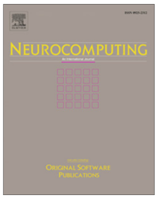
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Two-view fine-grained classification of plant species

Voncarlos M. Araújo [a,*], Alceu S. Britto Jr. [a,b], Luiz S. Oliveira [c], Alessandro L. Koerich [d]

[a] *Pontifical Catholic University of Paraná (PUCPR), Curitiba, PR, Brazil*
[b] *State University of Ponta Grossa (UEPG), Ponta Grossa, PR, Brazil*
[c] *Federal University of Paraná (UFPR), Curitiba, PR, Brazil*
[d] *École de Technologie Supérieure (ÉTS), Université du Québec, Montréal, QC, Canada*

ARTICLE INFO

Article history:
Received 22 August 2020
Revised 2 July 2021
Accepted 2 October 2021
Available online 7 October 2021
Communicated by Zidong Wang

Keywords:
Siamese neural network
Fine-grained classification
Plant species recognition
Deep metrics

ABSTRACT

Automatic plant classification is challenging due to the vast biodiversity of the existing plant species in a fine-grained scenario. Robust deep learning architectures have been used to improve the classification performance in such a fine-grained problem but usually build models that are highly dependent on a large training dataset and are not scalable. This paper proposes a novel method based on a two-view leaf image representation and a hierarchical classification strategy for fine-grained plant species recognition. It uses the botanical taxonomy as a basis for a coarse-to-fine strategy applied to identify the plant genus and species. The two-view representation provides complementary global and local features of leaf images. A deep metric based on Siamese Convolutional Neural Networks is used to reduce the dependence on many training samples and make the method scalable to new plant species. The experimental results on two challenging fine-grained datasets of leaf images (i.e., PlantCLEF 2015 and LeafSnap) have shown the proposed method's effectiveness, which achieved recognition accuracy of 0.87 and 0.96, respectively.

1. Introduction

Automated plant classification concerns the recognition of plant images into botanical species by applying machine learning algorithms [12,31,45]. The classification task may be performed on an entire plant's image or just on parts of it, such as branches, flowers, fruits, leaves, or stems. This pattern recognition task's main challenge is related to the vast biodiversity of the existing plant species. It is possible to observe the likeness between different species (high inter-class similarity) and sometimes significant differences among samples belonging to the same species (high intra-class variability). The blue dotted rectangle in Fig. 1 shows an example of the possible similarity among different species. In contrast, the red line rectangle presents an example of the difference between samples of the same species caused by shape, color, and texture changes. Such variability is usually caused by the plant maturity or even pose and illumination variation that may result from the image acquisition process. On top of that, there is the unbalancing data problem since some species are scarce in the flora environment and the scalability constraint, as the number of plant species being discovered by scientists is continuously growing.

In the literature, one may find some important strategies to deal with such difficulties inherent to the plant species recognition [9,42,47,48]. However, in the last years, methods based on fine-grained image classification (FGIC) have received special attention from the scientific community [5,14,33,38]. Such methods consist of discriminating between classes in a subcategory of objects, such as birds, animals, or vehicle models. Different from the traditional image classification methods, FGIC methods recognize coarse classes firstly. Then, it goes further by discriminating fine classes in which the classification difficulty is greater due to intra-class and inter-class variability like those observed in plant species. FGIC-based methods may explore the taxonomic relationship between the plant classes, which are hierarchically organized based on shared biological characteristics [32] into three levels of abstraction: *family*, *genus*, and *species*. Exploring these characteristics may help us distinguish very similar classes by first selecting candidates in a coarse level of the hierarchy, which can be distinguished in the finest level of the hierarchy (coarse-to-fine strategy). The visual recognition of leaves carried out by domain experts is generally based on this hierarchical strategy, the so-called "plant taxonomy relationship" [35].

With this in mind, we propose in this paper a two-view similarity learning strategy for the fine-grained plant classification, which

[*] Corresponding author.
E-mail addresses: voncarlos.araujo@gmail.com (V.M. Araújo), alceu@ppgia.pucpr.br (A.S. Britto Jr.), luiz.oliveira@ufpr.br (L.S. Oliveira), alekoe@gmail.com (A.L. Koerich).

https://doi.org/10.1016/j.neucom.2021.10.015
0925-2312/© 2021 Elsevier B.V. All rights reserved.

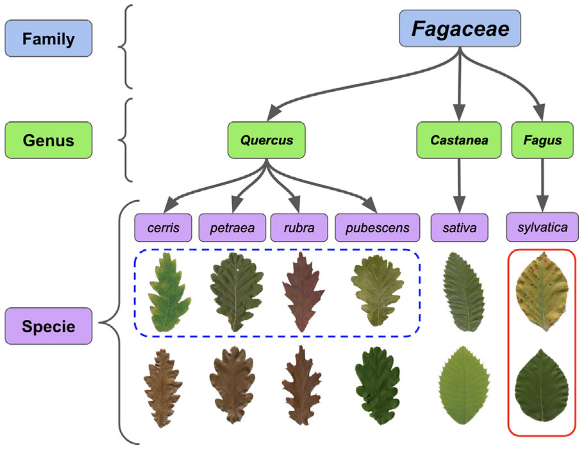


Fig. 1. Intra-class and inter-class problem. Inter-class (blue dotted rectangle), has species very similar and intra-class (red line rectangle) contains variations like background, occlusion, pose, color, illumination and plant maturity stages inside the same species.

consists of two stages that exploit different views of the leaf images. In the first stage, a coarse classification by plant genus is carried out using a deep metric based on a Siamese Convolutional Neural Network (S-CNN) to compute the similarity between a testing sample and the reference images previously defined for each plant genus. The deep metric learned from pairs of images provides the distance between two image samples represented by an unknown plant image and the genus reference images. At this stage, the entire leaf image is used, i.e., the S-CNN model's input is the whole leaf image characterizing a global view in terms of problem representation. The output of this stage is a ranked list of the best genus reference candidates.

In the second stage, a fine classification of plant species is performed. Similarly, an S-CNN is used as a deep metric. The similarity is computed between the cropped test sample and the cropped reference images representing the plant species in the ranking genus candidates list returned by the first stage. Here, a local representation (view) of the leaf image is used, i.e., the S-CNN receives as input a cropped image extracted from the center of the leaf image. The second stage's output is a final ranked list of plant species obtained by combining both stages' results.

The rationale behind the two-view scheme is to provide different representations of the problem. In the first view, the similarity computed by the S-CNN takes into account global features extracted from the entire leaf image (shape and color). In contrast, in the second view, local features based on texture and the plant veins are considered. Such a representation strategy allows us to treat some specific issues of the leaf classification problem. For instance, species of plants inside the same taxonomy level (e.g., *genus*) may look similar in terms of global features. Still, they present imperceptible tiny local changes in their texture and vein patterns that are important to characterize their species. Using a hierarchical classification decreases the number of classes to be evaluated in a *coarse-to-fine* strategy, reducing the complexity of the plant recognition task.

We carried out extensive experiments and compared the proposed method with state-of-the-art handcrafted approaches and methods based on deep learning using different CNN architectures. AlexNet[23], GoogLeNet[40] and VGG16[34] networks were selected as baseline of plant leaf recognition model, and transfer learning was used. For this purpose, a robust experimental protocol was defined based on two challenging fine-grained datasets of plant leaf images: PlantCLEF 2015 [19], and LeafSnap [24]. In most experiments, the proposed method outperforms several existing methods by achieving superior classification accuracy using few samples to compute the similarity between images. The learned model proposed does not need to be retrained when new plant species are added, which makes the proposed method highly scalable.

The contribution of this paper is fourfold: (i) the two-view representation of plant species enables the capture of the leaves' coarse and fine features, which are very useful to distinguish among different *genus* and different *species*, respectively; (ii) a slight comparative evaluation between deep models (AlexNet, GoogLeNet, and VGG16) and deep metric (S-CNN); (iii) the proposed method exploits the natural hierarchy of the problem combining coarse and fine representations into a hierarchical strategy that reduces the complexity of the classification task as a lower number of classes need to be disambiguated at each hierarchical level; (iv) the proposed method is highly scalable as new plant species can be easily added using few samples without retraining the S-CNN models. This is highly desirable in plant classification, where the number of new plant species is continuously growing, and the new ones often have few examples for training.

The paper is organized as follows. Section 2 reviews the relevant literature in plant classification. In Section 3, the proposed method is described in detail. Section 4 presents our experimental findings on plant classification. Finally, Section 5 presents our conclusions, future work perspectives, and final remarks.

2. Related Work

Recently, studies on plant classification based on image processing have become an interesting research topic in computer vision [4,5,11,17,46]. In the literature, there are many datasets that can be employed to evaluate plant classification methods such as Flavia [45], Foliage [20], Swedish [37], LeafSnap [24], PlantCLEF [19], ICL [38], and MalayaKew [26]. These datasets represent the problem domain well, exposing the many difficulties such as fine-grained complexity, imbalanced distribution, large intra-class variability, small inter-class variability, and noisy images.

One may find in the literature several contributions to plant species recognition. Naresh and Nagendraswamy [29] introduced a symbolic approach based on textural features extracted from leaf images for plant species recognition. A modified local binary pattern was proposed to extract features, and the classification was performed using a simple nearest neighbor classifier. Besides, the concept of clustering was used to define multiple class representatives by grouping similar leaf samples using a threshold to create clusters to decrease the intra-class variation. However, in their experiments, they observed the need to incorporate features extracted from other leaf views to improve the recognition accuracy. Aakif and Khan [1] proposed a shape-defining feature, which is combined with morphological and Fourier descriptors. These features were used with artificial neural networks. The method was evaluated on a proprietary dataset of 14 classes and on Flavia and ICL datasets. Their emphasis was more related to the performance in terms of computational time than the recognition accuracy.

Fine-grained recognition is a challenging problem that consists of recognizing subordinate categories such as species of birds [15], dog breeds [21] and flower species [3]. Over the past decade, fine-grained recognition has achieved high-performance levels thanks to the combination of deep learning techniques with large annotated training datasets [44]. Some recent works have considered deep learning techniques for fine-grained plant classification [5,7,25]. In particular, Barré et al. [7] and Lee et al. [25] have shown how convolutional neural networks (CNN) learn representations from plant leaf images using a deconvolutional approach. The most important finding is that shape information alone is not a good choice due to similar leaf contours, especially in closely related species. Therefore, it is important to exploit other kinds of features that may be present in leaf structure. Araújo et al. [5], explored two types of feature representations of the plant leaf using deep models. The first representation considered the entire image of the plant. After that, the central region of the image is cropped and highlight as the second representation. The approach used a hierarchical classification responsible for combining the outputs of global (entire images) and local (cropped images) features. The features are extracted by the GoogLeNet CNN model pre-trained on the ImageNet dataset. As a result, the authors showed an efficient use of the hierarchical classification instead of a traditional classification. Moreover, the two representations of leaves contributed to the complementarity of features. Despite that, in some cases, the combination of hierarchy levels using product fusion rule was not sufficient to discriminate species with similar characteristics due to not control the number of categories taken to the second level of the hierarchy. The CNN approach also depends on a huge quantity of data to create a robust deep model to handle unbalanced data. They achieved 0.86 using S metric in the final recognition performance for PlantCLEF 2015 dataset employing the data augmentation technique.

As observed in several works, CNN models usually need a high amount of data for training [18,30,41]. For instance, Barbedo [6] analyzed the impact of the number of training samples on a CNN's accuracy, and he found out that it requires a substantial number of training data to provide solid results. Barré et al.[7] described an approach based on CNNs for the plant classification, which employs data augmentation based on low-level transformations applied to the leaf images such as shifting, scaling, and rotation. They correctly recognized 86.3% of the 184 species on the LeafSnap dataset. The resulting CNN model needs to be retrained to include new plant species, which is a time-consuming process. Barré et al. [7] has also pointed out that most of the misclassified plants belonged to species that show strong visual similarities. Zhu et al. [48] introduced a two-way attention hierarchical model using CNNs. The first attention way consists of recognizing the family level based on plant taxonomy. The second attention way is to find a discriminative part of an input image by a heat-map strategy. They conducted experiments in Malayakew and ICL datasets, and the CNN with Xception architecture achieved an accuracy of 99% in both datasets. They used 90% of the datasets for training and the 10% reminding to test. Although the authors stated that they do not use any data augmentation strategy, they have balanced the training dataset. Thus, each class has a roughly equal number of samples.

A fine-grained classification approach may provide as output just a single class probability or a set of classes so-called "confidence-sets", which include the true class at a given confidence level. To this end, an input image is classified, and the top-k best-ranked classes are selected as the confidence-set. Sfar et al. [33] proposed a hierarchical classification of plants, in which they measured the posterior probabilities for each node of the hierarchy and then created the confidence-set using a confidence threshold. The experiments were carried out on four datasets, where three of them have a balanced number of samples per class in the training set. However, they observed a poor performance on the PlantCLEF 2011 dataset, which is imbalanced, since their strategy employed approach fails to recognize the species with few training samples.

Wang and Wang [43] used a few-shot learning method based on Siamese Convolutional Neural Network (S-CNN) to recognize leaf plants. The Euclidean metric was used to measure the distance between features. The structure of GoogLeNet inspired the S-CNN used by [43]. They evaluated the proposed method on Flavia, Swedish, and LeafSnap datasets. They used a small number of learning samples. The experimental results have shown that the highest classification accuracies (95.32%, 91.37%, and 91.75% for Flavia, Swedish and Leafsnap datasets, respectively.) were achieved using models trained with 20 samples per class. Zhi-Yong et al. [47] also used S-CNN for plant recognition. They proposed a spatial structure using a deep metric. The S-CNN was used to learn an embedding with similar and dissimilar pairs. Similar pairs were formed using the same plants' organs, and different species of plants organize different pairs. They used the PlantCLEF 2015, and the result was 0.84 using S metric, surpassing all other methods. It is worth mentioning that recurrent neural networks were used to model the spatial structure. Recently, Figueroa-Mata and Mata-Montero [13] proposed a way to learn a similarity metric that discriminates plant species. They compared whether S-CNN models are better than CNN models regarding the performance and computational cost. Also, new species (20 leaves of Costa Rican dataset) never seen by the model S-CNN were evaluated without retraining the proposed model. In their first experiment, they conclude that for datasets with fewer than 20 images per species, the S-CNN performed better than CNN in the context of plant recognition besides the fact of having a lower computational cost. The second experiment has shown that S-CNN can generalize to other plant species without retraining the model.

To the best of our knowledge, from the existing methods in the literature [7,16,27,39], only Wang and Wang [43], Zhi-Yong et al. [47] and Figueroa-Mata and Mata-Montero [13] have exploited deep metrics to compute the similarity between plant images. However, no previous work uses a two-view similarity scheme combined with a fine-grained classification of plants. Plant hierarchy and similarity learning make our method more accurate and scalable, as shown in the next sections.

3. Proposed Approach

We propose a fine-grained approach for the classification of plant species from the leaf image. The coarse-to-fine classification strategy unveils the plant genus in the first stage and then its species in the second stage as illustrated in Fig. 2. In the first stage, a coarse classification according to the plant genus is carried out using a deep metric based on a Siamese Convolutional Neural Network (named S-CNN (A)). It computes the similarity between a leaf image X_l and reference images X_r previously chosen to represent each plant genus. Features are extracted by the sub-networks of the S-CNN (A) employing the entire leaf image, which is considered the first view of the proposed approach. The rationale behind that is to compute the similarity between a test image and the genus reference images, assuming a global view of the plant, i.e., representing the leaf by general features such as the leaf shape and color. The first stage's output is the k-best genus candidates organized as a ranked genus reference list R_k, ordered by the similarity distance. We determine the k-best genus candidates in the first stage by computing the distance between the test image and each genus' reference. Then, we calculate w_i as the frequency of the i-th genus in the list of k-ranked candidates (R_k). We use such frequency information in the second stage to weigh the genus in the fusion process.

In the second stage, given the k-best genus candidates found in the first stage, a fine classification considering only the plant species which belong to such best candidate genus is performed. Similarly, an S-CNN is used as a deep metric, but now the S-CNN (B) is computed on a different view of the leaf images that consider a local representation (second view). For such an aim, the S-CNN (B) receives as input a cropped image extracted from the center of the leaf image. The idea behind this strategy is to perform a fine

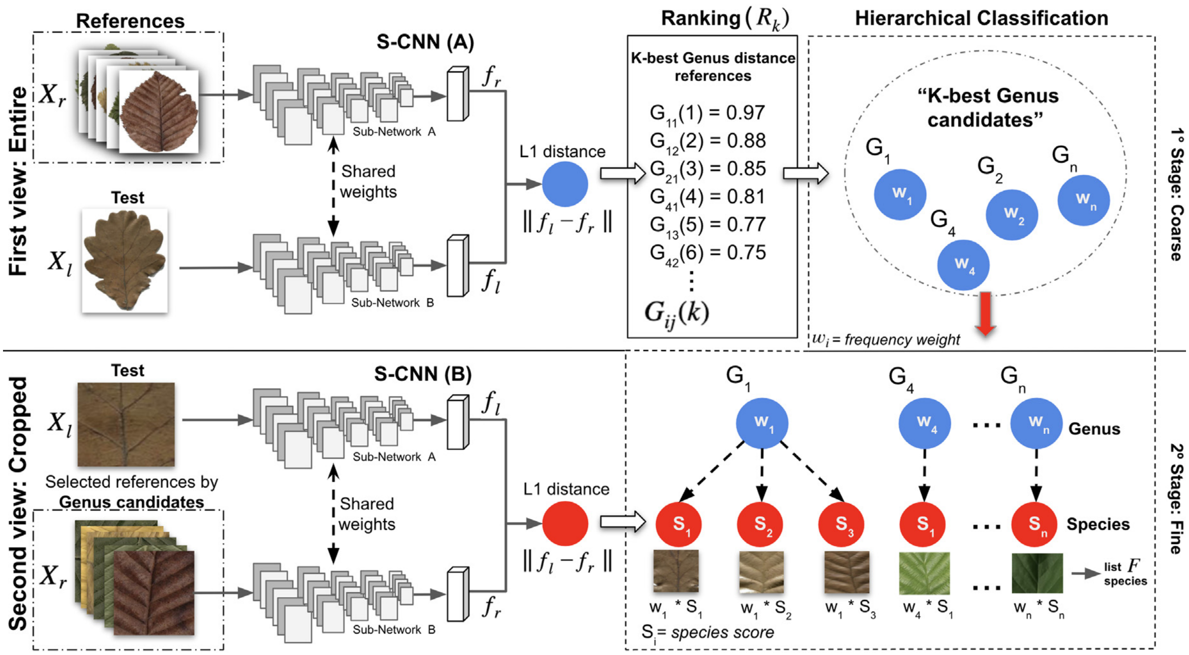


Fig. 2. Overview of the S-CNN proposed method for fine-grained recognition of plant species from the leaf image.

classification of plant species based on an adequate representation of the leaf image that focuses on local details such as the texture and the vein patterns that are usually present in the central part of leaves. The second stage's output is a ranked list F of plant species, which is weighted by the first stage's output, as shown in Fig. 2. In the next sections, we present the proposed method in detail.

3.1. S-CNN Learning

The similarity between the reference patterns and the leaf image is computed in both representation stages of the proposed method using an S-CNN [10]. The difference between the deep metric learning of both stages is that the S-CNNs are trained in different taxonomic groups and views of the leaf image. At the first stage, the S-CNN (A) is trained on the entire leaf image considering the genus group, while in the second stage (S-CNN (B)), only a square region that is taken from the central area of the leaf image is used to train the species group. In other words, we have two siamese nets, S-CNN (A) and S-CNN (B), similarly trained using different views in each taxonomic group (*genus-to-species strategy*).

Both S-CNNs are composed of two sub-networks that are one of the baselines CNN architectures presented in Section 3.2. Such twin CNNs are pre-trained on the ImageNet dataset and have shared weights. The original output layer of the pre-trained baseline CNN has 1,000 units that compute the class scores. Such an output layer is replaced by a layer that computes a distance metric between the last fully connected layers of each Siamese twin (Eq. 1). An image pair (X_{left}, X_{right}) is the input of the S-CNN (A) and (B). We fine-tune the pre-trained baseline S-CNN model with positive and negative pairs of leaf images, minimizing the loss function denoted in (Eq. 2). The positive and negative pairs are generated from each leaf plant dataset samples (see Table 3) to train the S-CNNs how to differentiate the leaf species. An overview of the iterative training algorithm [36] used to learn the deep metric models is presented in Algorithm 1.

Algorithm 1: Training algorithm.

```
   Input: $numEpochs$, training set ($X_T$, $y_T$),
          $batch\_size$
1  for $i = 1$ to $numEpochs$ do
2      $Batch = get\_batch\_pairs(X_T, y_T,$ batch_size$)$
3      for $j = 1$ to $batch\_size$ do
4          $X_{left}, X_{right}, y = Batch(j)$
5          $f_l = feature\_extraction\ (X_{left})$
6          $f_r = feature\_extraction\ (X_{right})$
7          $d_w = L1\_distance(f_l, f_r)$ using Eq. 1
8          $\delta = compute\_loss(d_w, y)$ using Eq. 2
9          $update\_net\_weights(\delta)$
10     end for
11 end for
```

As one can see, for each training epoch, a *Batch* structure containing image pairs and the corresponding labels (1-same class, 0-otherwise) is created using the image training dataset (X_T, y_T) (line 2). Then, the S-CNN extracts the feature vectors f_l and f_r from the images X_{left} and X_{right}, respectively. Line 7 computes the $L1$ distance between the extracted feature vectors (f_l, f_r), as denoted in Eq. 1, where M is the feature map size.

$$L1(f_l, f_r) = \sum_{k=0}^{M-1} ||f_{l,i} - f_{r,i}|| \tag{1}$$

According to [2], the $L1$ distance is consistently more preferable than other distance metrics (e.g., Euclidean and cosine distance) for high-dimensional vectors. For instance, when using a VGG16 to compose the S-CNN, we have feature vectors of 4,096 entries. Besides, the work that gave worldwide visibility for Siamese Networks uses the $L1$ distance [22].

The computed distance (d_w, in line 7) is the input of the last S-CNN layer in which the loss is computed (line 8), as described in Eq. 2.

$$Loss = -[y\log(d_{\mathrm{w}}) + (1-y)\log(1-d_{\mathrm{w}})] \tag{2}$$

where y is the image pair label (1 or 0). Finally, we update the network parameters (line 9).

3.2. Pre-trained CNN Baseline

We have selected a pre-trained CNN model to compose our Siamese Neural Network based on a set of experiments described in Section 4.3.1. The following CNN architectures were evaluated: AlexNet, GoogLeNet, and VGGNet.

Hinton and Alex Krizhevsky created AlexNet. The architecture of AlexNet used in this work is shown in Fig. 3. The network architecture contains eight weighted layers, the first five being the convolutional layers, and the remains are three (3) fully connected layers. Normalization and pooling layers follow the first two convolutional layers. A single pooling layer follows the last convolutional layer. The third, fourth, and fifth convolutional layers are connected directly. The second fully connected layer is provided to the softmax classifier with the numbers of class labels. ReLU, as the activation function of the first two fully connected layers (fc6, fc7), generates 4,096 values. Finally, the output of the seventh layer of 4,096 data is fully connected to the (n) neurons in the eighth layer (fc8), which (n) represents the number of classes. After training, it (fc8) outputs (n) floating-point values, the predicted result.

GoogLeNet was the winner of the ILSVRC 2014 competition, which carried out a top-5 error of 6.67%. It demonstrated the efficiency to deal with tasks extremely hard for humans, achieving good performance with a low error accuracy. GoogLeNet implements a different way of a network. It incorporates a modern section, which is named as inception module. The inception module uses variable receptive fields, which were created by different kernel sizes. These receptive fields created operations that captured sparse correlation patterns in the new feature map stack. GoogLeNet consisted of 22 layers in total, which was far greater than any network before it. For instance, GoogLeNet drastically reduces the number of parameters, which is only 1/12 of AlexNet. The overall structure of the GoogLeNet network is shown in Fig. 4. As shown in Fig. 4, the inception module (M1 until M9) is used on GoogLeNet. The main idea of inception is to localize an optimal local sparse structure and address it as an approximate dense component. The inception module adopts multiple convolutions (1×1, 3×3, 5×5) mixed with the max-pooling layer, which then combines the convolution and pooling results, making GoogLeNet different from a traditional multi-channel convolution. To avoid the heavy computation and over-fitting due to a large number of network parameters in the fully connected layers, it directly uses the strategies of averaging pooling and dropout after the inception module, which plays a role in reducing the dimension, as well prevents over-fitting to some extent. In this work, there are a total of nine inception models in GoogLeNet architecture. One can find a more detailed overview of this architecture in [40].

VGGNet, developed by Simonyan and Zisserman, was the runner up of the 2014 ILSVRC competition. VGG has been upgraded based on AlexNet. The entire network remains the size of 3×3 convolution kernel and 2×2 max-pooling sizes. The structure of VGG is shown in Fig. 5. VGG contains five blocks of convolutional layers interchanged with five max-pooling layers, followed by two fully connected layers and an output layer. The first two blocks have two convolutional layers with 64 and 128 filters, respectively. The other two blocks have three convolutional layers with 256, 512, and 512 filters, respectively. All filters have size 3×3, and the max-pooling layers have pool size and stride 2. VGG is the most popular option in the literature for extracting features from images. However, VGGNet has 138 million parameters, which can be a bit difficult to employ. In the literature, extensions of VGGNet are presented (VGG16 and VGG19). In our work was used VGG16 which consists of 16 deep convolutional layers.

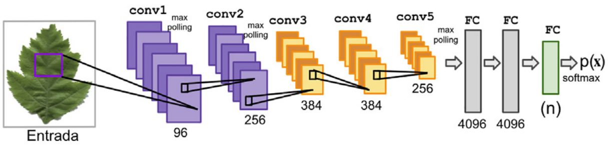


Fig. 3. The architecture of AlexNet.

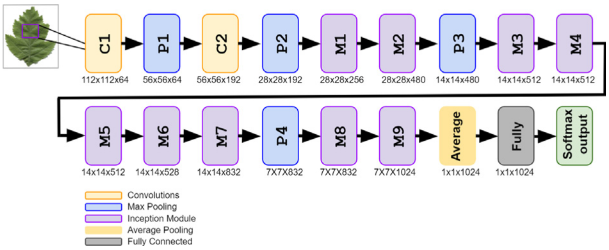


Fig. 4. The architecture of GoogLeNet.

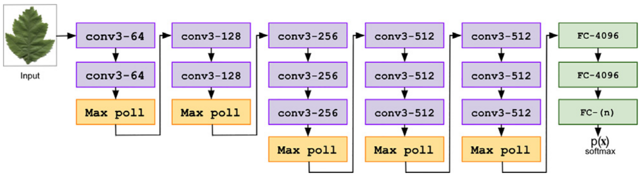


Fig. 5. The architecture of VGG16.

3.3. Hierarchical Classification Strategy

A coarse-to-fine classification is performed considering the hierarchical botanic taxonomy of plants. To this end, in the first stage of hierarchical classification schema in Fig. 2, the test leaf plant is classified using S-CNN (A) model according to its genus, then its species is defined with S-CNN (B) on the second stage. For each species, we have randomly selected some supervised samples as reference images. The number of reference images per species was experimentally defined (from 1 to 6). The reference images for each genus are those selected for each of its species. A ranked list (R_k) of the genus references is the output of the first stage, generating the k-best genus candidates of our hierarchical classification corresponding to the coarse classification step, which is based on the global representation of the leaf image. The k-best genus candidates are used to select the species to be evaluated in the second stage of the hierarchical classification, which is done considering the leaf image's local representation. Finally, the genus (coarse classification) and species (fine classification) are combined to produce a final ranked list F of the best plant species' hypotheses.

It is worth mentioning that the references for a genus are randomly selected from the training samples but contemplating each of its species. For this purpose, the selection algorithm tries to have at least one reference for representing each species inside the genus. For example, the genus *Prunus* has three (3) species, so the algorithm randomly took two (2) different samples of each species to form the set of six (6) references. It is important to say that the datasets used have a maximum of 6 species per genus. However, even if a genus has more species than the number of references considered in the system, we can see promising results (see our experiments with less than six references in Tables 7 and 8). In most cases, visual characteristics are relatively similar for families and genus in plant species context.

With this hierarchical classification, we can better deal with the inter-class and intra-class variations observed in the plant species context. For instance, in Fig. 6a, four samples of plants are plotted in the global view space without considering the plant hierarchy. In such a scenario, the discrimination must be carried out among all species, making more difficult the discrimination between species with similar characteristics, which increases the complexity of the problem. The hierarchical strategy alleviates the intra-class and inter-class problems, clustering the leaves which have similar characteristics. For instance, by grouping the similar leaves in Fig. 6b, we can see that the species belong to three different genus groups (G_1, G_2 and G_3). When the hierarchy is considered (the genus is used), the classification process becomes more manageable, mainly distinguishing species between the genus G_2 and G_3.

Besides, we observe that some species of the same genus may be a likeness, like those that belong to G_1. This motivation is why we proposed using an additional representation of leaf (local view) joined with *coarse-to-fine* hierarchical classification in this work. The rationale behind that is to perform some regional analyses of the leaf image (Fig. 6c) c) to deal with such a possible low variability in terms of shape and color between similar species inside some genus (G_1).

3.4. Fusion Schema

The final classification is given by the fusion of the outputs of the first and second hierarchical stages. The first stage's output is a ranking list R_k, representing the genus reference candidates for a given test sample X_l provided by the S-CNN (A) trained on the global view representation of the leaf image. It is important to remember that we can have up to six references by each genus, depending on the number of reference samples defined. Thus, R_k is a ranking list with k genus references, where each G_{ij} is the j_{th} reference of the genus i. Thus, we can compute each genus reference frequency that appears in R_k, which is used to weight (w) the output of the second stage. In the second stage, the similarity value S_i between the cropped test image (X_l) and the reference image (X_r) of species inside each genus G_i present in the ranking list R_k is provided by the S-CNN (B). Now, considering the local view representation of the leaf image. Finally, the second stage's output is a list of species F ordered by the score ζ, computed as described in Eq. 3.

$$\zeta = \frac{w_i \cdot S_i}{\sum w_i} \tag{3}$$

where w_i is the frequency weight of genus i_{th} computed in the first stage.

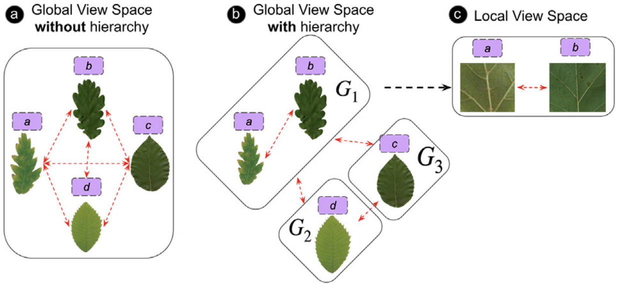


Fig. 6. a) Different species (a, b, c and d) with similar characteristics are plotted in the global view space; b) The similar categories are separated into three groups that represent the genus taxonomy (G_1, G_2 and G_3); c) Very similar species are discriminated in the local view space.

With such a fusion schema, we combine global and local views of the leaf image, considering a hierarchical strategy that reduces the number of species in the second stage, decreasing the classification complexity. The global score is the output of SCNN-A, while the local score is the output of SCNN-B. We evaluated different rules (sum, product, and majority voting) to combine the global and local scores produced in the system's first and second stages. The best results were observed when using the schema based on Eq. 3. As mentioned, the genus frequency is used as the weight for their respective species. The rationale behind this is that the correct species usually belongs to the genus that most appear in the candidates' list. Thus, we use such a frequency as a weight in the second stage, being important information in the final species decision.

4. Experimental Results

This section describes the datasets and experiments used to evaluate the proposed method.

4.1. Datasets

We have evaluated the proposed approach on two fine-grained datasets: PlantCLEF 2015 and LeafSnap. Table 1 presents the original number of classes, training, and testing images available in each dataset. The reason for choosing these two datasets is that both have a wide variability of plant leaf species, representing a challenging scientific task. These datasets' images were gathered by various photographers in globally distributed locations, engaging mixed conditions of background, position, color, and lighting, factors that significantly influence the images' quality. Moreover, both datasets are imbalanced, and for some plant species, there are few samples available for training. Since one of this paper's goals is to classify plant species with a small number of labeled samples, we have evaluated using only six images per class for training the models. It corresponds to the minimum number of training samples per species found in these datasets. Table 2 shows the total number of training images per taxonomic groups (families, genus, and species) when considering only six samples per class in each dataset to be used in the S-CNN proposed method.

We generated training subsets from the training images of Table 2, which have more non-similar pairs than similar ones, as recommended Melekhov et al.[28]. Table 3 shows the number of positive and negative samples in our training subsets. A positive sample pair (label 1) means that two images belong to the same category while a negative sample pair (label 0) means that two images belong to different categories.

The PlantCLEF 2015 dataset already has a pre-defined test set made up of 221 leaf images. On the other hand, for the LeafSnap dataset, we randomly choose 15 images per class to compose the test set, totaling 2,760 leaf images.

Table 1
Original numbers of classes and images for each taxonomic group of the PlantCLEF 2015 and LeafSnap datasets.

			Number of	
Dataset	Taxonomic		Training	Test
	group	Classes	Images	Images
PlantCLEF 2015	Family	29		
	Genus	43	6,527	221
	Species	60		
LeafSnap	Family	35		
	Genus	73	23,147	2,760
	Species	184		

Table 2
Original numbers of classes and images for each taxonomic group of the PlantCLEF 2015 and LeafSnap datasets.

		Number of			
Dataset	Taxonomic	Classes		Images	
	Group	Train	Test	Train	Test
PlantCLEF 2015	Family	205	29		
	Genus	260	43	12,605	221
	Species	351	60		
LeafSnap	Family	35	35		
	Genus	73	73	7,719	2,760
	Species	184	184		

4.2. Pre-processing

We use a two-step based pre-processing. First, we remove unwanted structures from the leaf, and then we crop the filtered image. The filtering process originally presented in [5] is used. First, a copy of the leaf image is converted to gray-scale, followed by a threshold operation using Otsu's method. Afterward, a top-hat technique is applied to remove unwanted objects as the leaf stem. Finally, the bounding box is used to detect just the leaf from the filtered image. The final image resolution was 224×224 pixels.

In the second step, the image containing just the leaf is cropped on its center using the algorithm proposed in [8]. We used different cropped image resolutions like 32×32, 64×64, and 128×128. The idea is to evaluate the performance of the S-CNN architecture considering different input resolutions.

4.3. Analysis and Experiments

We start this section by presenting four critical analyses that were necessary to define the proposed method. Section 4.3.1 evaluates the CNN architectures (AlexNet, GoogLeNet, and VGG16Net) considered alternatives to composing the used S-CNN. Section 4.3.2 is used to design the coarse-to-fine hierarchy classification. In Section 4.3.3, the configuration of the two-view hierarchical classification is defined. Following, Section 4.3.4, a fourth analysis shows the importance of the proposed two-view representation of the leaf images.

Additional experiments were performed to evaluate the proposed method. In Section 4.3.5, we have an overall performance evaluation. Section 4.3.6 shows some analysis using the proposed method for plant species recognition. Besides, we evaluated the proposed approach regarding the impact of unbalanced data (Section 4.3.7), scalability, and stability (Section 4.3.8). Finally, in Section 4.3.9, we compare the proposed method with the state-of-the-art.

For the PlantCLEF 2015 dataset, the overall results are computed using the average classification score S proposed in [19]. The S metric is defined in Eq. 4 and represents a score related to the rank of the correct species in the list of retrieved species.

Table 3
The number of positive and negative samples of the training subsets used in the proposed S-CNN method.

Dataset	Taxonomic Group	Positive Samples	Negative Samples
PlantCLEF 2015	Family	200	300
	Genus	400	600
	Species	800	1,200
LeafSnap	Family	300	450
	Genus	600	900
	Species	1000	1,500

$$S = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{P_u}\sum_{p=1}^{P_u}\frac{1}{N_{u,p}}\sum_{n=1}^{N_{u,p}}S_{u,p,n} \tag{4}$$

where U is the number of users (who have at least one image in the test data), (P_u) is the number of individual plants observed by the u-th user, ($N_{u,p}$) is the number of pictures taken from the p-th plant observed by u-th user, and ($S_{u,p,n}$) is the score between 0 and 1 which is equal to the inverse of the rank of the first correct math for the n-th pictures taken from the p-th plant observed by u-th user.

For the LeafSnap dataset, the recognition accuracy is computed by Eq. 5.

$$acc = \frac{\text{number of correctly classified samples}}{\text{total number of samples}} \tag{5}$$

4.3.1. *Evaluating different pre-trained models*

The recognition accuracy of AlexNet, GoogLeNet, and VGG16 is shown in Table 4. The last dense layer of each CNN was adapted to contemplate the number of classes (plant species) of each dataset. Stochastic gradient descent (SGD), with a momentum of 0.9, was used to fine-tuning these deep models. Besides, a learning rate of 0.001 was set with a decay of 0.5 at every 512 iterations in a total of 2,048 iterations. Hyper-parameters like batch size and the number of iterations were defined empirically as 32 and 2,048, respectively. We consider the classification of species based on entire images (global view). The experiments were conducted on the PlantCLEF 2015 dataset using the original dataset presented in Table 1. For the results on Tables 4–6, the original training dataset was divided using a randomly stratified strategy into training (70%) and validation (30%) subsets. We can observe that VGG16 provided

Table 4
Recognition accuracy (%) of different architectures considering entire images (just global view).

Taxonomic group	Model	Accuracy (%)
Species	AlexNet	71.98
	GoogLeNet	73.30
	VGG16	75.09

Table 5
Individual classification considering each taxonomic group: Family, Genus and Species and different views: global and local with multiple-resolutions: 32×32, 64×64 and 128×128.

Taxonomic	Accuracy (%)			
Group	Global View		Local View	
		32×32	64×64	128×128
Family	69.51	63.01	60.47	59.16
Genus	85.89	67.33	64.41	62.35
Species	75.09	74.00	72.35	71.18

Table 6
Diverse combination of *coarse-to-fine* hierarchical classification, according with taxonomic groups and representation of plant leaf views.

Model	ID	Hierarchical	View		Accuracy
	#	combination	Global	Local	(%)
VGG16	1	Family + Species		√	60.21
	2	Family + Species	√		62.76
	3	Family + Species	√	√	67.55
	4	Genus + Species		√	64.33
	5	Genus + Species	√		78.98
	6	Genus + Species	√	√	83.11

better results under the same experimental conditions, showing an accuracy of 75.09%.

4.3.2. ***Plant family, genus and species classification***

This section evaluates how to define the coarse and fine levels of the proposed hierarchical strategy. To this end, we performed the classification of leaf images using the pre-trained VGG16 model, taking each taxonomic group (family, genus, and species) individually and considering different views: global and local. Table 5 shows the results observed on the PlantCLEF 2015 validation dataset.

As we can see, the local view individually has a worse performance than the global one for all taxonomic groups, indicating that we must avoid the local view in the first level (coarse) of the hierarchy. Table 5 demonstrates that the VGG16 model is better when using the genus compared to family and species. To better understand the results of plant species recognition per each taxonomic group, confusion matrices are presented in Figs. 7 (i) and 8 (i). They consider family groups for better visualization and only two plant families (*Salicaceae* and *Rosaceae*) with the corresponding species. Still, in Table 5, we can see that for the local view representation is better the use of cropped images of resolution 32×32 compared to 64×64 and 128×128.

Fig. 7 (i) shows the confusion matrix using the families *Salicaceae* and *Rosaceae* of the PlantCLEF 2015 test set. The worse accuracies were observed for the species *S.cinerea*, *C.monogyna*, and *S.torminalis*. We notice that the family model cannot properly classify the leaf species that have different types of features inside the same family. The *S.cinerea* (Fig. 7 (D)) provided a false negative error with *C.germanica* (Fig. 7 (F)). This confusion may have been caused by the green body color and fine shape, which is quite different from the remaining species inside the *Salicaceae* family. On the other hand, *C.monogyna* (Fig. 7 (E)) and *S.torminalis* (Fig. 7 (G)) have characteristics dissimilar from leaves inside of *Rosaceae* family, causing misclassification. In any case, we can see that the confusions inside the family group are between species that contain different attributes in terms of color, shape, and features.

Fig. 8 (i) shows a confusion matrix for the same two families *Salicaceae* and *Rosaceae*. Using a VGG16 trained on the plant genus, instead of on the family, we separate the species in genus groups represented by the dotted lines in Fig. 8 (ii). By separating them, we have inside the *Salicaceae* family, two genus (*Populus* and *Salix*) and for *Rosaceae* family, three genus (*Crataegus*, *Sorbus* and *Prunus*) were formed. We observe in Fig. 8 (i) that the VGG16 model trained with genus increases recognition performance due to a better leaf features separability. However, the greatest confusions occur with similar species inside the same genus (highlighted with red dotted lines in the Fig. 8 (i)). Normally, the species inside the same genus have similar visual characteristics, such as the shape. For instance, the three most similar species are *P. nigra* (Fig. 8 (A)), *P.alba* (Fig. 8 (B)) and *P. tremula* (Fig. 8 (C)) represented by the genus named as *Populus*. This similarity is reflected in local confusions since these three species have curved boundaries and the same shape sketch. Despite the similarity between species, the hits in the confusion matrix provided by the VGG16 trained on the genus (Fig. 8 (i)) is better from that considering the VGG16 model trained on family (Fig. 7 (i)).

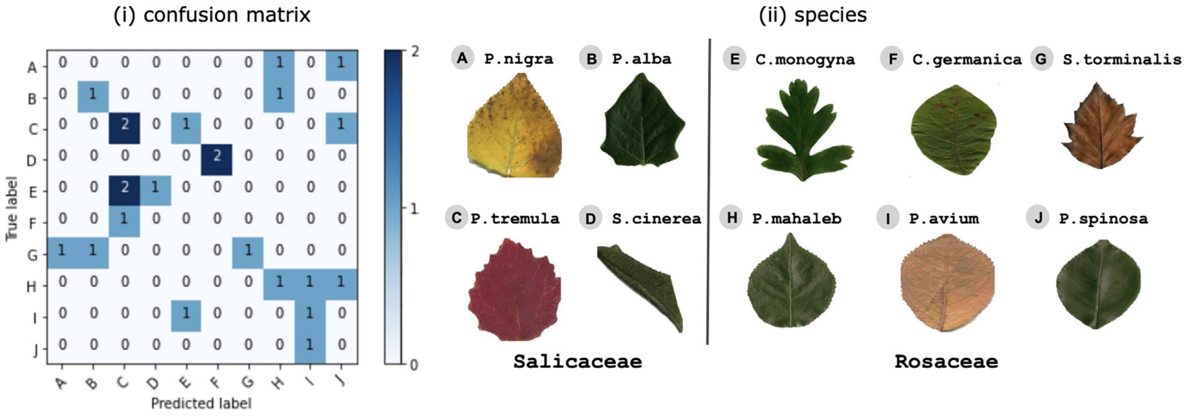


Fig. 7. Performance of VGG16 trained on Family groups: (i) confusion matrix (ii) Species separated by Families: *Salicaceae* and *Rosaceae*.

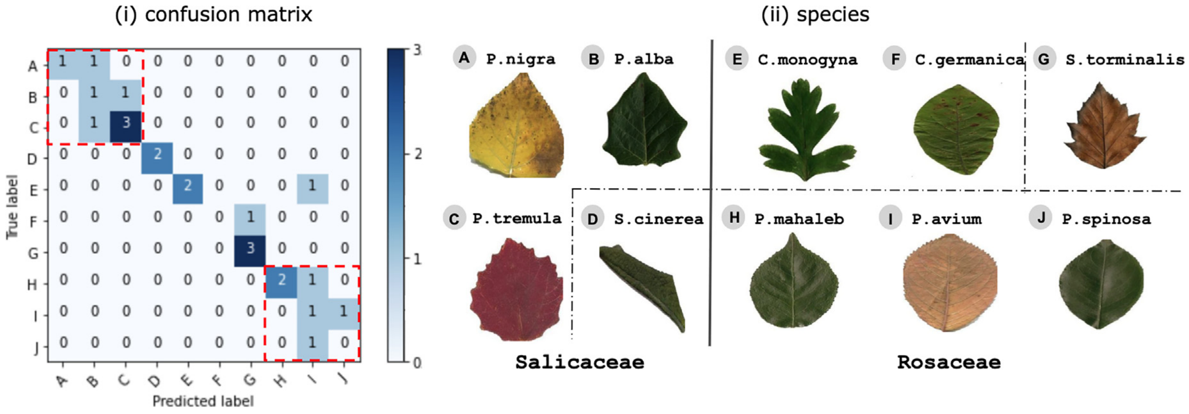
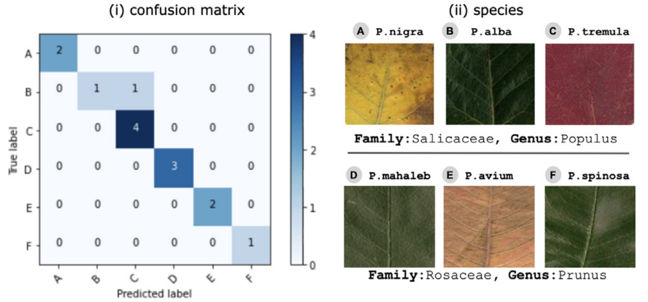


Fig. 8. Performance of VGG16 trained on Genus groups: (i) confusion matrix (ii) Species separated by Families: *Salicaceae* and *Rosaceae* and Genus: *Populus*, *Salix*, *Crataegus*, *Sorbus* and *Prunus*.



	A	B	C	D	E	F
A	2	0	0	0	0	0
B	0	1	1	0	0	0
C	0	0	4	0	0	0
D	0	0	0	3	0	0
E	0	0	0	0	2	0
F	0	0	0	0	0	1

Fig. 9. Performance of VGG16 trained with local-view representation on Genus groups: (i) confusion matrix (ii) Species separated by Families: *Salicaceae* and *Rosaceae* and Genus: *Populus* and *Prunus*.

These preliminary experiments indicate that the coarse-to-fine strategy must consider the genus in the first level of the hierarchy strategy instead of family or species. The results are also better using global view in the first level than local one as shown in Table 5. However, there are still mistakes related to species very similar in appearance (like those presented in red dotted line in Fig. 8 (i)), that require additional effort to be classified. To deal with this problem, we propose the two-view representation of the leaf image in which global and local features are combined in a hierarchical classification schema.

4.3.3. *Configuration of two-view hierarchical classification*

Table 6 shows the use of the taxonomic groups in a diverse pool of hierarchical combinations and which view is used in each level of the hierarchy. For instance, in ID #1, we evaluate the hierarchy classification using Family + Species, which means the Family taxonomic group is used to train the VGG16 model in the first level of the hierarchy, and the Species taxonomy group is trained posteriorly. So, the solution is tested in a *coarse-to-fine* hierarchy way. In the fourth column of Table 6, we choose which type of view (global or local) is used in each level of the hierarchy to evaluate diversified combinations. Whether two marks are selected in the fourth column means that the global view is used in the *coarse* level of the hierarchy and local view in the *fine* one. Otherwise, one mark, only the marked view is used in both levels.

The lowest performances are achieved when the combinations are carried out with the family taxonomic group as the first level of the hierarchical classification (ID #1, ID #2, and ID #3), even though using different views. The individually poor performance for the family taxonomic group achieved in Table 5 makes all combinations that use the family group have the worst accuracy. Because of a hierarchy classification propriety, the error persists in the following classification levels independent of which taxonomy or type of view is used. The ID #4, ID #5, and ID #6 start by using genus in the first level (*coarse*) of the hierarchical classification and species secondly (*fine*). The performance has a soft advance with ID #5 and ID #6. Despite ID #4 remains the poor performance due to the use of local view in both levels of hierarchy, ignoring discriminating characteristics like leaf shape and boundary. ID #5 uses global features for both levels. The errors occur in the second level. Inside the same genus, species with similar characteristics are hard to discriminate using global view only. Finally, ID #6 brings the best result by use two-view in a hierarchical classification strategy.

Assuming the results without hierarchical classification in Table 5, we have a performance of 75.09% using species as final output. Using the hierarchical combination (Table 6), we increase the accuracy by 3.89 percentage points with ID #5 for the same case (global-view), confirming the assumption that the hierarchy strategy can improve the performance of classification. Besides using ID #6, we increase by 8.02 percentage points, further enhancing the classification by adopting a local view of the plant leaf image in a two-view hierarchical *coarse-to-fine* classification solution.

Finally, in Fig. 9, we evaluate the impact of using the local view in the second level of the hierarchy. To show the effectiveness, we plot the confusion matrix for specific cases in which the error persists when used just global views. The specific cases have already been presented in Fig. 8 (i) highlighted by dotted red lines and were plotted in a new confusion matrix in Fig. 9 (i) with a new representation (local view) in the Fig. 9 (ii). We noticed an increase in the performance results in almost all species when the local view is employed. This occurs because, in terms of global features (general characteristics), the species inside the same genus (red dotted lines in Fig. 8 (ii)) are very similar. However, when plotted in a local view (Fig. 9 (ii)), they present tiny and discriminate differences among species. The performance using local view extract features in detail (patterns veins and texture of leaf plant) able to distinguish similar species. Comparing with ID #5 in Table 6 that uses just global view in both levels, the results improve by 4.13 percentage points when adopting the local view in the second level of hierarchy (ID #6). In these experiments, we consider the top class returned in the hierarchical classification to pass to another level. Thus, after this previous analysis, we defined our S-CNN proposed approach as genus-to-specie hierarchical classification, using genus as coarse classification and species as the fine classification. Employing global view and local view for the first and second levels of hierarchy respectively as seen in the scheme of Fig. 2.

4.3.4. *Analysis of the two-view representation*

As we may see in Table 6, the two-view representation (ID #6 in that table) is a promising alternative to boost the classification performance for the proposed hierarchical classification. With the experiments presented in the last two sections, we gain two essential intuitions regarding leaf features. Firstly, leaf shape alone is not the right choice for identifying plants because of the common occurrence of similar leaf contours, especially in closely related species. In these situations, the venation pattern is a more powerful discriminating feature. Secondly, the strategy of combining global and local features presented in the leaf images is promising. However, what does CNN learns in each view (entire image and cropped image)? To answer this question, we explored the CNN layers of each view in Fig. 10.

We observed that, at each CNN layer, similar information is extracted but from different perspectives. Similarly, in the sense that both models provide in the first layers more low-level features, while in the deeper layers, we can observe more class-specific features. Different perspectives since the models have as inputs different views of the plant leaf. For instance, the first convolutional layer tends to extract low-level features like borders. However, it is clear the complementarity between the two networks. The global view provides the whole leaf's edges (shape information), while the local view provides venation-like features. Similar complementarity can be visually observed between each

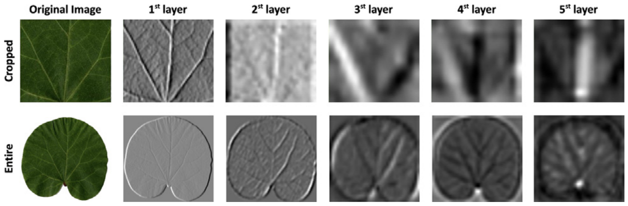


Fig. 10. Two-view representation - the feature maps along the layers of the S-CNN for entire and cropped images.

corresponding layer. In general, the entire leaf image (global view) provides features related to the leaf image's general shape and texture. On the other hand, the features extracted from the cropped leaf image (local view) tend to capture inherent local patterns of venation. Such a general-to-specific leaf representation allows us to combine information from the leaf shape and venation, as usually done by plant taxonomists.

4.3.5. ***S-CNN performance evaluation***

In this section, the proposed S-CNN hierarchical classification based on a two-view similarity scheme is evaluated. Tables 7 and 8 present the results for PlantCLEF 2015 and LeafSnap datasets, respectively. The best results for both datasets were achieved using six reference samples per class (N_r=6) and 30 genus reference candidates in the ranked list R_k. In the second stage, the accuracy is computed considering the first result returned in the list F of species (top-k = 1). We reached a 1.0 accuracy rate with top-k = 5 for both datasets as reported in Table 9.

It is important to notice that the number of references N_r has a significant impact on the results. Therefore, we show in Fig. 11 different N_r values over different genus references candidates R_k. It is expected that as the ranking list R_k grows, the performance should increase. However, we notice that there is a decrease in performance when using R_k=50. This is directly related to the *coarse-to-fine* hierarchical classification, in which at the coarse stage, we define how many species references will be taken to the fine stage using the genus reference candidates that appear in the ranking list R_k.

Table 7
Average classification score (S) of the S-CNN proposed method for the PlantCLEF 2015 dataset.

Hierarchical Stage	(R_k)	Number of References (N_r)		
		1	3	6
1st (Genus)	5	0.72	0.77	0.77
1st (Genus)	15	0.86	0.84	0.81
1st (Genus)	**30**	**0.98**	**0.96**	**0.95**
1st (Genus)	50	0.99	0.98	0.98
2nd (Species)	1	0.81	0.86	0.87

N_r: number of reference samples per class in the classification phase.

Table 8
Overall accuracy (acc) of the S-CNN proposed method for the LeafSnap dataset.

Hierarchical Stage	(R_k)	Number of References (N_r)		
		1	3	6
1st (Genus)	5	0.91	0.92	0.95
1st (Genus)	15	0.96	0.96	0.98
1st (Genus)	**30**	**0.99**	**0.98**	**0.98**
1st (Genus)	50	0.99	0.98	0.97
2nd (Species)	1	0.91	0.95	0.96

N_r: number of reference samples per class in the classification phase;

Table 9
Final accuracy of the S-CNN proposed method considering N_r=6, R_k=30 in the first stage (genus) and top-k=1, 3 and 5 in the second stage (species). VGG16 baseline is comparable for each dataset using top-5 results.

Dataset	Method	Top-1	Top-3	Top-5
PlantCLEF 2015	S-CNN	0.87	0.94	1.0
	VGG16	0.78	0.81	0.85
LeafSnap	S-CNN	0.96	0.99	1.0
	VGG16	0.88	0.90	0.93

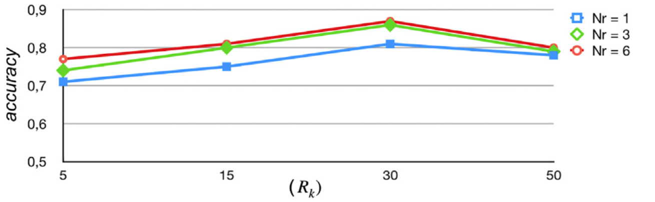


Fig. 11. Classification accuracy considering different number of references (N_r) and sizes for the ranking list of genus reference candidates (R_k) in the first stage (*coarse*) of hierarchy using the PlantCLEF 2015 dataset.

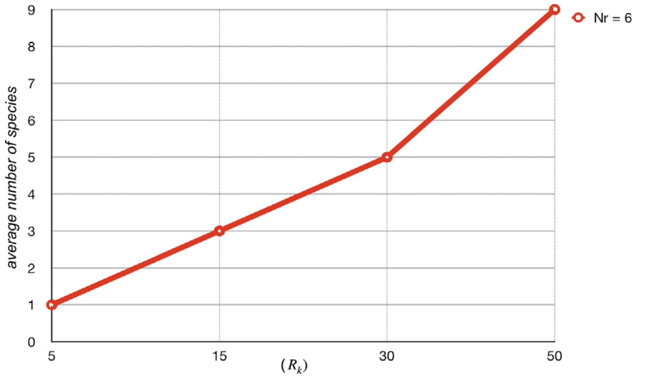


Fig. 12. Average number of species provided by the coarse stage, when using (N_r=6) concerning different sizes of genus references candidates returned in the ranking list (R_k).

The idea behind Fig. 12 is to show the number of species (in average) that is sent to the second stage of the system when considering different sizes (k values) for the ranking list R_k of genus candidates generated in the first system stage. The species selected are those that belong to the genus candidates in the list. However, the number of species may vary according to list size. For instance, with $N_r = 6$ (number of references), the average number of species processed in the second stage for $R_k = 50$, 30, and 15 is 9, 5, and 3, respectively. Such an analysis gave us some idea about decreasing the classification complexity in the second stage when we reduce the size of the list of genus candidates.

4.3.6. *Analysis of the S-CNN proposed method*

Fig. 13 shows a spider web chart in which one can see the percentage of correctly recognized leaf images by each species considering the S-CNN proposed method using just one-view (global representation in both levels of hierarchy), with two views (global and local representations) and the pre-trained VGG16 model. We considered the S metric results on the PlantCLEF 2015 test dataset. According to Fig. 13, the S-CNN two-view proposed method (red line) improves the classification rates of several species compared to the one-view purpose (blue line). The use of the local view applying cropped leaf images was suitable to reduce the conflict between species, corroborating with the preliminary experiment handled in Section 4.3.3. For instance, the eighth species, named *quercus cerris*, has 0.0 of accuracy when using just the one-view representation. On the other hand, it increased to 0.55 when we used the two-view proposed method. As we can see, when just one view is considered, the *quercus cerris* is confused with *quercus petraea*, *quercus rubra*, and *quercus pubescens*. Fig. 14 shows the leaves *quercus cerris*, *quercus petraea*, *quercus rubra*, and *quercus pubescens* species respectively. Noticeably, the four species in the Fig. 14 have similar morphological characteristics, which explains the confusion when just one view is adopted.

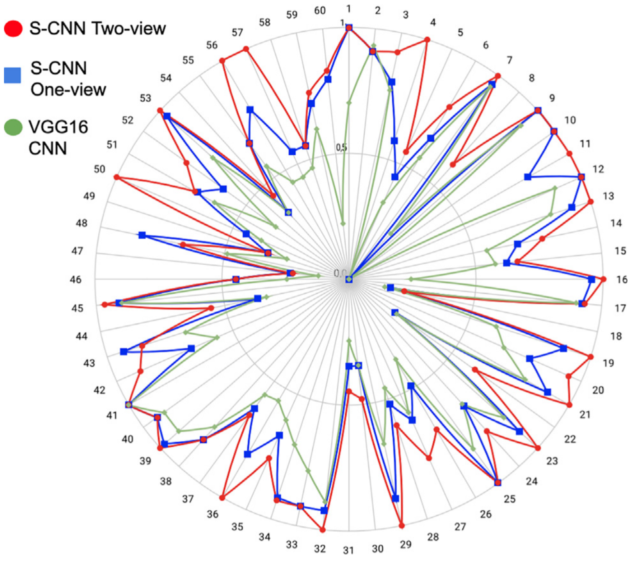


Fig. 13. Comparison of the proportion of correctly recognized leaves of each species at the coarse-to-fine classification using one-view (S-CNN one-view), the proposed method using two-view (S-CNN two-view) and pre-trained VGG16. 60 species are evaluated from PlantCLEF 2015 dataset.

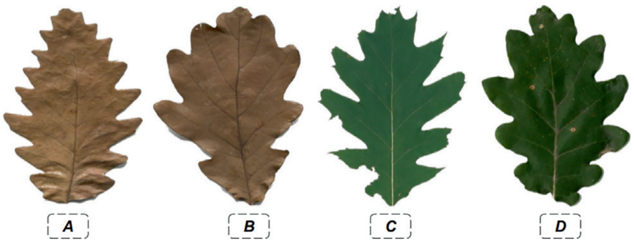


Fig. 14. Samples of confused species (global representation): a)*quercus cerris*; b) *quercus petraea*; c)*quercus rubra*; d)*quercus pubescens*.

However, it is important to observe that the accuracy has dropped off for few species using the S-CNN two-view proposed method. This is the case of the forty-eighth species in Fig. 13. It is the *betula pendula* specie. Observing the output of the two-view proposed method for related species, we found confusion between it and the *betula betulus* and *betula avellana*. They have very similar texture and vein patterns as shown in Fig. 15.

Finally, we compare S-CNN approaches with the pre-trained VGG16 method presented in Section 4.3.3. Table 10 shows the final accuracies of VGG16, S-CNN two-view proposed method, and the S-CNN using one-view on the test set for both datasets: PlantCLEF 2015 and LeafSnap. The final results improved the recognition rates for the two-view S-CNN, in which we assure the effectiveness of the proposed method.

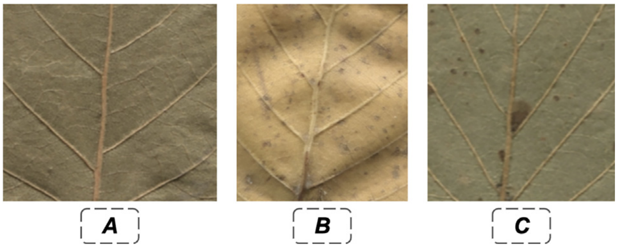


Fig. 15. Samples of confused species (local representation): a)*betula pendula*; b) *betula betulus* and c)*betula avellana*.

Table 10
Final accuracies for VGG16, S-CNN using one-view (global) and S-CNN proposed method for each dataset, PlantCLEF 2015 and LeafSnap.

Approach	PlantCLEF 2015 (*S*)	LeafSnap (*acc*)
VGG16	0.78	0.88
S-CNN one-view	0.81	0.92
S-CNN two-view proposed method	**0.87**	**0.96**

4.3.7. *Impact of unbalanced data*

Table 11 shows the impact of unbalanced data in the species accuracy when using the S-CNN proposed method and the pre-trained VGG16 model. As we can see, the second and third columns show the number of training images per species used for both models. It is important to say that the VGG16 model is trained using the original number of training samples shown in Table 1 for the PlantCLEF 2015 dataset, while the S-CNN model is trained using only six samples per class.

We observed that for almost all species, the performance of the S-CNN model is equal or better than the VGG16. Most of the mistakes of the VGG16 model occur in the species with few samples for training. As one may see, *L. triphylla* species (bold line in Table 11) has nine samples for training and seven samples for testing, it is most difficult for the VGG16 to generate an appropriate model to discriminate *L. triphylla* among all species with so few samples for training. On the other hand, for the same case (*L. triphylla* species), the S-CNN model trained with only 6 samples was able to recognize 6 out of 7 test samples correctly. The reason is that the S-CNN is not trained to learn a regular classifier of plant species but a distance metric to provide the similarity between two images (reference and test image). Therefore, the S-CNN model is capable of dealing with unbalanced data. Consequently, the S-CNN uses a total of 360 images to train the model employing just six samples per class, and achieved a total of 182 hits. In contrast, the VGG16 hit 147 test samples using 6,527 training samples with an unbalanced number of samples per class.

We also performed experiments considering the unbalanced data on the LeafSnap dataset. Table 12 shows the use of different sizes of training samples, starting from 1, 3, 6, 10, 15, 25, and between 30 to 300 samples per species. For testing the LeafSnap dataset, we select randomly 15 samples by each class to compose the testing set. Concerning the first experiment (#1) in Table 12, it is difficult to get a good classifier in both cases since the S-CNN needs more than one pair of positive and negative samples to train and update the loss function to converge the model, while the VGG16 model requires a substantial number of training samples to provide solid results. The main advantage of the S-CNN is observed in the experiments #2 and #3, in which it outperforms widely the VGG16 using few training samples. The performance of the S-CNN starts to fall in the experiments #4, #5, #6, and #7, due to a huge quantity of image pairs (positive and negative) generates to train, causing over-fitting in the Siamese network. Differently, the performance of the VGG16 starts to increase in the experiment #6, achieving 0.84 of accuracy (*acc*) when using 25 training samples per species. In experiment #7 we have the best result for VGG16 with 0.88 of accuracy, when between 30 and 300 training samples were used. However, S-CNN outperforms the VGG16 in experiment #3, reaching 0.96 of accuracy using just six training samples per class. These last experiments corroborate the ability of the S-CNN to deal with unbalanced data.

4.3.8. *Stability and Scalability*

Since the final result may be affected by the random selection of reference samples used to represent the plant species, we have

Table 11
Species and number of training and test images from the PlantCLEF 2015 dataset. Hits are the quantity of correct prediction per species held by the proposed S-CNN approach or VGG16.

	Training			Hits			Training			Hits	
Species	VGG16	S-CNN	Test	VGG16	S-CNN	Species	VGG16	S-CNN	Test	VGG16	S-CNN
V. opulus	60	6	9	8	8	M. papyrifera	115	6	1	1	0
V. tinus	243	6	3	3	3	M. carica	116	6	1	1	1
L. styraciflua	61	6	3	3	3	M. rubra	88	6	1	1	1
A. prostrata	6	6	1	0	1	Fra. excelsior	60	6	9	4	5
A. cotinus	165	6	4	3	3	Fra. ornus	73	6	1	1	1
A. pistacia	157	6	1	1	1	Fra. vahl	132	6	7	4	4
I. aquifolium	84	6	1	1	1	O. europaea	311	6	1	1	1
H. helix	264	6	9	5	6	S. bulgaris	101	6	3	2	3
R. aculeatus	290	6	25	22	23	P. hispanica	119	6	5	4	4
A. trichomanes	6	6	1	0	1	C. monogyna	197	6	3	2	2
A. vulgaris	6	6	1	0	1	C. germanica	50	6	1	1	1
B. glutinosa	51	6	2	2	2	P. avium	67	6	2	2	2
B. pendula	122	6	6	2	4	P. mahaleb	56	6	3	2	3
B. betulus	124	6	4	3	3	P. spinosa	46	6	1	1	1
B. avellana	148	6	2	1	1	U. minor	382	6	2	2	2
C. australis	190	6	4	1	1	Po. alba	197	6	2	1	2
F. cercis	142	6	2	2	2	Po. nigra	222	6	2	1	2
F. robinia	109	6	3	3	3	Po. tremula	97	6	4	3	4
Q. cerris	125	6	9	2	5	S. cinerea	24	6	2	2	2
Q. sylvatica	96	6	2	1	2	A. pseudo.	44	6	3	2	3
Q. pubescens	104	6	6	3	2	A. sacchar.	42	6	5	2	4
Q. sativa	77	6	2	2	1	A. negundo	111	6	1	1	1
Q. petraea	76	6	1	1	1	A. platanoides	67	6	5	3	5
Q. rubra	48	6	2	1	2	A. campestre	160	6	13	9	12
G. genarium	19	6	1	1	1	A. monspess	167	6	1	1	1
Gi. biloba	127	6	9	8	9	**L. triphylla**	**9**	**6**	**7**	**2**	**6**
L. nobilis	151	6	3	2	3	A. altissima	82	6	4	1	3
L. tulipifera	70	6	2	1	2	T. baccata	10	6	2	1	2
T. tilia	71	6	4	2	3	S. torminalis	36	6	3	3	3
T. cordata	28	6	3	1	2	B. davidii	126	6	1	1	1

Table 12
Accuracy achieved for different quantity of training images per class (data size) using S-CNN proposed method and VGG16 model for LeafSnap dataset.

Experiment	Dataset Size		S-CNN	VGG16
	Training	Testing	(*acc*)	(*acc*)
#1	1	15	0.51	0.53
#2	3	15	0.85	0.52
#3	6	15	0.96	0.53
#4	10	15	0.94	0.65
#5	15	15	0.92	0.78
#6	25	15	0.91	0.84
#7	[30–300]	15	0.88	0.88

evaluated the stability of the proposed method in such a situation. With that in mind, we performed five executions of the proposed method with different plant samples as references (without repetitions). We performed a data augmentation for species with few training samples using rotations in the original leaf image as presented in [5]. Table 13 shows the results of the five executions for the PlantCLEF 2015 and LeafSnap test sets. As one can see, the method performance is stable even using different sets of references.

The scalability of the proposed method can be evaluated by considering plant species unseen during the training step. For such an aim, in the proposed method, it is just necessary to add reference images of the new species to be considered. Therefore, one of the proposed method's main advantages is that it does not require retraining the S-CNNs, avoiding such a time-consuming process.

Table 13
Executions with distinct sets of reference images, six references are randomly selected to compose the reference sets for each dataset, PlantCLEF 2015 and LeafSnap. Av = average performance.

Dataset	Execution					
	1	2	3	4	5	**Av**
PlantCLEF 2015	0.87	0.86	0.84	0.85	0.87	**0.86**
LeafSnap	0.96	0.94	0.93	0.95	0.93	**0.94**

We experimentally evaluated the impact on the system recognition performance by considering incrementally new plant species not seen during training. For this purpose, we prepared small subsets with 12, 50, 100, 150, and 184 plant species using the LeafSnap dataset. For each new subset added, we computed the performance of the system originally trained on the PlantCLEF 2015 dataset (60 species), with and without a retraining process. As expected, we always observed the best performance with the system retraining. However, we noticed a small loss in performance (up to 0.4) when considering unseen species, as shown in Fig. 16. As we can see, the accuracy before considering unseen classes was 0.87 for the 60 classes of the PlantCLEF 2015 dataset. After adding 12, 50, 100, 150, and 184 new species, which belong to the LeafSnap dataset, the accuracy dropped to 0.86, 0.85, 0.83, 0.82, and 0.81, respectively. It is important to note that even adding 184 unseen species (new ones) from another different dataset, the proposed method sustained an accuracy close to that achieved

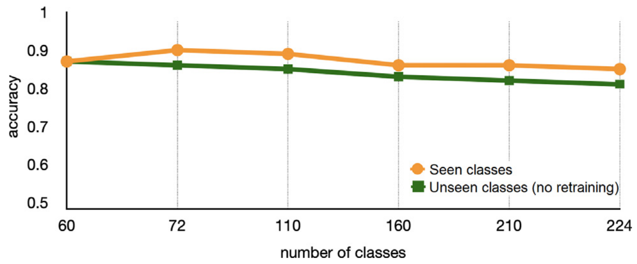


Fig. 16. System scalability considering unseen and seen leaf species.

for only 60 species. This behavior is impressive since the model never saw these new species during training. The S-CNNs can compute the similarity between the new references and the test images. With these results, we can say that the proposed method scales relatively well. Besides, they also indicate that the system performed well in the case of a cross dataset evaluation.

Table 14 shows the necessary time to classify a single leaf image for a growing number of species. For calculating the computational time, we look at just the test phase. We observed that the computational time grows slower than the linear function as the number of reference images grows.

4.3.9. ***Comparison with the State-of-the-Art***

We compared the result achieved by the proposed method with those achieved by related works, which have also used either PlantCLEF 2015 or LeafSnap datasets, as shown in Table 15. Six out of eight of the recent approaches for plant recognition use CNN models [39,27,16,5,7,9], while the other two works use S-CNN models [47,43]. As we can see in Table 15, in a previous work [5], we reported almost the same results with a CNN approach on PlantCLEF 2015 dataset. It worth mentioning that there are no significant differences regarding the accuracy. However, the novel approach proposed here works better in unbalanced data scenarios, using the S-CNN architecture trained on few samples, which avoid a substantial computational cost of using a data augmentation strategy as previously employed in [5]. Another important improvement is that the proposed method is easily scalable, and new plant species can be easily integrated into the S-CNN models without retraining them.

In [47] to address the fine-grained problem, a piece of additional information is used considering multi-views of the plant organs like a leaf, entire plant, and flowers. Our method uses just the leaf features to extract two different representations (global and local features) that minimize the difficulty related to species with similar characteristics. Wang and Wang [43] showed the power of S-CNNs to represent unbalanced categories using a small subset of samples per class. However, the authors evaluated the experiments using just ten (10) plant classes with different small training subsets (5, 10, 15, and 20 training samples), whose better results were observed with twenty (20) samples by class. Differently, our S-CNN based method uses only six (6) training samples per class and is evaluated on well-known datasets with a huge quantity of classes. The use of the S-CNN with two-view representation in a hierarchical classification helps increase the performance using a minor subset of training samples. Differently from all these methods, we have combined a fine-grained strategy with a two-view representation of the leaf image. As one may see, the proposed method outperforms the related works, besides producing a scalable solution.

Table 14
Computational time for classify one leaf plant considering the scalability of classes.

Number of Classes	Dataset	Time (sec)
60	PlantCLEF 2015	0.2010
72	PlantCLEF 2015 + LeafSnap	0.3965
110	PlantCLEF 2015 + LeafSnap	0.9276
160	PlantCLEF 2015 + LeafSnap	1.3830
210	PlantCLEF 2015 + LeafSnap	1.6789
244	PlantCLEF 2015 + LeafSnap	1.8458

Table 15
Comparison with the state-of-the-art for leaf classification for PlantCLEF 2015 and LeafSnap datasets.

Reference	Approach	PlantCLEF 2015 (*S*)	LeafSnap (*acc*)
Sungbin [39]	CNN	0.76	-
Lee et al. [27]	CNN	0.80	-
Araújo et al. [5]	CNN	0.86	-
Ghazi et al. [16]	CNN	0.84	-
Zhi-Yong et al. [47]	S-CNN	0.84	-
Barré et al. [7]	CNN	-	0.86
Bodhwani et al. [9]	CNN	-	0.93
Wang and Wang [43]	S-CNN	-	0.91
Proposed Method	**S-CNN**	**0.87**	**0.96**

5. Conclusion

We proposed a novel method based on a two-view leaf image representation and hierarchical classification strategy for fine-grained plant species recognition. The botanical taxonomy is used to drive a coarse-to-fine classification strategy to identify the plant genus and species. The two-view representation of a plant leaf can improve recognition performance using global (shape and color) and local features (texture and plant veins). A deep metric based on an S-CNN was used to reduce the dependence of the proposed method on a large amount of training samples. Besides that, the S-CNN makes the proposed method scalable and new plant species can be easily integrated into the S-CNN models without the need of retraining them.

The experiments on two challenging fine-grained datasets of leaf images (PlantCLEF 2015 and LeafSnap) confirmed the effectiveness of the proposed method - the recognition accuracy over those two datasets reaches 0.87 and 0.96, respectively. As future work, we plan to deal with auto-encoders to learn representations inside an S-CNN architecture with hierarchical property for leaf plant recognition.

CRediT authorship contribution statement

Voncarlos M. Araújo: Data curation, Methodology, Software, Writing - original draft, Visualization, Investigation. **Alceu S. Britto Jr.:** Conceptualization, Writing - review & editing, Supervision, Resources, Funding acquisition. **Luiz S. Oliveira:** Conceptualization, Supervision. **Alessandro L. Koerich:** Conceptualization, Resources, Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Aakif, M.F. Khan, Automatic classification of plants based on their leaves, Biosyst. Eng. 139 (2015) 66–75, https://doi.org/10.1016/j.biosystemseng.2015.08.003.

[2] C.C. Aggarwal, A. Hinneburg, D.A. Keim, On the Surprising Behavior of Distance Metrics in High Dimensional Space, in: J. den Bussche, V. Vianu (Eds.), Database Theory — ICDT 2001, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 420–434.

[3] Angelova, A., Zhu, S., 2013. Efficient Object Detection and Segmentation for Fine-Grained Recognition, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 811–818. doi: 10.1109/CVPR.2013.110..

[4] V. Aráujo, A.S. Britto, A.L. Brun, A.L. Koerich, R. Falate, Multiple classifier system for plant leaf recognition, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2017, https://doi.org/10.1109/SMC.2017.8122891.

[5] Araújo, V.M., Jr., A.S.B., Brun, A.L., Koerich, A.L., Oliveira, L.E.S., 2018. Fine-grained hierarchical classification of plant leaf images using fusion of deep models, in: IEEE 30th International Conference on Tools with Artificial Intelligence, ICTAI 2018, 5-7 November 2018, Volos, Greece, pp. 1–5. doi: 10.1109/ICTAI.2018.00011..

[6] J.G.A. Barbedo, Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification, Computers Electronics Agric. 153 (2018) 46–53, https://doi.org/10.1016/j.compag.2018.08.013.

[7] P. Barré, B.C. Stöver, K.F. Müller, V. Steinhage, LeafNet: A computer vision system for automatic plant species identification, Ecological Informatics 40 (2017) 50–56, https://doi.org/10.1016/j.ecoinf.2017.05.005.

[8] Bloice, M.D., Stocker, C., Holzinger, A., 2017. Augmentor: An image augmentation library for machine learning. arXiv:1708.04680..

[9] V. Bodhwani, D.P. Acharjya, U. Bodhwani, Deep residual networks for plant identification, Procedia Computer Science 152 (2019) 186–194, https://doi.org/10.1016/j.procs.2019.05.042.

[10] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature Verification Using a "Siamese" Time Delay Neural Network, in: Proceedings of the 6th International Conference on Neural Information Processing Systems, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993, pp. 737–744.

[11] J. Chaki, R. Parekh, S. Bhattacharya, Plant leaf recognition using texture and shape features with neural classifiers, Pattern Recogn. Lett. 58 (2015) 61–68, https://doi.org/10.1016/j.patrec.2015.02.010.

[12] E. Elhariri, N. El-Bendary, A.E. Hassanien, Plant classification system based on leaf features, in: Proceedings of 2014 9th IEEE International Conference on Computer Engineering and Systems ICCES 2014, 271–276, 2014, https://doi.org/10.1109/ICCES.2014.7030971.

[13] G. Figueroa-Mata, E. Mata-Montero, Using a convolutional siamese network for image-based plant species identification with small datasets, Biomimetics 5 (2020), https://doi.org/10.3390/biomimetics5010008.

[14] Z. Ge, A. Bewley, C. McCool, P. Corke, B. Upcroft, C. Sanderson, Fine-grained classification via mixture of deep convolutional neural networks, in: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2016, pp. 1–6, https://doi.org/10.1109/WACV.2016.7477700, arXiv:1511.09209.

[15] Z. Ge, C. McCool, C. Sanderson, A. Bewley, Z. Chen, P. Corke, Fine-grained bird species recognition via hierarchical subset learning, in: 2015 IEEE International Conference on Image Processing (ICIP), 2015, pp. 561–565, https://doi.org/10.1109/ICIP.2015.7350861.

[16] M.M. Ghazi, B. Yanikoglu, E. Aptoula, Plant identification using deep neural networks via optimization of transfer learning parameters, Neurocomputing 235 (2017) 228–235, https://doi.org/10.1016/j.neucom.2017.01.018.

[17] G.L. Grinblat, L.C. Uzal, M.G. Larese, P.M. Granitto, Deep learning for plant identification using vein morphological patterns, Computers Electronics Agric. 127 (2016) 418–424, https://doi.org/10.1016/j.compag.2016.07.003.

[18] D. Han, Q. Liu, W. Fan, A new image classification method using cnn transfer learning and web data augmentation, Expert Syst. Appl. 95 (2018) 43–56, https://doi.org/10.1016/j.eswa.2017.11.028.

[19] Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.P., Planqué, R., Rauber, A., Palazzo, S., Fisher, B., Müller, H., 2015. LifeCLEF 2015: Multimedia Life Species Identification Challenges, in: Proceedings of the 6th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction - Volume 9283, pp. 462–483. doi: 10.1007/978-3-319-24027-5_46..

[20] A. Kadir, L.E. Nugroho, A. Susanto, P.I. Santosa, Performance Improvement of Leaf Identification System Using Principal Component Analysis, Int. J. Adv. Sci. Technol. 44 (2012) 113–124.

[21] Khosla, A., Jayadevaprakash, N., Yao, B., fei Li, F., L.: Novel dataset for fine-grained image categorization, in: First Workshop on Fine-Grained Visual Categorization, CVPR (2011)..

[22] Koch, G.R., 2015. Siamese neural networks for one-shot image recognition..

[23] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates Inc, pp. 1097–1105. URL:http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf..

[24] Kumar, N., Belhumeur, P.N., Biswas, A., Jacobs, D.W., Kress, W.J., Lopez, I.C., Soares, J.V.B., 2012. Leafsnap: A Computer Vision System for Automatic Plant Species Identification, in: Computer Vision – ECCV 2012, pp. 502–516..

[25] S.H. Lee, C.S. Chan, S.J. Mayo, P. Remagnino, How deep learning extracts and learns leaf features for plant classification, Pattern Recogn. 71 (2017) 1–13, https://doi.org/10.1016/j.patcog.2017.05.015.

[26] S.H. Lee, C.S. Chan, P. Wilkin, P. Remagnino, Deep-plant: Plant identification with convolutional neural networks, in: Proceedings - International Conference on Image Processing ICIP 2015-December, 452–456, 2015, https://doi.org/10.1109/ICIP.2015.7350839, arXiv:1506.08425.

[27] S.H. Lee, Y.L. Chang, C.S. Chan, P. Remagnino, Hgo-cnn: Hybrid generic-organ convolutional neural network for multi-organ plant classification, in: 2017 IEEE Int'l Conference on Image Processing (ICIP), 2017, pp. 4462–4466, https://doi.org/10.1109/ICIP.2017.8297126.

[28] I. Melekhov, J. Kannala, E. Rahtu, Siamese network features for image matching, in: 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 378–383, https://doi.org/10.1109/ICPR.2016.7899663.

[29] Y.G. Naresh, H.S. Nagendraswamy, Classification of medicinal plants: An approach using modified LBP with symbolic representation, Neurocomputing 173 (2016) 1789–1797, https://doi.org/10.1016/j.neucom.2015.08.090.

[30] P. Pawara, E. Okafor, L. Schomaker, M. Wiering, Data Augmentation for Plant Classification, in: Advanced Concepts for Intelligent Vision Systems, Springer International Publishing, Cham, 2017, pp. 615–626.

[31] C.A. Priya, T. Balasaravanan, A.S. Thanamani, An efficient leaf recognition algorithm for plant classification using support vector machine, in: 2012 International Conference on Pattern Recognition, Informatics and Medical Engineering, PRIME, 2012, pp. 428–432, https://doi.org/10.1109/ICPRIME.2012.6208384.

[32] R.T. Schuh, A.V.Z. Brower, Biological Systematics: Principles and Applications, 1 ed.,., Cornell University Press, 2009.

[33] A.R. Sfar, N. Boujemaa, D. Geman, Confidence Sets for Fine-Grained Categorization and Plant Species Identification, Int. J. Comput. Vision 111 (2015) 255–275, https://doi.org/10.1007/s11263-014-0743-3.

[34] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556..

[35] M. Simpson, Plant Systematics, Elsevier Science, 2010.

[36] Snell, J., Swersky, K., Zemel, R.S., 2017. Prototypical networks for few-shot learning. CoRR abs/1703.05175. arXiv:1703.05175..

[37] Söderkvist, O., 2001. Computer vision classification of leaves from swedish trees..

[38] M. Šulc, J. Matas, Fine-grained recognition of plants from images, Plant Methods 13 (2017) 1–14, https://doi.org/10.1186/s13007-017-0265-4.

[39] C. Sungbin, Plant identification with deep convolutional neural network snumedinfo at lifeclef plant identification task 2015, CLEF (Working Notes), 2015.

[40] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9, https://doi.org/10.1109/CVPR.2015.7298594, URL:https://ieeexplore.ieee.org/document/7298594.

[41] Taylor, L., Nitschke, G., 2017. Improving deep learning using generic data augmentation. CoRR abs/1708.06020. arXiv:1708.06020..

[42] J. Wäldchen, M. Rzanny, M. Seeland, P. Mäder, Automated plant species identification-Trends and future directions, PLoS Comput. Biol. 14 (2018) 1–19, https://doi.org/10.1371/journal.pcbi.1005993.

[43] B. Wang, D. Wang, Plant Leaves Classification: A Few-Shot Learning Method Based on Siamese Network, IEEE Access 7 (2019) 151754–151763, https://doi.org/10.1109/ACCESS.2019.2947510.

[44] X.S. Wei, P. Wang, L. Liu, C. Shen, J. Wu, Piecewise Classifier Mappings: Learning Fine-Grained Learners for Novel Categories With Few Examples, IEEE Trans. Image Process. 28 (2019) 6116–6125, https://doi.org/10.1109/tip.2019.2924811, arXiv:1805.04288.

[45] Wu, S.G., Bao, F.S., Xu, E.Y., Wang, Y.X., Chang, Y.F., Xiang, Q.L., 2007. A leaf recognition algorithm for plant classification using probabilistic neural network. ISSPIT 2007–2007 IEEE International Symposium on Signal Processing and Information Technology, 11–16. doi: 10.1109/ISSPIT.2007.4458016, arXiv:0707.4289..

[46] B. Yanikoglu, E. Aptoula, C. Tirkaz, Automatic plant identification from photographs, Mach. Vis. Appl. 25 (2014) 1369–1383, https://doi.org/10.1007/s00138-014-0612-7.

[47] G. Zhi-Yong, H.X. Xie, J.F. Li, S.L. Liu, Spatial-structure siamese network for plant identification, Int. J. Pattern Recognit Artif Intell. 32 (2018) 1850035, https://doi.org/10.1142/S0218001418500350, arXiv:https://doi.org/10.1142/S0218001418500350.

[48] Y. Zhu, W. Sun, X. Cao, C. Wang, D. Wu, TA-CNN: Two-way attention models in deep convolutional neural network for plant recognition, Neurocomputing (2019), https://doi.org/10.1016/j.neucom.2019.07.016.

Voncarlos M. Araújo received M.Sc. degree in Applied Computing from the State University of Ponta Grossa (UEPG) in 2016, and Ph.D. degree in Computer Science from the Pontifícia Universidade Católica do Paraná (PUCPR, Brazil) in 2021 Currently, he is a Post-doctoral researcher at the Université du Québec en Outaouais (UQO) in partnership with UQAM, Montreal, Canada. His research interests include Computer Vision, Machine Learning and Privacy Attacks.

Alceu S. Britto Jr received M.Sc. degree in Industrial Informatics from the Centro Federal de Educacão Tecnológica do Paraná (CEFET-PR, Brazil) in 1996, and Ph.D. degree in Computer Science from the Pontifícia Universidade Católica do Paraná (PUCPR, Brazil) in 2001. In 2013, he worked as a postdoctoral researcher with CNPq financial support at LIVIA Lab, École de Technologie Supérieure (ÉTS), Montreal, Canada. In 1989, he joined the Informatics Department of the Universidade Estadual de Ponta Grossa (UEPG, Brazil). In 1995, he also joined the Computer Science Department of the Pontifícia Universidade Católica do Paraná (PUCPR) and, in 2001, the Post-graduate Program in Informatics (PPGIa). From 1998 to 2002, he was a visiting scientist at the CENPARMI, Montréal, Canada. His current interests include Pattern Recognition, Machine Learning, Image Analysis, and Evolutionary Computation.

Luiz S. Oliveira received the B.S. degree in computer science from Unicenp, Curitiba, PR, Brazil, the M.Sc. degree in electrical engineering and industrial informatics from the Centro Federal de Educacão Tecnológica do Paraná (CEFET-PR), Curitiba, and the Ph.D. degree in computer science from the École de Technologie Supérieure, Université du Québec, in 1995, 1998, and 2003, respectively. From 2004 to 2009, he was a Professor with the Computer Science Department, Pontifical Catholic University of Paraná, Curitiba. In 2009, he joined the Federal University of Paraná, Curitiba, where he is a Professor with the Department of Informatics and the Head of the Graduate Program in computer science. His current interests include Pattern Recognition, Machine Learning, Image Analysis, and Evolutionary Computation

Alessandro L. Koerich is an Associate Professor in the Dept. of Software and IT Engineering of the École de Technologie Supérieure (ÉTS), University of Québec, Montréal, Canada. He received the B.Sc. degree in electrical engineering from the Federal University of Santa Catarina, Brazil, in 1995, the M.Sc. degree in electrical engineering from the University of Campinas, Brazil, in 1997, and the Ph.D. degree in engineering from the ÉTS, in 2002. From 1997 to 1998, he was a lecturer at the Federal Technological University of Paraná. From 1998 to 2002, he was a visiting scientist at the CENPARMI, Montréal, Canada. From 2003 to 2015 he was with the Pontifical Catholic University of Parana, Curitiba, Brazil, where he became professor in 2010 and served as chair of Graduate Studies in Computer Science from 2006 to 2008. From 2009 to 2015 he was also an associate professor in the Dept. of Electrical Engineering of Federal University of Paraná. In 2004, he was nominated IEEE CS Latin America Distinguished Speaker. He was a visiting researcher at INESC-Porto, Portugal from 2009 to 2012 and served as a Fulbright Visiting Professor in the Dept. of Electrical Engineering at Columbia University, New York, USA, in 2013. Prof. Lameiras Koerich is the author of more than 100 papers and holds four patents. He is an associate editor of the Pattern Recognition journal and served as the general chair of the 14th Intl Society for Music Information Retrieval Conference, which was held in Curitiba, Brazil in 2013. His current research interests include computer vision, machine learning and music information retrieval.