustrated at the top of Fig. 3, which is substantially deeper than Inception V3. In the below figure, the detailed version of the same network is indicated, where repeated residual blocks were compressed. It can be seen that the inception blocks were simplified, with fewer parallel towers than Inception V3. Regarding the accuracy, the Inception-ResNet-v2 outperformed than previous models, including Inception V3, ResNet V2 200, and ResNet 152 [26].

Mobilenet is a model often used in mobile environments because of its compact nature and in terms of accuracy and speed. MobileNet supports the classification and detection that can run deep networks on mobile devices. This improves user experience, energy consumption, security, and privacy. In the emerging development of new applications, the demand for more capable neural networks is growing, and Mobilenet allows users to interact in real-time with the real world.

Whereas having the same accuracy throughout the whole latency spectrum, MobileNetV2 models are faster compared to MobileNet V1. Especially, MobileNetV2 models use 30% fewer parameters, 2 times fewer operations, and are about 30% to 40% faster on Google Pixel phones than MobileNetV1, while accomplishing higher accuracy. MobileNetV2 is a very compelling feature extractor for object detection and segmentation [27].

The MobileNetV2 architecture [27], as illustrated in Fig. 3, is based on the original MobileNet. This model uses 3×3 depth-wise separable convolutions, together with an inverted residual structure in shortcut connections between thin bottleneck layers in order to reduce the size of input and output. At the time for the evaluation of ImageNet, this model outperformed the state-of-the-art networks such as MobileNet and ShuffleNet.

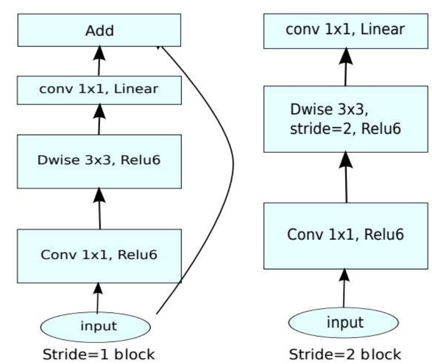


**Fig. 4.** MobileNetV2 depth-wise convolution [27].

*4. VGG16*

This deep learning method is one of the first attempts at adding depth to improve classification accuracy. VGG16 is a CNN architecture that was used to win the ImageNet ILSVR competition 2014. It is as yet considered as one of the outstanding vision model architecture. The most unique thing about VGG16 is that instead of having a large number of hyper-parameters, it uses convolution layers of 3 x 3 filter of stride 1 and the same padding and max pool layer of 2x2 filter of stride 2. As shown in Fig. 5, it follows this convolution arrangement and max pool layers through the whole architecture consistently. As the output, two fully connected layers are created, followed by a SoftMax. The 16 in VGG16 implies 16 layers that have weights. This network contains approximately 138 million parameters.

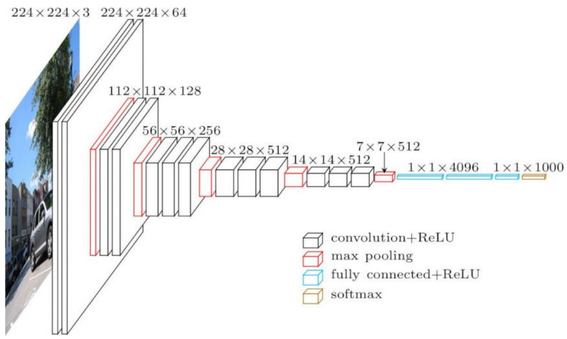


**Fig. 5.** The architecture of VGG16 [28].

*B. Classification Methods*

In this research, the Support Vector Machine (SVM) and k-nearest neighbors (KNN) were used in the plant image classification system. As these classifiers are supervised machine learning algorithms, which commonly used for classification to optimize classifying accuracy with fully connected layers, these algorithms are fit for such a large feature set generated by convolutional neural networks. Moreover, SVM and KNN possess outstanding generalization capability and reputation in the training data set to achieve high accuracy. These methods are based on the structural risk minimization principle and statistical learning theory.

*1. Support Vector Machine*

As a state-of-the-art classifier, SVM has been widely used in many classification applications of input samples [29, 30]. Let $\{(x_i, y_i)\}_{i=1}^{N}$ be a set of N training samples, where $x_i$ is the $i^{th}$ sample in the input space x, and $y_i \in \{+1, -1\}$ is the class of $x_i$ label. The decision function of SVM that classifies a new test sample $x$ can be represented as

$$f(z) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i k(x_i, z) + b\right)$$

where $z$ is an unclassified sample, $\alpha_i$ is the Lagrange multiplierof a dual optimization problem that describes the separating hyperplane; $k(\cdot,\ \cdot)$ denotes the kernel function which should satisfy Mercer's condition; and $b$ is the hyperplane threshold parameter [31]. The training sample $x_i$ (with $\alpha_i > 0$) is called support vectors, and the SVM classifier finds the optimal hyperplane that maximizes the separating margin between two classes, as shown in Fig. 6 [32].

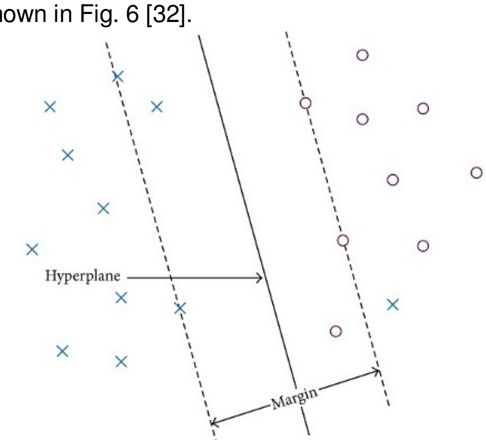


**Fig. 6.** Optimal hyperplane of SVM in non-separable cases [32].

*2. K-nearest neighbors*

K-nearest neighbor is the most widely used clustering algorithm and classification. The k-nearest neighbors (kNN) principle is that an instance is classified by a majority vote of its neighbors [33]. It provides a simple and intuitive rule for pattern discrimination, which has resulted in its extensive use in a variety of applications and gains a high classification rate [34, 35].

**Input**: Let T be a set of feature vectors of training images, x is a feature vector of the image used to test, *L* is a set of class labels used to assign to x.
**Output**: the class label of x ($c_x \in L$)
**for** each a ∈ $T$**do**
Compute *d(x, a)*, the distance of x and a;
**end**
Select the set S ⊆ T of *k* nearest feature vectors from x;

$$c_x = \operatorname*{argmax}_{y \in L} \sum_{a \in S} I(y = class(c_a))$$

where the indicator function *I*(.) returns the value 0 if its argument is false and 1 otherwise.

## III. EXPERIMENTAL RESULTS

*A. Dataset Collection*

The plant image dataset was collected from the PlantCLEF2003 dataset, which is a part of the Pl@net project. The final training dataset is composed of 51,273 environmental images of 609 plant species. The training dataset has a total of 609 classes, corresponding to 609 species, with an average of 84 images each class. The number of images for each class ranges from 7 to 234. In Table 1, the detailed statistical data is presented, and the distribution of plant species is shown in Fig. 7.

**Table 1: Statistical data of the training dataset.**

| Total amount | Average | Median | Max | Min |
|---|---|---|---|---|
| 51273 | 84 | 91 | 234 | 7 |

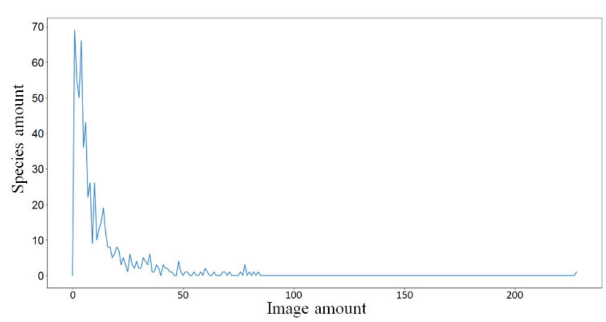


**Fig. 7.** Distribution of image amount for plant species in the training dataset.

After training, different deep learning models and classification methods have been tested on a test dataset of 1071 images.

*B. Experiments*

This project's purpose is to construct a convolutional neural network system to perform feature extraction using different deep learning models in large-scale plant classification methods. Our system was implemented in Keras using TensorFlow backend on a computer equipped with CPU Intel Core (™) i5 processor, 8GB RAM, and GPU GTX1070Ti.

For better classification results, the work was divided into two steps. Firstly, deep learning models were built to distinguish plant species through embedding matrices. There are many ways to distinguish and dissect the characteristic matrix, and in this paper, four different pre-trained deep learning models were used, including Resnet50V2, Inception Resnet V2, MobilenetV2, and VGG16. The authors have utilized and trained deep learning model parameters using the triplet loss function [36], which used three inputs called anchors, positives, and negatives. The positives are images that have the same class as the anchors, and otherwise, the negatives have a different class. Secondly, the classes were classified with separated matrix by 2 classification methods, SVM and KNN.

The target is to evaluate the efficiency of different deep convolutional features with different classifiers. Fig. 8 to 11 have indicated the embedding models using VGG16, MobilenetV2, InceptionResnetV2, and ResnetV2, respectively. To compare all methods used in this research, F-Score was also used to indicate the precision and recall, which is calculated as below.

$$F_1 = \left(\frac{recall^{-1} + precision^{-1}}{2}\right)^{-1} = 2.\frac{precision\,.\,recall}{precision + recall}$$

In the above equation, the Precision is the proportion of positive results that truly are positive and the Recall is the ability of a test to correctly identify positive results to get the true positive rate. The F score reaches the best value at a value of 1, meaning perfect precision and recall. The worst F score, which means the lowest precision and lowest recall, would be a value of 0.

Table 2 and Fig. 12 compare the accuracy and F-Score of different deep learning models using different classifiers. It is noticeable that VGG16 causes the overfitting issue, leading to low efficiency. Notably, it also leads to a training failure when working with the SVM classifier. Although the InceptionResnetV2 has average accuracy, its training and evaluation time is much longer than other models. With a unique structure, ResnetV2 also has a relatively low loss. Despite the compactness, the MobilenetV2 reaches the highest accuracy, making it a very suitable model for running on online servers. By experiment, it is realized that by using SVM with linear activation, some embedding models reach a quite high accuracy, ranging from about 67.8% to 78.0%. Although KNN takes a longer time to access memory, which is not suitable for an online operation model, KNN has a lower accuracy than the SVM classification method overall. In the best case, the plant identification model reached the highest accuracy of 95.6% on MobilenetV2 model with the SVM classifier. For general evaluation, the results are illustrated in Table 2, which shows that the MobilenetV2 model outperformed the other deep learning models for plant recognition in both SVM and KNN classification methods.

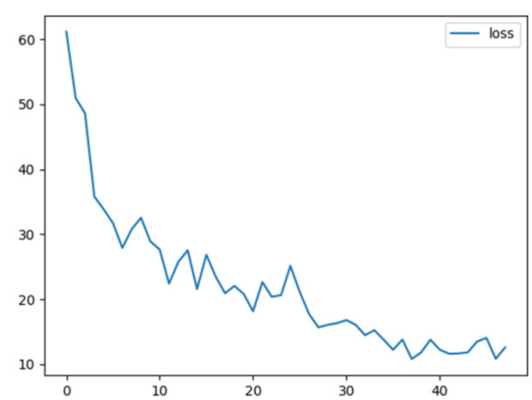


**Fig. 8.** Embedding model using Resnet50V2 with loss: 7.8.

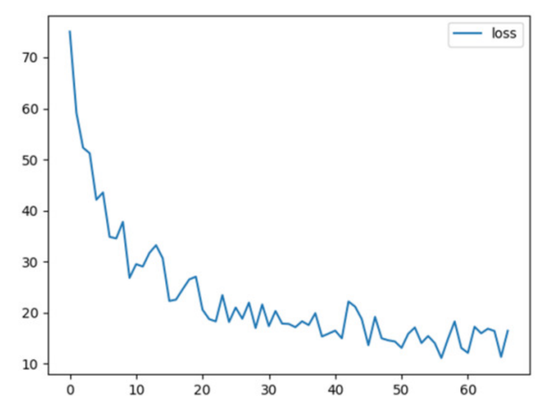


**Fig. 9.** Embedding model using InceptionResnetV2 with loss: 10.8.

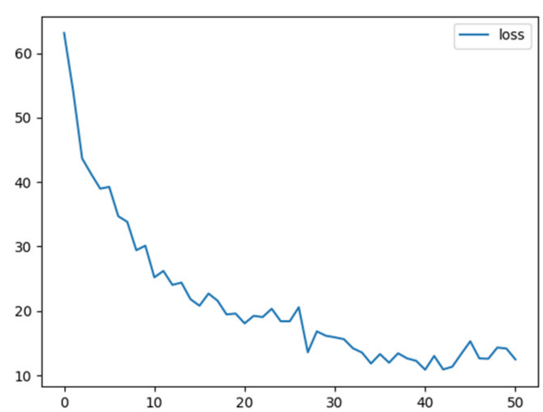


**Fig. 10.** Embedding model using MobilenetV2 with loss: 12.4.

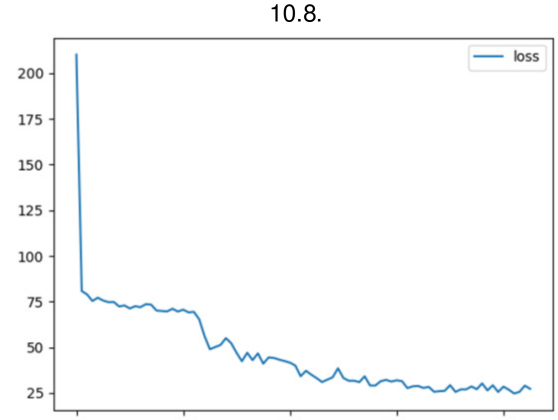


**Fig. 11.** Embedding model using VGG16 with loss: 26.2.

**Table 2: Comparison of two classification methods using four different deep learning models.**

| Classification Method | Deep Learning Model | Training loss | Training time (minutes) | On 50 classes Accuracy (%) | On 50 classes F-Score | On 100 classes Accuracy (%) | On 100 classes F-Score | On 200 classes Accuracy (%) | On 200 classes F-Score | Execution time (seconds) |
|---|---|---|---|---|---|---|---|---|---|---|
| Support Vector Machine (SVM) | VGG16 | 26.2 | 16.7 | Failure | 0.06 | Failure | 0.07 | Failure | 0.05 | 8.70 |
| | MobilenetV2 | 12.4 | 32.4 | 77.4 | 0.92 | 78.8 | 0.90 | 95.6 | 0.89 | 7.41 |
| | InceptionResnetV2 | 10.8 | 23.1 | 80.3 | 0.73 | 76.5 | 0.71 | 90.9 | 0.67 | 36.80 |
| | ResnetV2 | 7.8 | 33.2 | 71.0 | 0.86 | 69.0 | 0.81 | 75.5 | 0.82 | 13.00 |
| K-nearest neighbors (KNN) | VGG16 | 26.2 | 29.9 | 62.5 | 0.85 | 57.6 | 0.75 | 72.8 | 0.61 | 8.70 |
| | MobilenetV2 | 12.4 | 42.3 | 68.1 | 0.89 | 77.8 | 0.88 | 78.0 | 0.88 | 7.41 |
| | InceptionResnetV2 | 10.8 | 27.8 | 59.2 | 0.85 | 64.5 | 0.84 | 67.8 | 0.78 | 36.80 |
| | ResnetV2 | 7.8 | 36.2 | 71.4 | 0.82 | 71.1 | 0.85 | 74.2 | 0.80 | 13.00 |

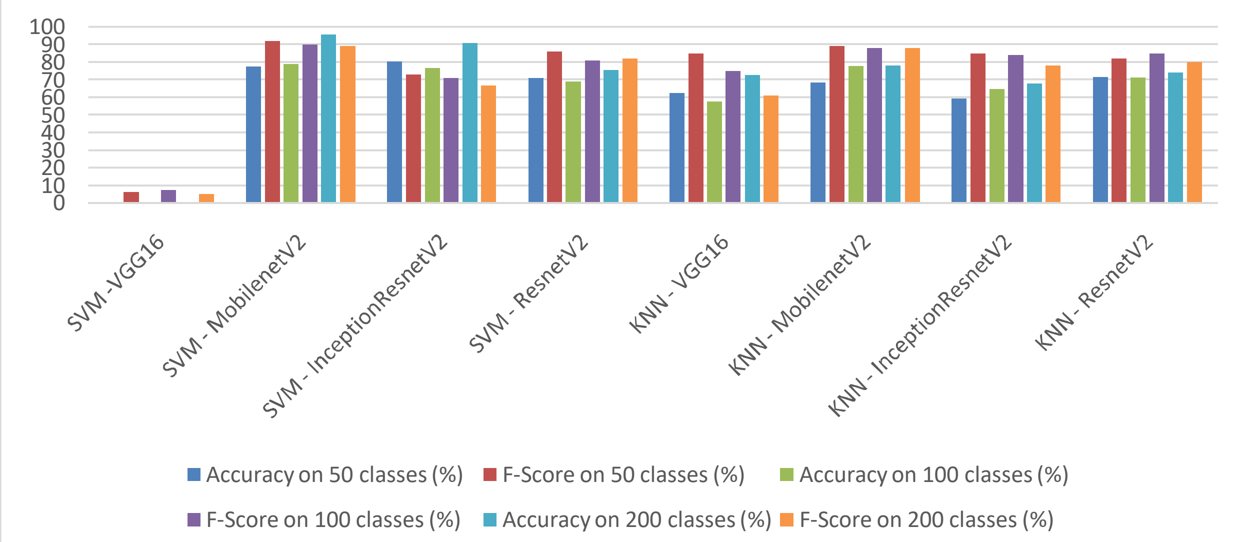


**Fig. 12.** Comparative chart on Accuracy and F-Score of different deep learning models using different classifiers.

## IV. CONCLUSION

This project's purpose is to construct a convolutional neural network system to perform feature extraction using different deep learning models in large-scale plant classification methods. In this paper, an evaluation of the performance of pre-trained deep convolutional feature extraction methods with different classifiers was conducted to identify plant species. Through experiments, it is concluded that MobilenetV2 is outperformed than the other deep learning models, including Resnet50V2, InceptionResnetV2, MobilenetV2, and VGG16, in both SVM and KNN classification methods. Moreover, the SVM classifier has outperformed the KNN in terms of accuracy in the plant image recognition system. The MobilenetV2 also attained the highest accuracy of 95.6% and the highest F-Score of 0.92 in plant identification, which is a promising result for future work. Furthermore, the MobilenetV2 model assists plant identification systems in applying in the real world as it outputs not only high accuracy but also the compactness in the application process.

## V. FUTURE SCOPE

The outcomes of this study open up new avenues for future research and can serve as a hypothetical source for future plant identification systems. Although substantial efforts have been made in the past (Danzi W. *et al.* 2019 [17], Josef H. *et al.* 2018 [16], and Sophia *et al.* 2019 [25]), our research proposed high-efficiency models for plant image recognition in large-scale plant classification methods. In future research, attempt gears towards using more classification models, rather than SVM and KNN to improve the performance of the model. In order to reach a higher accuracy, a comparison between state-of-the-art classification models is needed to upgrade the current work. Future work will also focus on using a larger dataset and high-performance computing facilities to investigate a higher performance of plant image identification in the natural environment.

**Conflict of Interest.** No conflict of interest occurred as the study is based on comprehensive literature reviews and expert hypotheses to develop the scale of plant image recognition systems.

## REFERENCES


[1]. Kumar, N. (2012). Leaf snap: A Computer vision system for automatic plant species identification. *Proc. Computer Vision – ECCV, Springer*, 502–516.

[2]. Du, M., Zhang, S., and Wang, H., (2009). Supervised Isomap for Plant Leaf Image Classification.*5th International Conference on Emerging Intelligent Computing Technology and Applications*, 627-634.

[3]. Hossain, J., and Amin, M.A., (2010). Leaf Shape Identification Based Plant Biometrics.*13th International Conference on Computer and Information Technology*, 458-463.

[4]. Du, J.X., Wang, X.F., and Zhang, G.J., (2007). Leaf shape-based plant species recognition. *Applied Mathematics and Computation*, 185.

[5]. Wu, S.G. *et al.* (2007). A Leaf Recognition Algorithm for Plant Classification using Probabilistic Neural Network.*7th IEEE International Symposium on Signal Processing and Information Technology*, 11-16.

[6]. Hernandez-Serna, A., and Jimenez-Segura, L.F., (2014). Automatic Identification of species with neural networks.*Peer J.*, 563.

[7]. Chaki, J., Parekh, R., and Bhattacharya, S., (2015). Plant leaf recognition using texture and shape features with neural classifiers. *Pattern Recognition Letters*, Vol. 58: 61-68.

[8]. Siravenha, A.C.Q., and Carvalho, S.R., (2015). Exploring the use of Leaf Shape Frequencies for Plant Classification.*28th SIBGRAPI Conference on Graphics, Patterns and Images*, 297-304.

[9]. Le, T.L., Tran, D.T., and Hoang, V.N., (2014). Fully Automatic leaf-based plant identification, application for Vietnamese medicinal plant search. *Fifth Symposium on Information and Communication Technology*, 146-154.

[10]. Herdiyeni, Y., and Wahyuni, N.K.S., (2012). Mobile Application for Indonesian Medicinal Plants Identification using Fuzzy Local Binary Pattern and Fuzzy Color Histogram. *International Conference on Advanced Computer Science and Information Systems (ICACSIS),* 301-306.

[11]. Arai, K., Abdullah, I.N., and Okumura, H., (2013). Identification of Ornamental Plant Functioned as Medicinal Plant Based on Redundant Discrete Wavelet Transformation. *International Journal of Advanced Research in Artificial Intelligence*, 60-64.

[12]. Yu, S., Yuan, L., Guan, W., and Haiyan, Z., (2017). Deep Learning for Plant Identification in Natural Environment. *Computational Intelligence and Neuroscience*.

[13]. Clough, P.D., and Sanderson, M., (2003). The CLEF 2003 Cross Language Image Retrieval Track, in press.*Cross Language Evaluation Forum (CLEF)*.

[14]. Hervé, G., Pierre, B., and Alexis, J., (2016). Plant Identification in an Open-world (LifeCLEF 2016).*CLEF: Conference and Labs of the Evaluation Forum*, 428-439.

[15]. Alexandru, T., Liviu, D., and Bogdan, I., (2017). Automatic Plant Image Identification using Transfer Learning via Convolutional Neural Networks. *CEUR Workshop Proceedings,* Vol. 1866.

[16]. Josef, H., Stefan, K., Danny, K., and Maximilian, E., (2018). Large-Scale Plant Classification using Deep Convolutional Neural Networks. *CEUR Workshop Proceedings*.

[17]. Danzi, W. *et al.,* (2019). Deep Learning with Taxonomic Loss for Plant Identification. *Computational Intelligence and Neuroscience*.

[18]. He, K., Zhang, X., Ren, S., and Sun, J., (2016). Deep residual learning for image recognition.*CVPR*.

[19]. Packt, (2020). ResNet v2.*Big Data and Business Intelligence,* https://subscription.packtpub.com/.

[20]. Russakovsky, O. (2015). ImageNet Large Scale Visual Recognition Challenge. *IJCV*.

[21]. Nair, V., Hinton, G.E., (2010). Rectified linear units improve restricted boltzmann machines, *ICML.*

[22]. Lin, T.Y., (2014).Microsoft COCO: Common objects in context, *ECCV*.

[23]. Rowel Atienza, (2018). Advanced Deep Learning with Keras. 59-63.

[24]. Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A.A., (2017). Inception-v4, inception-resnet and the impact of residual connections on learning, *AAA*,.

[25]. Sophia, C., Kiat, J.H., Teck, W.C., MD, Abdullah A., M., and Yang, L.C., (2019). Plant Identification on Amazonian and Guiana Shield Flora: NEUON, in press, *Life CLEF 2019 Plant*.

[26]. Nguyen Van Hieu, Ngo Le Huy Hien, (2020). Automatic Plant Image Identification of Vietnamese species using Deep Learning Models. *International Journal of Engineering Trends and Technology*, Vol.68(4): 25-31.

[27]. Mark, S., and Andrew, H., (2018). MobileNetV2: The Next Generation of On-Device Computer Vision Networks.*Google Research*.

[28]. Muneeb, ul H., (2018). VGG16 – Convolutional Network for Classification and Detection.*Neurohive*.

[29]. Agarap, A.F.,(2017). An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification.*arXiv 2017*, arXiv:1712.03541.

[30]. Kaiming, H., Xiangyu, Z., Shaoqing, R., and Jian, S., (2015). Deep Residual Learning for Image Recognition.*Microsoft Research*.

[31]. Ribeiro, B., (2005). Support vector machines for quality monitoring in a plastic injection molding process, *IEEE Transactions on Systems. Man and Cybernetics C.*, 401–410.

[32]. Song, Q, Hu, W, and Xie, W., (2002). Robust support vector machine with bullet hole image classification .*IEEE Transactions on Systems, Man and Cybernetics C.*, 440–448.

[33]. Ma, J., Sheridan, R.P., A., Dahl, G.E., and Svetnik, V., (2015). Deep neural nets as a method for quantitative structure-activity relationships. *J Chem Inf Model*, 263–274.

[34]. Remus, J.J., Morton, K.D., Torrione, P.A., Tantum, S.L., and Collins, L.M., (2018). Comparison of a distance-based likelihood ratio test and k-nearest neighbor classification methods. *IEEE workshop on machine learning for signal processing*, 362–367.

[35]. Fukunaga, K., and Hostetler, L.D., (1975). k-Nearest-neighbor Bayes-risk estimation.*IEEE Trans Inf Theory*, 285–93.

[36]. Marc-Olivier, A., (2018). Lossless Triplet loss.*Towards Data Science*.

[37]. Jana, W., Michael, R., Marco, S., and Patrick, M., (2018). Automated plant species identification—Trends and future directions. *PLoS Comput Biol.*



**How to cite this article:** Hieu, N. V. and Hien, N. L. H. (2020). Recognition of Plant Species using Deep Convolutional Feature Extraction. *International Journal on Emerging Technologies*, *11*(3): 904–910.