IET Image Processing

Research Article

DDLA: dual deep learning architecture for classification of plant species

ISSN 1751-9659
Received on 21st March 2019
Revised 17th June 2019
Accepted on 23rd July 2019
E-First on 25th September 2019
doi: 10.1049/iet-ipr.2019.0346
www.ietdl.org

Anubha Pearline Sundara Sobitha Raj[1] ✉, Sathiesh Kumar Vajravelu[1]
[1]Department of Electronics Engineering, Madras Institute of Technology, Anna University, Chennai, India
✉ *E-mail: anubhapearl@gmail.com*

Abstract: Plant species recognition is performed using a dual deep learning architecture (DDLA) approach. DDLA consists of MobileNet and DenseNet-121 architectures. The feature vectors obtained from individual architectures are concatenated to form a final feature vector. The extracted features are then classified using machine learning (ML) classifiers such as linear discriminant analysis, multinomial logistic regression (LR), Naive Bayes, classification and regression tree, *k*-nearest neighbour, random forest classifier, bagging classifier and multi-layer perceptron. The dataset considered in the studies is standard (Flavia, Folio, and Swedish Leaf) and custom collected (Leaf-12) dataset. The MobileNet and DenseNet-121 architectures are also used as a feature extractor and a classifier. It is observed that the DDLA architecture with LR classifier produced the highest accuracies of 98.71, 96.38, 99.41, and 99.39% for Flavia, Folio, Swedish leaf, and Leaf-12 datasets. The observed accuracy for DDLA + LR is higher compared with other approaches (DDLA + ML classifiers, MobileNet + ML classifiers, DenseNet-121 + ML classifiers, MobileNet + fully connected layer (FCL), DenseNet-121 + FCL). It is also observed that the DDLA architecture with LR classifier achieves higher accuracy in comparable computation time with other approaches.

1 Introduction

Plants play a major role in sustaining human life [1]. They are diverse (intra-class and inter-class variations) in nature and available in abundance. Traditional plant species identification requires a lot of manual effort [2]. Plant species or taxon identification is made easier through computer vision and pattern recognition methods. The plant parts such as flower, fruits, and leaves are used in plant species recognition. Aakif and Khan [3] suggest that the leaves are useful in the identification of plant species as they are available almost throughout the year.

Conventional computer vision techniques [1–12] and deep learning (DL) approaches [13–27] are utilised in the identification of plant species. The conventional computer vision method involves the pre-processing of images, extracting the handcrafted features, and classifying them using machine learning (ML) algorithms. DL method requires the images to be preprocessed and it also performs the process of both feature extraction and classification. Some of the reported accuracies for plant recognition using conventional computer vision approaches are 76.3 [2], 90 [4], and 91% [1]. The recognition rate is further improved by DL approaches. Liu and Kan [13] reported that the automatic identification of plant species using DL (accuracy of 93.9%) method performs better than traditional classification methods.

Convolutional neural network (CNN) [43] is a supervised DL architecture. It consists of convolution layers, pooling layers, and fully connected layers (FCLs). The introduction of AlexNet [28] CNN architecture opened the gateway to the development of new CNN models. Some of the DL CNN architectures are VGG-16, VGG-19 [29], Inception-v3 [30], Inception-ResNet-v2 [31], Xception [32], ResNet50 [33], DenseNet-121, DenseNet-169, DenseNet-201 [34] MobileNet [35] etc.

In this paper, the two CNN architectures, namely MobileNet and DenseNet, are combined to extract the features without their FCL. Later, the plant species recognition is performed by using ML classifiers. The classifiers used are linear discriminant analysis (LDA), multinomial logistic regression (LR), Naive Bayes (NB), classification and regression tree (CART), *k*-nearest neighbour (KNN), random forest classifier (RFC), bagging classifier (BC), and multi-layer perceptron (MLP) [36, 37]. Performance analysis is carried out for four datasets. Three benchmark datasets considered in the studies are Folio [6], Swedish leaf [7], and Flavia [4] datasets. The fourth dataset is our own developed dataset, collected under various lighting conditions and different colour backgrounds.

The organisation of this paper is as follows. Section 2 discusses the related work on plant species recognition. Section 3 describes the methodology for plant species recognition. Section 4 quotes the experimental analysis and the observation on plant species recognition. Section 5 summarises the results.

2 Related work

In this section, reported works on plant species recognition are discussed. Wang *et al.* performed plant species recognition by extracting the features (entropy sequence) using pulse-coupled neural network and classified using support vector machine (SVM). The approach of Wang *et al.* is assessed for three datasets, namely Flavia, Intelligent Computing Laboratory (ICL) and MEW2012. The reported accuracies for the above datasets are 96.67, 91.56, and 91.2% [5].

In the process of feature extraction, Liu *et al.* combined several texture and shape features to form a final feature vector. Texture features considered are local binary pattern, Gabor filters, and grey-level co-occurrence matrices. Hu moment invariants and Fourier descriptors are used as shape features. Improved deep belief networks with dropout are used for classification [13]. This method is tested with ICL leaf dataset and resulted in an accuracy of 93.9%.

Tan *et al.* proposed D-Leaf network, a CNN-based method for feature extraction. Extracted features are classified using several ML classifiers such as SVM, artificial neural network (ANN), KNN, NB, and CNN. Also, the results are evaluated for Sobel segmented veins using ANN [14].

Ghazi *et al.* performed plant species identification using pre-trained DL model such as AlexNet, GoogleNet, and VGGNet. LifeCLEF 2015 plant dataset is used for the analysis. The data augmentation is performed to minimise overfitting. It is reported that the performance of pre-trained models is affected depending on the number of iterations and data augmentation [15].

Sun *et al.* [16] designed a 26-layer residual network (ResNet) model for plant species identification. This model is tested over a real-time dataset collected from Beijing known as BJFU100

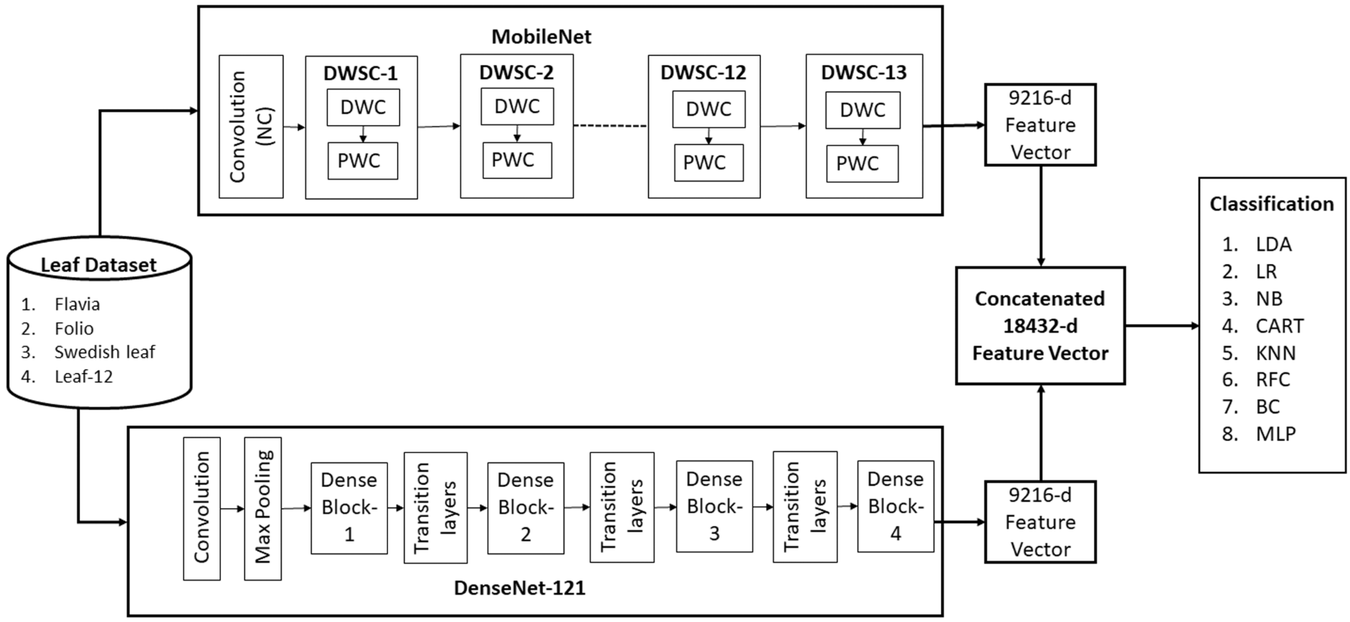

Fig. 1 *Proposed dual deep learning architecture for plant species recognition*

dataset. BJFU100 dataset contains 100 classes of plant species. Also, this method is validated using the Flavia dataset.

Lee *et al.* [17] proposed a CNN-based DL model. It is used to compute the features of the raw input image. Also, the deconvolution network is used to characterise the leaf data. The authors reported that the leaf venation produced better results compared with the leaf shape. Improved performances are observed for hybrid local–global features using DL methods compared with other approaches.

Liu *et al.* [18] proposed a hybrid DL model for feature extraction from a plant leaf. The hybrid DL method is a fusion of two methods, AutoEncoder (AE) and CNN. The extracted features are classified using SVM. On testing the method with ICL leaf dataset, the authors concluded that the hybrid DL method (93%) performed better than pure SVM (88%), pure AE (90%), and pure CNN (91%).

Barré *et al.* [19] developed a DL-based plant identification system called LeafNet. The authors show that the CNN model learns specific features better than handcrafted features. LeafNet consists of several sets of convolution layer with max pooling followed by three FCLs. This CNN model is evaluated using three datasets (LeafSnap, Flavia, and Foliage).

Pawara *et al.* [20] compared the performance of the conventional method and DL method in plant species identification using AgrilPlant, LeafSnap, and Folio datasets. AgrilPlant is a dataset created by the authors. It contains 10 classes (300 images per class) of agricultural plants. For DL method, AlexNet and GoogleNet models are trained from scratch as well as fine-tuned. In the conventional method, the authors utilise the histogram of oriented gradient (HOG) with KNN classifier. The other conventional method used by the authors is a bag of visual words (BOWs) with HOG as feature extraction and SVM, MLP as a classifier. For AgrilPlant dataset, 79.43 and 98.33% are the highest accuracies obtained for HOG–BOW with SVM and fine-tuned GoogleNet, respectively.

Hu *et al.* [21] proposed a multi-scale fusion CNN (MSF-CNN) for plant leaf recognition. Images are downsized to 256×256, 128×128, 64×64, and 32×32 using bilinear interpolation operations. Convolution, batch normalisation, and rectified linear unit (ReLU) unit is applied individually to each of the mentioned image sizes. Concatenation operation is performed for feature fusion between two different scales. Two datasets, namely Malayakew dataset and LeafSnap dataset are used for evaluation of MSF-CNN method.

He *et al.* [22] proposed a bi-channel DL framework for plant species identification. The authors combined two pre-trained models, namely VGG-16 and SqueezeNet [38], using a stacking layer. This bi-channel DL framework is evaluated over the plant dataset consisting of plant species of the Orchidaceae family. The authors reported an accuracy of 96.81%. Sulc *et al.* [23] proposed a fine-tuned ResNet for plant species recognition. FCL of the ResNet is replaced with two randomly initialised FCL with the maxout function. This method is tested over the PlantCLEF 2016 challenge.

Based on the extensive literature survey, it has been identified that the performance analysis of fused DL architectures for standard datasets and the real-time dataset is sparse.

In this paper, it is proposed to study the performance of dual deep learning architecture (DDLA) with ML classifiers in the context of plant species recognition. It is termed as DDLA, as two DL models are combined for feature extraction.

3 Proposed methodology

The workflow for plant species recognition is represented in Fig. 1. DL-based CNN models act as feature extractors. Two pre-trained DL models (MobileNet and DenseNet-121) are combined and used as feature extractors. The FCLs in DL models are removed. Furthermore, the ML classifiers are utilised for classification. Four datasets, namely Flavia [4], Folio [6], Swedish leaf [7], and Leaf-12 datasets are used for the analysis.

3.1 Image reconstruction and pre-processing

Images are zero padded and resized (300×300) using a super-sampling technique known as anti-aliasing. Anti-aliasing is employed to reduce oversampling while resizing the image. Images are resized by maintaining their aspect ratio and image quality. Furthermore, the images are resized to 100×100 using the nearest interpolation method.

3.2 Feature extraction

In this research work, the MobileNet and DenseNet-121 architectures are combined and used as a feature extractor. ML classifiers replace FCL in modified architecture.

The proposed DDLA method is appropriate for feature extraction rather than for classification as the datasets experimented are small (Table 1 mentions the details of the dataset). Thus, the convolutional base of MobileNet and DenseNet-121 are used.

DenseNet-121 architecture contains 121 layers, the minimum among its variants (DenseNet-169 and DenseNet-201). The reason for choosing MobileNet and DenseNet-121 combination is due to their properties such as lightweight and improved performance. MobileNet is a lightweight model [35] with decent performance

Table 1 Details of datasets

Dataset	Number of classes	Number of images in each class
Flavia	32	50
Folio	32	18
Swedish leaf	15	75
Leaf-12	12	320

and performance of DenseNet-121 [34] is better than the ResNet model. 9216-d feature vector is obtained from the individual architectures, namely MobileNet and DenseNet-121. Then, the feature vectors from the two architectures are concatenated to obtain 18432-d feature vector.

3.2.1 MobileNet: MobileNet [35] is a deep learning model designed to embed a computer vision application in mobile phones. MobileNet model is lightweight, less complex, much faster compared with other state-of-the-art models. It uses a depthwise separable convolution (DWSC). In this model, the first convolution is followed by 13 DWSCs. Batch normalisation and ReLU activation function follow every convolution in MobileNet. Normal convolution [39] slides over the image (y) through a filter containing weights (W) as represented in the equation below:

$$\mathrm{NC}(W,\ y)_{(i,\ j)} = \sum_{k=1}^{k=K}\sum_{l=1}^{l=L}\sum_{m=1}^{m=M} W_{(k,\ l,\ m)} \cdot y_{(i+k,\ j+l,\ m)} \tag{1}$$

where $k = 1$ to K rows (width of the image); $l = 1$ to L columns (height of the image); (i, j) index position of the image; and $m = 1$ to M number of filters.

DWSC is made up of two layers, namely depthwise convolution (DWC) [39] and pointwise convolution (PWC) [39]. In DWC, the filters or weight matrix is applied to each channel of the input image. DWC acts as a feature learning for each colour channel. Equations (2)–(4) [39] represent DWSC, DWC, and PWC. In (2), DWC is applied to the image, y using weight filter W_d. $\mathrm{PWC}_{(i,j)}$ makes use of weight filter W_p. Equation (2) depends on (3) and (4)

$$\mathrm{DWSC}(W_\mathrm{p},\ W_\mathrm{d},\ y)_{(i,\ j)} = \mathrm{PWC}_{(i,\ j)}(W_\mathrm{p},\ \mathrm{DWC}_{(i,\ j)}(W_\mathrm{d},\ y)) \tag{2}$$

$$\mathrm{DWC}(W_\mathrm{d},\ y)_{(i,\ j)} = \sum_{k=1}^{k=K}\sum_{l=1}^{l=L} W_{\mathrm{d}(k,\ l)} \odot y_{(i+k,\ j+l)} \tag{3}$$

where $k = 1$ to K rows (width of the image); $l = 1$ to L columns (height of the image); and (i, j) index position of the image.

PWC consists of 1×1 convolution to merge the outputs of DWC. The main feature of PWC is that it reduces the dimensions of output feature maps from DWC. As seen in (4), output feature maps of PWC are directly dependent on the number of filters (m) used

$$\mathrm{PWC}(W_\mathrm{p},\ y)_{(i,\ j)} = \sum_{m=1}^{M} W_m \cdot\ y_{(i,\ j,\ m)} \tag{4}$$

Here, W_m is the weight filter of PWC and $m = 1$ to M number of filters.

3.2.2 DenseNet-121: DenseNet [34] is a variant of ResNet. It solves the problem of vanishing gradient. It is better than ResNet in terms of a number of parameters and floating-point operations per second. It is comprised of four dense blocks and three transition layers.

Dense blocks contain 3×3 and 1×1 convolution sets. Here, 3×3 and 1×1 convolutions are repeated within the four dense blocks as 6, 12, 24, and 16 times. A transition layer is embedded in between two dense layers. Inside a dense block, every convolution layer is connected to other convolution layers in a feed-forward manner. The dense connection is given by

$$a_\mathrm{L} = D_\mathrm{L}([a_0,\ a_1,\ \ldots,\ a_{\mathrm{L}-1}]) \tag{5}$$

In (5), a_L is the current layer and it is a concatenation of feature maps of all previous layers $a_0, a_1, \ldots, a_{\mathrm{L}-1}$. DenseNet uses a hyperparameter called 'growth rate'. The growth rate keeps track of information from the previous layer. The dense connections follow a feed-forward mechanism between the layers. The transition layer consists of batch normalisation, 1×1 convolution, and 2×2 average pooling with a stride value of 2.

3.3 Classification

FCLs in deep learning architectures are replaced with ML classifiers such as LR, LDA, NB, KNN, CART, RFC, BC, and MLP [36, 37].

3.3.1 Linear discriminant analysis: LDA [36] is a statistical method. Although LDA is a well known technique for feature reduction, it is also used as a classifier. It supports the multi-class classification. LDA computes the mean and covariance for the entire dataset to identify within-class scatter and between-class scatter. For analysis, 'svd' is used as a solver [40].

3.3.2 Logistic regression: LR [41] predicts the outcome based on dependent variables. LR is a multi-equation model based on logits. Logits uses a logarithmic function that restricts the probability between 0 and 1. Equation (6) [41] represents the probability of sample data 'x' belonging to the class 'i' (if 1 to m classes are present), $y^{(i)}$) is the class label and $\boldsymbol{w}$ is the weight vector of class 'i'

$$P(y^{(i)=1|x,\ \boldsymbol{w}}) = \frac{\exp\left(w^{(i)^\mathrm{T}}x\right)}{\sum_{j=1}^{m}\exp\left(w^{(j)^\mathrm{T}}x\right)} \tag{6}$$

3.3.3 Naive bayes classifier: NB [36, 37] is a classifier based on probability. NB uses the product of the conditional probability of all elements with respect to the classification. Class label (y) in (7) is assigned based on the maximum probability of a class C_i and conditional probabilities of individual features $X_k^j = a_k$ for vector elements varying from $k = 1, \ldots, K$

$$y\ = \mathrm{argmax}_{k\epsilon 1,\ \ldots,\ K}\left[P(C_i)\prod_k P\left(X_k^j = a_k \middle| C_i\right)\right] \tag{7}$$

3.3.4 K-Nearest neighbour classifier: KNN is an instance-based learning method that uses the standard distance metric, Euclidean distance. If x_i is an instance to be classified and k is the number of nearest neighbours (NNs), calculate the distance between x_i and other k number of NNs. If $k = 1$, x_i is assigned to the class of single NN. If k is >1, then the majority voting rule is applied [37]. In this work, the best k value is identified between $k = 1, 3, 5, \ldots, 49$.

3.3.5 Classification and regression tree: CART [36] is a supervised ML algorithm. In CART, every internal node performs a test on the attributes and the leaf node holds the class label.

3.3.6 Random forest classifier: It is a supervised ML technique [36, 37]. This algorithm is suited for both classification and regression. Multiple trees are created using CART. Random forest uses majority voting for classification and average mean for

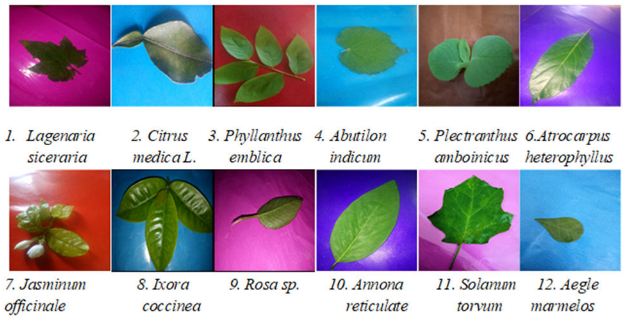

Fig. 2 *Sample images of the Leaf-12 dataset*

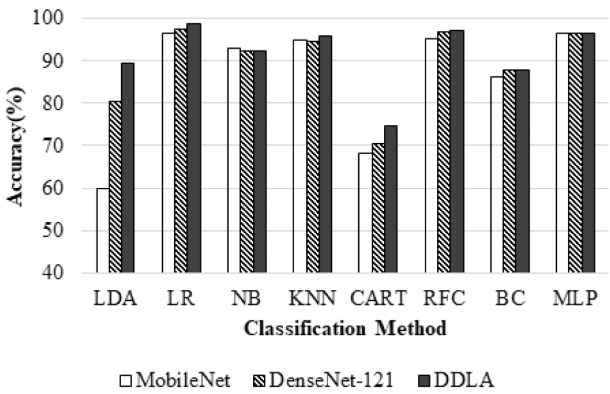

Fig. 3 *Result analysis of Flavia dataset*

regression. The classifier does not overfit the model and does not require *k*-fold cross-validation as the classifier considers random samples. In the experimental analysis, the number of trees considered is 200.

3.3.7 Bagging classifier: BC [36] is an ensembling technique. It uses bootstrap aggregation method. Bootstrap gets a subsample from the dataset with replacement. To each subsample, CART is applied and each of their results is aggregated together.

3.3.8 Multi-layer perceptron: MLP [36, 37] is an ANN model. It follows the backpropagation algorithm of the neural network. In the studies, one hidden layer with 512 neurons is used. For MLP, the results are obtained by using Adam optimiser with a learning rate of 0.001. A maximum number of iterations are being set to 200. A parameter 'tol' is set to 0.0001. 'Tol' parameter checks for the convergence of iterations, i.e. iterations stop if ten consecutive iterations do not show at least an improvement in accuracy by 0.0001 [40].

4 Experimental analysis

Plant datasets considered in the studies are Flavia [4], Folio [6], Swedish leaf [7], and custom created Leaf-12 dataset. The detailed description of the datasets is shown in Table 1. Flavia (1600 images), Folio (576 images), and Swedish leaf (1125 images) are the three benchmark leaf datasets. These dataset images are with white background. Flavia and Folio datasets contain an uneven number of images, and hence for experimental analysis, the equal number of images is selected.

Leaf-12 is our own custom collected dataset. The images are photographed under varied lighting conditions, various colour backgrounds, cluttered backgrounds, and different viewpoints. Sample images of Leaf-12 dataset are shown in Fig. 2.

For all the four datasets, train and test split are 70 and 30%, respectively. The experiments are performed using Python framework in Windows 10 64 bit OS with Intel Core i7-4790 CPU and NVIDIA Titan X GPU with 3584 CUDA cores. Several Python-based packages such as keras [42], Scikit-learn [40], Numpy, Matplotlib, h5py etc. are used. A DL package 'keras' with Tensorflow backend is used. For each standard leaf dataset, the results are compared with the existing DL and non-DL approaches of plant species recognition.

Table 2 Proposed method with existing methods for Flavia dataset

Method	DL/non-DL approach	Accuracy, %
PNN [4]	non-DL	90
RBPNN [8]	non-DL	93.82
ConvNet [24]	DL	94.6875
D-Leaf [14]	DL	94.63
proposed method (DDLA + LR)	**DL**	**98.71**

4.1 Analysis of Flavia dataset

The results of Flavia dataset are shown in Fig. 3 for classifiers such as LDA, LR, NB, KNN, CART, RFC, BC, and MLP. An accuracy of 98.71% is observed within 239.27 s for DDLA with LR classifier. DenseNet-121 with LR classifier produced an accuracy of 97.92% within 114.07 s. MobileNet with LR classifier produced an accuracy of 96.67% within 102.23 s. Other than LR classifier, accuracies of the RFC, MLP, and KNN as classifiers along with DDLA are >95%.

Performance analysis is also carried out by considering DenseNet-121 and MobileNet architectures as feature extractor and classifier. An FCL having the same number of neurons as that of the output class with softmax activation function is added. The results of MobileNet and DenseNet-121 with one FCL are obtained by setting the parameters (batch size of 32, number of epochs is 50, the learning rate of 0.0001, and optimiser used is Adam).

Using MobileNet and DenseNet-121 architectures as feature extractor and classifier, accuracies of 97.29 and 98.12% are obtained within 125.88 and 395.55 s. It is observed that the individual models used as both feature extraction and classification resulted in lesser accuracies than that of DDLA + LR. The times taken by MobileNet + FCL and DenseNet-121 + FCL are comparatively higher than MobileNet + LR and DenseNet-121 + LR. Also, DenseNet-121 + FCL takes longer time than DDLA + LR.

Experimental studies clearly show that DDLA with ML classifiers outperforms the single deep learning architectures in terms of accuracy and comparable computational time. Hence, the experiments for DDLA as both feature extraction and classification are not considered in the studies.

DDLA architecture-based plant species recognition resulted in higher accuracies compared with the existing methods and is tabulated in Table 2. The result of the proposed methodology is better than non-DL approaches (probabilistic neural network (PNN), radial basis PNN (RBPNN), and DL approaches (ConvNet and D-Leaf)).

4.2 Analysis of Folio dataset

DDLA with ML classifiers is applied for plant species recognition using Folio dataset. The best accuracies of MobileNet, DenseNet-121, and DDLA are attained with LR classifier. The DDLA along with LR classifier resulted in an accuracy of 96.38%. It outperformed MobileNet + LR (92.49% in 32.19 s) and DenseNet-121 + LR (95.95% in 52.76 s).

The results are shown in Fig. 4. It is observed that the LR classifier performed the best among other classifiers. Also, KNN is the second best classifier for DDLA with an accuracy of 93.06%.

The MobileNet and DenseNet architectures are tested for feature extraction and classification. Parameters used are one FCL with softmax activation function, the Adam optimiser with a learning rate of 0.0001, batch size of 32, and epochs of 50. MobileNet + FCL and DenseNet-121 + FCL produced accuracies of 86.13% in 50.22 s and 92.49% in 169.61 s, respectively. It is observed that the individual model accuracies are comparatively lesser than DDLA + LR. It is also to be noted that the MobileNet + LR and DenseNet-121 + LR produced better accuracies compared

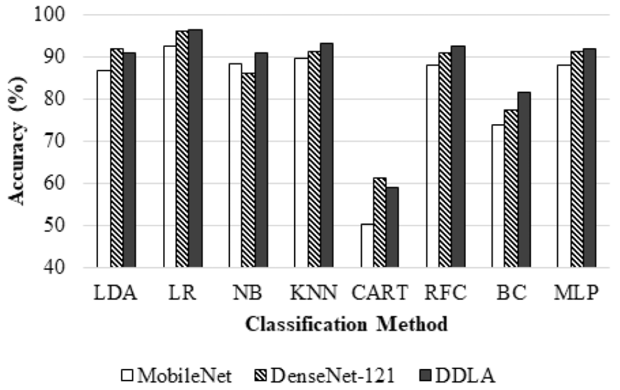

Fig. 4 *Result analysis of Folio dataset*

Table 3 Proposed method with existing methods for Folio dataset

Method	DL/non-DL approach	Accuracy, %
KNN [6]	non-DL	87.3
HOG + KNN [20]	non-DL	84.3 ± 1.62
AlexNet scratch [20]	DL	84.83 ± 2.85
GoogleNet scratch [20]	DL	89.75 ± 1.74
proposed method (DDLA + LR)	**DL**	**96.38**

to MobileNet + FCL and DenseNet-121 + FCL. The computational time of DDLA + LR is 60.82 s and the time taken by the proposed methodology is lesser compared with DenseNet-121 + FCL.

The result of the proposed method is compared with the reported works on Folio dataset in Table 3. Pawara *et al.* [20] trained AlexNet and GoogleNet from scratch and their reported accuracies are 84.83 and 89.75%, respectively. The proposed method outperformed non-DL approaches of Munisami *et al.* (shape features with KNN) [6] and Pawara *et al.* [20] (histogram of gradients (HOG) with KNN).

4.3 Analysis of Swedish leaf dataset

Accuracy chart for plant species recognition using DDLA with ML classifiers is shown in Fig. 5. It is observed that the DDLA attained an accuracy of 99.41% (80.62 s) with LR classifier for the Swedish leaf dataset. Similar accuracy of 99.41% (66.5 s) is obtained for MobileNet-121 with LR. DenseNet-121 + LR produced an accuracy of 98.93% in 87.04 s. Accuracies of RFC and MLP are are >95% irrespective of the feature extraction method.

MobileNet + FCL produced an accuracy of 97.04% in 92.07 s. MobileNet + FCL accuracy is lesser and takes longer computational time in comparison with MobileNet + LR. DenseNet-121 + FCL (99.41% in 296.29 s for 50 epochs) produced a similar accuracy as DDLA + LR but it takes more computational time.

Table 4 lists the comparison between the proposed methodologies with reported works in the literature for the Swedish leaf dataset. The DDLA achieved higher accuracy than non-DL approaches such as multi-scale distance matrix (MDM) with 1-NN classification (1-NN) [9] and a bag of contour fragments (BCFs) with SVM [10]. It is also higher than D-Leaf, as reported by Tan *et al.* [14]. It is observed that the results of DDLA perform better in comparison to the CNN model of Atabay with a variation of 0.3% [25].

4.4 Analysis of Leaf-12 dataset

Performance of DDLA architecture with ML classifier is shown in Fig. 6. It is observed that the DDLA + LR method resulted in an accuracy of 99.39% for Leaf-12 dataset. Other than LR classifier, LDA and MLP classifiers attained better accuracies (>96%) for the proposed feature extraction method.

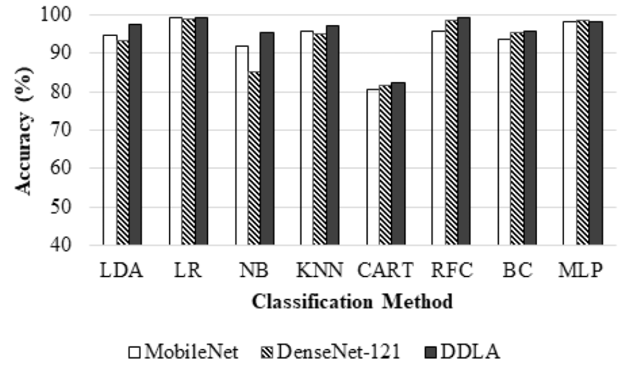

Fig. 5 *Result analysis of Swedish leaf dataset*

Table 4 Comparison of the proposed method with existing methods for Swedish leaf dataset

Method	DL/non-DL approach	Accuracy, %
MDM + 1-NN [9]	non-DL	91.33
BCF + SVM [10]	non-DL	96.56 ± 0.67
D-Leaf [14]	DL	98.09
CNN [25]	DL	99.11
proposed method (DDLA + LR)	**DL**	**99.41**

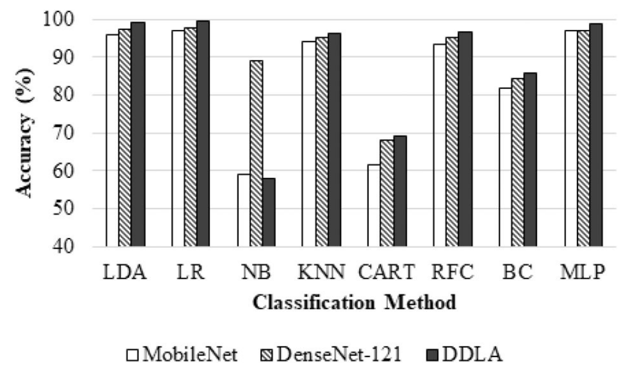

Fig. 6 *Result analysis of Leaf-12 dataset*

MobileNet + LR and MobileNet + FCL obtained accuracies of 97.14% in 255.49 s and 98.52% in 290.69 s. DenseNet-121 + LR and DenseNet-121 + FCL achieved accuracies of 97.92% in 262.46s s and 98.26% in 910.23 s. Computational time of DDLA + LR for Leaf-12 dataset is 541.45 s and is lesser than DenseNet-121 + FCL.

5 Summary of results

Table 5 shows the time taken in seconds, accuracy, and other performance metrics obtained for the four datasets considered.

The performance metrics considered are rank-1 accuracy, rank-5 accuracy, precision, recall, and F1-score. It is noted that the time taken for the proposed methodology varies with respect to the datasets considered.

Rank-1 is checking whether the top class is the same as the target label. In rank-5 accuracy, the target label is one among the top five predictions.

Average Rank-*n* accuracy [21] is given by

$$\text{Rank} - n = \frac{1}{m} \cdot \sum_{i=1}^{m} \sum_{j=1}^{n} I(f(x_i, W) = y_i) \tag{8}$$

where, m is the number of samples for testing, $n = 1$, 5, or 10, y_i is the correct prediction class for test images, x_i, $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$.

Table 5 Metrics and time taken for the proposed methodology (DDLA + LR classifier)

Dataset	Time, s With GPU	Accuracy, % Rank-1	Rank-5	Precision	Recall	F1-score
Flavia	239.27	98.71	99.79	0.98	0.98	0.98
Folio	60.82	96.38	100	0.96	0.95	0.95
Swedish Leaf	80.62	99.41	100	0.99	0.99	0.99
Leaf-12	541.45	99.39	100	0.99	0.99	0.99

Table 6 Accuracies and time taken for DDLA, MobileNet, and DenseNet-121

Method	Flavia		Folio		Swedish Leaf		Leaf-12	
	Accuracy, %	Time, s	Accuracy, %	Time, s	Accuracy, %	Time, s	Accuracy, %	Time, s
DDLA + LR	98.71	239.27	96.38	60.82	99.41	80.62	99.39	541.45
MobileNet + LR	96.67	102.23	92.49	32.19	99.41	66.5	97.14	255.49
DenseNet-121 + LR	97.92	114.07	95.95	52.76	98.93	87.04	97.92	262.46
MobileNet + FCL	97.29	125.88	86.13	50.22	97.04	92.07	98.52	290.69
DenseNet-121 + FCL	98.12	395.55	92.49	169.61	99.41	296.29	98.26	910.23

Rank-1 accuracy for Leaf-12 dataset (99.39%) is almost similar to the Swedish leaf dataset (99.41%). Rank-5 accuracy of the proposed method is 100% for the three datasets, except for Flavia dataset.

Precision is calculated by the ratio of the number of true positives (TPs) to the sum of TP and false positive (FP) as given in (9). The recall is the ratio of TPs to the sum of TP and false negative (FN) as specified in (10). F1-score is the weighted average of precision and recall as given in (11). Precision, recall, and F1-score use a weighted averaging to handle class imbalance [40]. This weight $(1/(\sum_{l \in L} |y_l|))$ is based on the number of ground truth samples (y_l). For each class, this weight is multiplied with precision, recall, and F1-score to calculate their average values

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{F1} - \text{score} = 2 \times \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \tag{11}$$

Based on the experimental studies, it is observed that the DDLA (MobileNet + DenseNet-121) with LR classifier achieved higher accuracy for plant species recognition using benchmark datasets (Flavia, Folio, and Swedish leaf) and custom developed dataset (Leaf-12).

6 Conclusion

In this paper, a dual deep learning architecture (DDLA) is used in the context of plant species recognition. The proposed method is used as a feature extraction (MobileNet + DenseNet-121 architectures) and is classified using ML classifiers. Three benchmark datasets (Flavia, Folio, and Swedish leaf) and one custom dataset known as Leaf-12 are considered in the studies. It is observed that the proposed method (DDLA + LR) achieved higher accuracies for both standard and custom datasets. The proposed methodology obtained an accuracy of 99.39% for Leaf-12 dataset. It is also observed that the DDLA + LR architecture resulted in higher accuracy compared with the individual architectures with LR classifier. The computation time of DDLA + LR is comparable with individual architectures as summarised in Table 6.

7 Acknowledgment

The authors thank NVIDIA for providing GPU under the University Research Grant Programme.

8 References

[1] Du, J.X., Wang, X.F., Zhang, G.J.: 'Leaf shape based plant species recognition', *Appl. Math. Comput.*, 2007, **185**, (2), pp. 883–893

[2] Jin, T., Hou, X., Li, P., *et al.*: 'A novel method of automatic plant species identification using sparse representation of leaf tooth features', *PLoS One*, 2015, **10**,, (10), p. e0139482

[3] Aakif, A., Khan, M.F.: 'Automatic classification of plants based on their leaves', *Biosyst. Eng.*, 2015, **139**, pp. 66–75

[4] Wu, S.G., Bao, F.S., Xu, E.Y., *et al.*: 'A leaf recognition algorithm for plant classification using probabilistic neural network'. IEEE Int. Symp. Signal Processing and Information Technology, Cairo, Egypt, December 2007, pp. 11–16

[5] Wang, Z., Sun, X., Zhang, Y., *et al.*: 'Leaf recognition based on PCNN', *Neural Comput. Appl.*, 2016, **27**, (4), pp. 899–908

[6] Munisami, T., Ramsurn, M., Kishnah, S., *et al.*: 'Plant leaf recognition using shape features and colour histogram with *K*-nearest neighbour classifiers', *Procedia Comput. Sci.*, 2015, **58**, pp. 740–747

[7] Söderkvist, O.: 'Computer vision classification of leaves from Swedish trees'. Master's thesis, Linkoping University, 2001

[8] Kulkarni, A.H., Rai, H.M., Jahagirdar, K.A., *et al.*: 'A leaf recognition technique for plant classification using RBPNN and Zernike moments', *Int. J. Adv. Res. Comput. Commun. Eng.*, 2013, **2**, (1), pp. 984–988

[9] Hu, R.X., Jia, W., Ling, H., *et al.*: 'Multiscale distance matrix for fast plant leaf recognition', *IEEE Trans. Image Process.*, 2012, **21**, (11), pp. 4667–4672

[10] Wang, X., Feng, B., Bai, X., *et al.*: 'Bag of contour fragments for robust shape classification', *Pattern Recognit.*, 2014, **47**, (6), pp. 2116–2125

[11] Saleem, G., Akhtar, M., Ahmed, N., *et al.*: 'Automated analysis of visual leaf shape features for plant classification', *Comput. Electron. Agric.*, 2019, **157**, pp. 270–280

[12] Kour, V.P., Arora, S.: 'Particle swarm optimization based support vector machine (P-SVM) for the segmentation and classification of plants', *IEEE Access*, 2019, **7**, pp. 29374–29385

[13] Liu, N., Kan, J.M.: 'Improved deep belief networks and multi-feature fusion for leaf identification', *Neurocomputing*, 2016, **216**, pp. 460–467

[14] Tan, J.W., Chang, S.W., Kareem, S.B.A., *et al.*: 'Deep learning for plant species classification using leaf vein morphometric', *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2018, p. 1, https://ieeexplore.ieee.org/abstract/document/8388220

[15] Ghazi, M.M., Yanikoglu, B., Aptoula, E.: 'Plant identification using deep neural networks via optimization of transfer learning parameters', *Neurocomputing*, 2017, **235**, pp. 228–235

[16] Sun, Y., Liu, Y., Wang, G., *et al.*: 'Deep learning for plant identification in natural environment', *Comput. Intell. Neurosci.*, 2017, **2017**, p. 6

[17] Lee, S.H., Chan, C.S., Mayo, S.J., *et al.*: 'How deep learning extracts and learns leaf features for plant classification', *Pattern Recognit.*, 2017, **71**, pp. 1–13

[18] Liu, Z., Zhu, L., Zhang, X.P., *et al.*: 'Hybrid deep learning for plant leaves classification'. Int. Conf. Intelligent Computing, Cham, August 2015, pp. 115–123

[19] Barré, P., Stöver, B.C., Müller, K.F., *et al.*: 'LeafNet: a computer vision system for automatic plant species identification', *Ecol. Inf.*, 2017, **40**, pp. 50–56

[20] Pawara, P., Okafor, E., Surinta, O., *et al.*: 'Comparing local descriptors and bags of visual words to deep convolutional neural networks for plant recognition'. ICPRAM, Porto, Portugal, February 2017, pp. 479–486

[21] Hu, J., Chen, Z., Yang, M., *et al.*: 'A multiscale fusion convolutional neural network for plant leaf recognition', *IEEE Signal Process. Lett.*, 2018, **25**, (6), pp. 853–857

[22] He, G., Xia, Z., Zhang, Q., *et al.*: 'Plant species identification by bi-channel deep convolutional networks', *J. Phys., Conf. Ser.*, 2018, **1004**, (1), IOP Publishing, pp. 1–6, https://iopscience.iop.org/article/10.1088/1742-6596/1004/1/012015/meta

[23] Sulc, M., Mishkin, D., Matas, J.: 'Very deep residual networks with maxout for plant identification in the wild'. Working Notes of CLEF, Evora, Portugal, September 2016

[24] Zhang, C., Zhou, P., Li, C., *et al.*: 'A convolutional neural network for leaves recognition using data augmentation'. 2015 IEEE Int. Conf. Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), Liverpool, UK, October 2015, pp. 2143–2150

[25] Atabay, H.A.: 'A convolutional neural network with a new architecture applied on leaf classification', *IIOAB J.*, 2016, **7**, (5), pp. 226–331

[26] Kaya, A., Keceli, A.S., Catal, C., *et al.*: 'Analysis of transfer learning for deep neural network based plant classification models', *Comput. Electron. Agric.*, 2019, **58**, pp. 20–29

[27] Pearline, S.A., Kumar, V.S., Harini, S.: 'A study on plant recognition using conventional image processing and deep learning approaches', *J. Intell. Fuzzy Syst.*, 2019, **36**, (3), pp. 1–8

[28] Krizhevsky, A., Sutskever, I., Hinton, G.E.: 'ImageNet classification with deep convolutional neural networks', In NIPS'12 Proc. of the 25th Int. Conf. on Neural Information Processing Systems, **vol 1**, Lake Tahoe, Nevada, December 2012, pp. 1097–1105

[29] Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for large-scale image recognition'. 2014, arXiv preprint arXiv:1409.1556

[30] Szegedy, C., Vanhoucke, V., Ioffe, S., *et al.*: 'Rethinking the inception architecture for computer vision'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, USA, 2016, pp. 2818–2826

[31] Szegedy, C., Ioffe, S., Vanhoucke, V., *et al.*: 'Inception-v4, inception-ResNet and the impact of residual connections on learning'. Proc. 31st AAAI Conf. Artificial Intelligence, San Francisco, USA, February 2017

[32] Chollet, F.: 'Xception: deep learning with depthwise separable convolutions'. 2017, arXiv preprint, 1610–02357

[33] He, K., Zhang, X., Ren, S., *et al.*: 'Deep residual learning for image recognition'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, USA, 2016, pp. 770–778

[34] Huang, G., Liu, Z., Van Der Maaten, L., *et al.*: 'Densely connected convolutional networks'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Honolulu, USA, 2017, pp. 4700–4708

[35] Howard, A.G., Zhu, M., Chen, B., *et al.*: 'MobileNets: efficient convolutional neural networks for mobile vision applications'. 2017, arXiv preprint arXiv:1704.04861

[36] Marsland, S.: '*Machine learning: an algorithmic perspective*' (Chapman and Hall/CRC, Boca Raton, USA, 2011)

[37] Mitchell, T.M: '*Machine learning*' (McGraw-Hill, New York, USA, 1997)

[38] Iandola, F.N., Han, S., Moskewicz, M.W., *et al.*: 'SqueezeNet: AlexNet-level accuracy with 50 × fewer parameters and <0.5 mb model size'. 2016, arXiv preprint arXiv:1602.07360

[39] Kaiser, L., Gomez, A.N., Chollet, F.: 'Depthwise separable convolutions for neural machine translation'. 2017, arXiv preprint arXiv:1706.03059

[40] Pedregosa, F., Varoquaux, G., Gramfort, A., *et al.*: 'Scikit-learn: machine learning in python', *J. Mach. Learn. Res.*, 2011, **12**, (Oct), pp. 2825–2830

[41] Krishnapuram, B., Carin, L., Figueiredo, M.A., *et al.*: 'Sparse multinomial logistic regression: fast algorithms and generalization bounds', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, **27**, (6), pp. 957–968

[42] 'Keras documentation', 2015. Available at: https://keras.io/getting-started/faq/#how-should-i-cite-keras

[43] Goodfellow, I., Bengio, Y., Courville, A., *et al.*: '*Deep learning*', vol. **1**, (MIT Press, Cambridge, 2016)), pp. 322–366