Multi-Organ Plant Classification based on Convolutional and Recurrent Neural networks

Sue Han Lee, *Student Member, IEEE,* Chee Seng Chan, *Senior Member, IEEE,* and Paolo Remagnino, *Senior Member, IEEE*

***Abstract*—Classification of plants based on a multi-organ approach is very challenging. Although additional data provides more information that might help to disambiguate between species, the variability in shape and appearance in plant organs also raises the degree of complexity of the problem. Despite promising solutions built using deep learning enable representative features to be learned for plant images, the existing approaches focus mainly on generic features for species classification, disregarding the features representing plant organs. In fact, plants are complex living organisms sustained by a number of organ systems. In our approach, we introduce a hybrid generic-organ convolutional neural network (HGO-CNN), which takes into account both organ and generic information, combining them using a new feature fusion scheme for species classification. Next, instead of using a CNN based method to operate on one image with a single organ, we extend our approach. We propose a new framework for plant structural learning using the recurrent neural network (RNN) based method. This novel approach supports classification based on a varying number of plant views, capturing one or more organs of a plant, by optimizing the contextual dependencies between them. We also present the qualitative results of our proposed models, based on feature visualisation techniques and show that the outcomes of visualisations depict our hypothesis and expectation. Finally, we show that by leveraging and combining the aforementioned techniques, our best network outperforms the state-of-the-art on the PlantClef2015 benchmark.**

***Index Terms*—Plant classification, deep learning.**

I. Introduction

BIODIVERSITY is declining steadily throughout the world, mainly due to direct or indirect human activities. To protect biodiversity, people have begun building knowledge of accurate species to recognize unknown plant species. Taxonomists, botanists, and other professionals determine plant species from field observation based on a substantial species knowledge gained through their field work and studies. Categorisation of plants still remain a tedious task due to limited knowledge and information of world's plant families. For this reason, taxonomists started seeking methods that can meet species identification requirements, such as developing digital image processing and pattern recognition techniques [1].

Recent progress in computer vision makes it possible to assist botanists in plant identification tasks. The majority of computer vision approaches utilizes leaves for discrimination, as leaf characters have been predominantly used to clarify plants. Characters such as shape, texture and venation are the features most generally used to distinguish leaves of different species [2]. Nevertheless, due to the intra or interspecies diversity of plants in nature, some species are difficult or impossible to differentiate from one another using only the leaf organ. In fact, this ambiguity occurs also in other organs. For example as shown at the top of Fig. 2, the images of fruits are visually similar. Using solely a single image of a fruit organ makes it considerably hard to differentiate between species, especially for non-botanists who have limited knowledge of plant characters. However, if we extend our observation to multiple organs such as branches and leaves (as shown at the bottom of Fig. 2), together with fruits, we can easily find out that they have discriminative patterns, as a significant cue for plant recognition. For example, the differences between the appearance of branches as well as the venations of leaves. So, in this case, it is obvious that observing different organs can help to ease the plant identification task. On the other hand, there are times when certain organs are not in season, for example, during winter we can only observe the bark of a deciduous plant. Under these circumstances, it is known to be more informative to capture multiple viewpoints of the bark to increase the species discrimination [3].

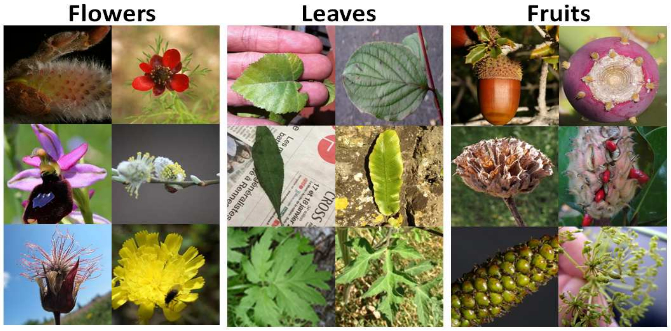


Fig. 1: Example of plant organs images, and we can observe the large variability in their appearance.

Therefore, in connection with the aforementioned studies, researchers started to focus on the automatic analysis of multiple images exploiting different views of a plant capturing one or more organs. However, it is a challenging task to classify different organs plant images. For example, in Fig 1, we can observe the large variability in the appearance of plant organs. Even within the same organ, large differences can occur. Furthermore, images of plants taken in the field

S.H Lee and C.S Chan are with the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia e-mail: (leesuehan@siswa.um.edu.my (S.H Lee), cs.chan@um.edu.my (C.S Chan).

Paolo Remagnino is with the Faculty of Science, Engineering and Computing, Kingston University, KT1 2EE, United Kingdom, email:(p.remagnino@kingston.ac.uk)

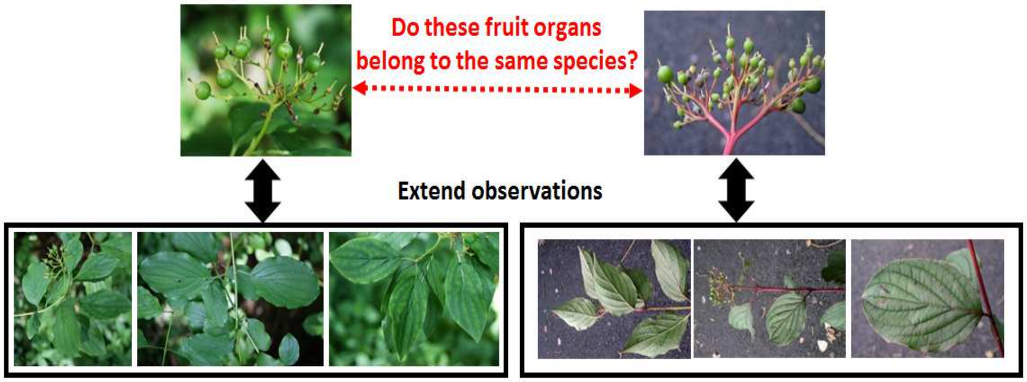


Fig. 2: Examples of very similar appearance of fruit organs between species (right: Cornus mas L., left: Cornus sanguinea L.). However, by extending our observation to different views capturing one or more organs such as branches and leaves, we can easily find out the discriminative patterns. For examples, color and texture of the branches as well as the venation structure of the leaves.

with clutter in the background are more difficult to recognize. For this, researchers generally adopt organ-specific features for discrimination [4]–[9]. They first group images of plants into their respective organ categories. Then, based on each organ category, organ-specific features are extracted using feature engineering approaches such as Scale-invariant feature transform (SIFT), Bag of Word (Bow), Speeded-Up Robust Features (SURF), Gabor, Local Binary Pattern (LBP). During the species classification stage, the computed features for each organ category are trained individually using conventional machine learning algorithms such as Support Vector Machine (SVM), k-means clustering, Weighted Probability (WP) approach, nearest neighbour classifier and random forest. Although successful, to design or decide which feature descriptors to use for each organ is highly dependent on the prior knowledge of plant organs, and, this information is usually only partially available or incomplete for non-specialist users.

Deep Learning (DL) [10] is an emerging technology that has proved extremely high recognition capabilities with very large datasets, replacing the need of designing hand-crafted features as to previous approaches [4]–[9]. The Convolutional Neural Network (CNN), as one of the most used DL methods has been employed to learn generic representation for images of plants [11]–[14]. Specifically, M-class species classifier is trained, irrespective of the organ or organ structure. Although generic features can model target species classes, they might not be able to provide an appropriate description for a plant. For example, for a leaf image taken with a noisy background, as the leaf on the newspaper shown in Fig. 1, generic features focus on the holistic representation of the image. In such case, text might be considered erroneously as one of the discriminative features for the species. This is not surprising, as a generic network learns irrelevant features, especially when they appear to be discriminative among species. For this reason, we propose a new CNN architecture that can go beyond the regular generic description of a plant, integrating the organ-specific features together with the generic features to explicitly force the designed network to focus on the organ regions during species classification.

Although existing CNN methods can model a suitable feature representation for a plant image, they lack the capability to model the global relationship between different plant views (or organs) captured of a plant. The reason is that existing CNN based approaches were designed to operate on a single plant image, focusing on capturing the similar region-wise patterns within an image but not the structural patterns of a plant seen from multiple views with one or more of its organs. This is particular important as these images captured from a same plant contain structural information that is not mutually exclusive. In fact, they share overlapping characteristics which are useful for species recognition. Henceforth, this motivates us to move beyond existing practice, proposing a new plant classification framework that takes into consideration contextual dependencies between varying plant views capturing one or more organs of a plant.

In this work, we present two frameworks to classify different plant organs images. First, we present a novel CNN architecture called the hybrid generic-organ convolutional neural network, abbreviated HGO-CNN. Specifically, it extracts prior organ information, and, classifies one image based on the correlation between the chosen organ and generic-based features. Second, we propose a new framework of plant structural learning based on recurrent neural networks (RNN), namely the Plant-StructNet. Specifically, it takes in a varying number of plant views images composed of one or more organs, and, optimizes the contextual dependencies between them for species classification. To summarize our major contributions:

1) We present two novel plant classification frameworks, namely the HGO-CNN (Sec. III) and Plant-StructNet (Sec. IV). The HGO-CNN can be seen as a per-image modeling focusing on feature representation of one image capturing a single plant view (or organ), while the Plant-StructNet can be as a multi-image modeling that operates on multiple plant views capturing one or more organs of a plant.
2) We experimentally show that modeling the dependencies between plant views can essentially improve the performance of plant classification (Sec. VII-A). In addition,

we demonstrate that the ensemble model combining the enhanced HGO-CNN and Plant-StructNet architectures outperforms the state-of-the-art (SOTA) on the Plant-Clef2015 [15] dataset (Sec. VII-C).

3) Besides quantitatively analyzing our proposed models, we go deeper into exploring, analyzing and understanding the learned features through feature visualisation techniques. Through the deconvolution approach [16], we show that both the organ and generic features learned in HGO-CNN exhibit different contextual information of a plant image (Sec.VI-C). Furthermore, through the t-SNE [17], we can observe the discriminative behavior of both HGO-CNN and Plant-StructNet that reflects the quantitative results (Sec. VII-A).

A preliminary version of the HGO-CNN was presented earlier [18]. The present work adds to the initial version in significant ways. Firstly, we present and analyze in this paper various improvements we have made to our previous HGO-CNN, and, find that enhancing the feature fusion can further improve its classification performance. Secondly, we propose a Plant-StructNet that supports classification based on varying plant views, and to our surprise, it is able to improve the prediction of the less informative plant organ that is hardly handled by the HGO-CNN. Next, we experimentally prove that the ensemble model combining both the proposed HGO-CNN and Plant-StructNet, outperforms the SOTA result.

Our paper begins with a comprehensive review of existing methods of plant identification in Sec. II. Inspired by the success of RNN in modeling long-term dependency, we also review RNN and its varying application. Sec. III introduces the idea of HGO-CNN for end-to-end automatically processing and classification for the multi-organ plant data. Next, we introduce the Plant-StructNet architecture that built upon the concept of RNN to distinguish plant species at Sec IV. The experiments of HGO-CNN and Plant-StructNet are given in Sec. VI and Sec. VII respectively. Finally, we conclude this paper in Sec. VIII.

II. Related Work

Plant identification. Over the past few years, researchers have worked on recognizing plant species using solely a single plant organ. A majority of the studies have utilized leaves to identify species. Leaf characters such as shape [19]–[22], texture [21], [23], and venation [24]–[26] are the most generally used features to distinguish leaves of different species. Lately, [2] proposed the use of deep learning for reverse engineering features of leaf, and, found out that different orders of leaf venation are more discriminatory than leaf shape features. Other than leaf, there are also researchers focus on using flower [27]–[29] to identify species.

To fit better with a real scenario where a botanist generally tries to identify a plant by observing several plant organs or a similar organ from different viewpoints during times when other organs are not in season, researchers in computer vision have focused on designing an automated plant classification system to identify different organs plant images. Earliest attempts [4]–[9], [30] in general, adopt feature extraction and classification as two separate steps, and, they engineered the features. For example, to support large-scale plant species identification, a course-to-fine method was introduced through constructing a hierarchical classifier [30]. Although reliable performance was reported, cascaded inferences in the hierarchical classifier are very much affected by the selection of the best subset of handcrafted features, which are in turn, task or dataset dependent. Lately, [31] proposed using an end-to-end CNN to replace those hand-crafted feature extractors. They introduced organ-specific CNN models where each model is trained on dedicated plant organs. Although CNN is powerful in learning discriminative features, constraining it to learn on specific organ categories might restrict its performance.

Other research [11]–[14] has focused on using CNN to learn generic features of plants, irrespective of their organ information. In this case, multi-organ plant images are trained together using a generic CNN model. In the LifeClef2015 challenge, [12] showed that using the deepest network of GoogLenet, could provide the best result. However, generic features tend to focus on the holistic structures of an image, neglecting relevant attributes describing characteristics of a plant organ. Our work aims to solve this problem by designing a new CNN model that can extract prior organ information, and, subsequently combine it with generic features for species recognition.

RNN based classification The RNN has received great attention due to its capability of processing sequential data such as language translation [32]–[34] and action recognition [35]–[37]. Recently, CNN and RNN have been employed to combine information, integrating the domain of computer vision and natural language processing. For example, image [38], [39] or video [40], [41] to text translation and reasoning as well as question and answering based on images [33], [42]–[44]. Due to the inherent sequential nature of video and language, Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) are the generally used architectures to process these data.

Other than the capability of modeling video or language data, lately, a few publications have showed the effectiveness of RNN based approaches to process variable length of fixed-sized data in a sequential manner though data originally is not in a form of sequences. Specifically, RNN is used to model the dependencies between pixels or regions within an image. For example, it has been actively explored in segmentation [45]–[47], scene labeling [48]–[50], object recognition [51], [52] and detection [53], [54], as well as image generation [55]. Our work builds on the foundations laid in these approaches. Nevertheless, instead of using RNN based method to process pixels [46], [47] or regions [51], [55] level information of an image, we formulate it to process the structural level information of a plant based on several images captured from its various organs or different viewpoints of a similar organ. In particular, this can be seen as a first step towards plant semantic learning systems that modeling plant species based on multiple plant views capturing one or more organs of plant.

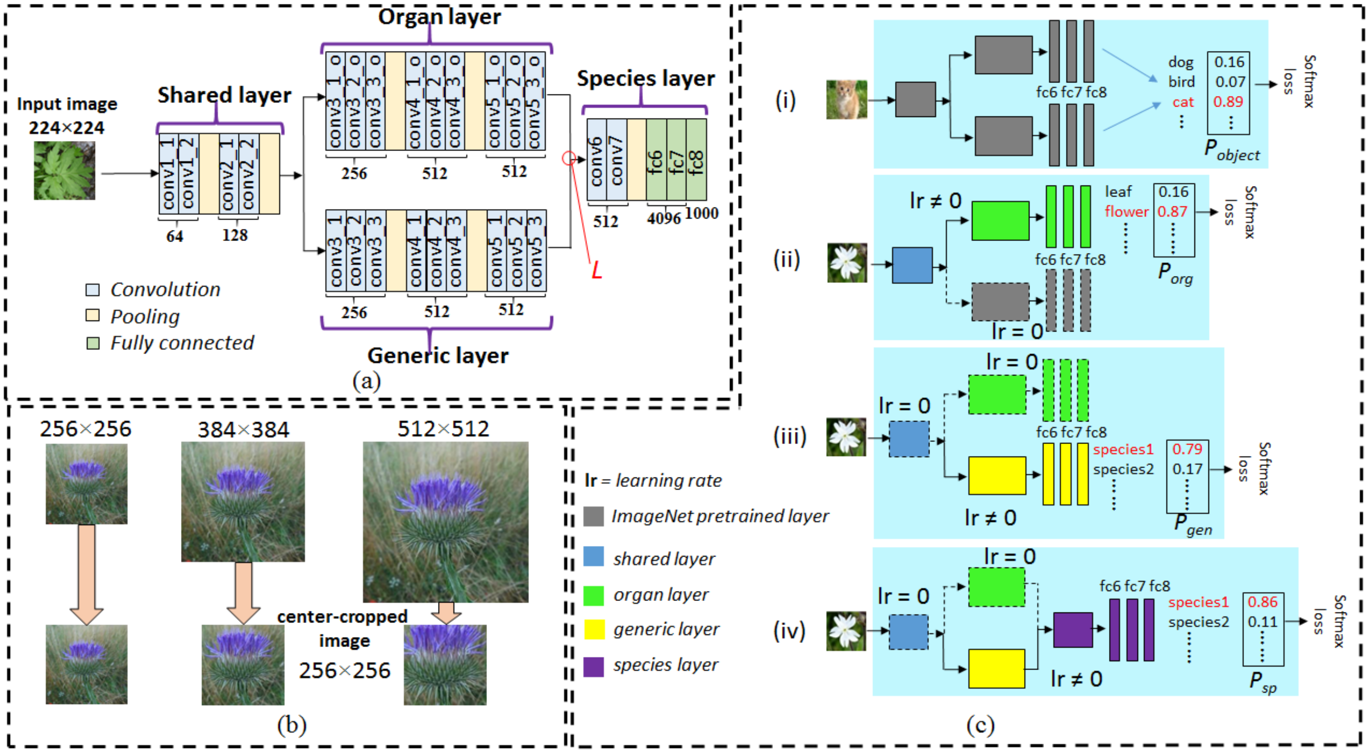


Fig. 3: Overview of HGO-CNN framework. (a) The architecture of HGO-CNN; (b) Multi-scale plant images generation: given a plant image, we isotropically rescale the training images into three different sizes: 256, 384 and 512. Then, for 384 and 512 image sizes, we crop 256 × 256 center pixels; (c) The HGO-CNN feature fusion scheme: (i) during training, the two-path CNN is initially pretrained with the ImageNet dataset [56]. (ii) Then, one of the CNN path is repurposed for the organ task, while (iii) the another CNN path is repurposed for the generic task. (IV) Finally, new species layers are introduced to train the correlation between both the organ and generic components. Best viewed in color.

III. The HGO-CNN

Generally, botanists can classify plants by observing and studying their features, usually using all the plant organs. Plant organs are known prior to explore the characteristic of a species. For instant, when botanists study a leaf, they focus on the leaf characters such as its *margin* or *venation* patterns, and, when they study a flower, they focus on the characteristics of its *petals*, *sepals* and *stamen* to identify unknown plant species. So, it is logical to believe that a better recognition method for plant species might require prior information of their organs. We propose an end-to-end network, namely the HGO-CNN, to classify different plant organ images. HGO-CNN is able to encapsulate organ and generic information prior to species classification. Fig. 3(a) depicts its architecture.

A. Architecture

The proposed HGO-CNN comprises four layers or components: (i) a shared layer, (ii) an organ layer, (iii) a generic layer, and (iv) a species layer. The rationale behind proposing a shared layer is inspired by: (1) the work of [16], [57], who demonstrated that bottom layers in deep networks respond to low-level features, such as corners and edges, in turn crucial to the classification of any high level features, and, (2) the fact that such layers help reducing the number of training parameters.

Input to our HGO-CNN is a 224 × 224 color image. For the convolutional layer, we utilise 3 × 3 convolution filters with spatial resolution preserved using stride 1. Max pooling is performed using a 2 × 2 pixel window with stride 2. Three fully connected layers, which have 4096, 4096 and 1000 channels respectively, follow behind the stacks of convolutional layers. The output of the last hidden layer is normalized with the softmax function:

$$P(r|\mathbf{I}) = \frac{e^{s_r(\mathbf{I})}}{\sum_{m=1}^{M} e^{s_m(\mathbf{I})}} \tag{1}$$

where M and r stand for the total number of classes and the target class respectively, while $\mathbf{s}(\mathbf{I})$ stands for the final activation of the input plant image I obtained at the last hidden layer. After performing the softmax operation, we find the maximum likelihood of the sample by applying the objective function, $L_{hgo} = -logP(r|\mathbf{I})$.

B. Multi-scale plant images generation

To increase the robustness of a system in recognising multi-organ plant images, we generate multi-scale plant images for training as depicted in Fig. 3(b). We isotropically rescale the training images into three different scales: 256, 384 and 512, where each scale be the smallest side of an isotropically-rescaled training image. Then, we crop 256 × 256 center pixels. By doing this, the crops from the larger scale images will correspond to a small parts of images or particularly subparts of organs, while, the crops from the smaller scale images hold the information of the entire organs. During network training, 224 × 224 pixels are randomly cropped from the rescaled images and fed into the network. During testing, we do apply a similar multi-scale process to obtain three sets of testing images for a query image. An averaging fusion method is then used to combine their softmax scores to output a final result for a query image.

C. Feature Fusion Scheme

In order to train the HGO-CNN to capture prior organ information, and, subsequently integrate both generic and organ-based information for species classification, we propose a feature fusion scheme. It is based on a novel step-by-step training strategy (illustrated in Fig. 3(c)):

i. Pre-Training CNN layers HGO-CNN uses a two-path CNN as shown in Fig. 3(c)(i) for the purpose of training generic and organ based features at a later stage. This two path CNN is similar to the architecture depicted in Fig. 3(a), except that, it does not include the interconnection between paths, and, each path has its own fully connected layers. These are initially pre-trained using the ImageNet challenge dataset [56].

ii. Organ layer After we obtained the pre-trained two-path CNN, one of the CNN paths is repurposed to extract organ features. This organ layer is trained together with the shared layer, using seven kinds of organ labels predefined in the PlantClef2015 dataset. The organ labels are branch, entire, flower, fruit, leaf, stem and leafscan. We obtain organ-based feature maps, $\mathbf{x_{org}} \in \mathbb{R}^{H \times W \times Z}$ where H, W and Z are the height, width and number of channels of the respective feature maps. We train the shared layer based on the organ labels is because the shared layer that corresponds to the low-level features is more appropriate to be trained upon the course-level organ classes instead of the class-specific species classes. So that, it can be more generalised to fit in the modeling of both target classes.

iii. Generic layer After training the organ layer, another CNN path is repurposed to extract the generic features. This generic layer is trained using the 1000 species labels predefined in the PlantClef2015 dataset, regardless of organ information. We obtain generic-based feature maps, $\mathbf{x_{gen}} \in \mathbb{R}^{H \times W \times Z}$. To allow both the organ and generic layers to share the common proceeding layer, we keep the shared layer's weights to be consistent. This is achieved by setting their learning rate to zero.

iv. Species layer To introduce correlation between both the organ and generic components, we introduce a fusion function g_{cat}. It is employed at stage L (after the last convolutional layer for both components as shown in Fig. 3(a)). In our model, g_{cat} first concatenates $\mathbf{x_{gen}}$ and $\mathbf{x_{org}}$ along the channel axis, forming a stacked data, $\mathbf{x_{cat}} = [\mathbf{x_{gen}}, \mathbf{x_{org}}]$, where $\mathbf{x_{cat}} \in \mathbb{R}^{H \times W \times 2Z}$. Then, $\mathbf{x_{cat}}$ will subsequently convolves with a set of filters $\mathbf{f} \in \mathbb{R}^{p \times q \times 2Z \times N}$ with dimension $p \times q \times 2Z$ and biases $\mathbf{b} \in \mathbb{R}^N$: $\mathbf{y_{cat}} = \mathbf{x_{cat}} * \mathbf{f} + \mathbf{b}$. We set $N = Z$ so that we can reduce the dimensionality of the output feature maps $\mathbf{y_{cat}}$, while, at the same time, modeling the correspondence between the two feature maps $\mathbf{x_{gen}}$ and $\mathbf{x_{org}}$. The feature maps $\mathbf{y_{cat}}$ will then go through convolution layers to learn the combined representation of generic and organ features. Since these two convolution layers are new randomly-initialised, we set their learning rate to be 10 times higher than the other layers during training.

IV. The Plant-StructNet

Plants are complex living organisms sustained by a number of organ systems. To recognize plant species, botanists usually observe multiple plant structures captured from a same plant to encounter the local ambiguities of features between species brought by the intra and interspecies diversity of plants in nature. For example as shown in Fig. 2, incorporating observation from multiple plant organs such as branches, leaves and fruits provides a better understanding on the discriminative patterns to distinguish plant species. There are also times people extend their observation to multiple views of a similar organ when other types of organs are not in season [3]. Although the existing CNN methods allow us to extract the discriminative features of a plant image without the needs of handcrafting features, it has been designed to operate on a single plant image which in turns is incapable of modeling the contextual dependencies between varying plant views capturing one or more organs of a plant. We believe that different plant views captured from a same plant contain structural information that is not mutually exclusive, but in fact, they are correlated. For this reason, we move beyond existing practice, proposing the Plant-StructNet to model high level contextual dependencies between plant views comprising varying organs or different viewpoints of a similar organ.

A. Architecture

It is known that human brain processes information iteratively, where it keeps the current state in an internal memory and uses it to infer future observation, capturing the potential relationships between them [58], [59]. Driven by this insight, we build the Plant-StructNet upon the RNN, which it can hold and relate different structural information of a plant. It would be also versatile to deal with arbitrary number of plant images. Plant-StructNet is based on a probabilistic framework that can directly maximize the probability of the correct species label, conditioned on all other related plant images by using the following cross entropy function:

$$L_t = -logP(r_t | \mathbf{I_t}, \{r_d\}_{d \neq t}) \tag{2}$$

where $t = 1, ..., T$ are the states corresponding to the indices of plant images captured from a same plant. Contrary to modeling video or language data where variable number of inputs are conditioned upon their previous states, $P(r_t | \mathbf{I_t}, r_1, ..., r_{t-1})$, in our case, it is logical to condition

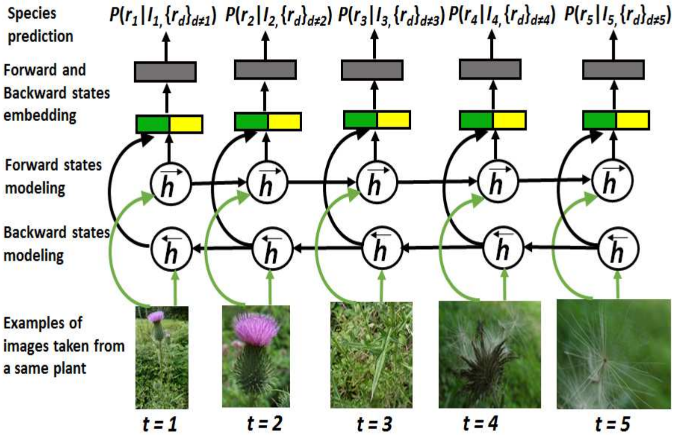

Species prediction
$P(r_1|I_{1,}\{r_d\}_{d\neq 1})$
$P(r_2|I_{2,}\{r_d\}_{d\neq 2})$
$P(r_3|I_{3,}\{r_d\}_{d\neq 3})$
$P(r_4|I_{4,}\{r_d\}_{d\neq 4})$
$P(r_5|I_{5,}\{r_d\}_{d\neq 5})$
Forward and Backward states embedding
Forward states modeling
$\overrightarrow{h}$
Backward states modeling
$\overleftarrow{h}$
Examples of images taken from a same plant
t = 1
t = 2
t = 3
t = 4
t = 5

Fig. 4: The architecture of the proposed Plant-StructNet in classifying different plant views capturing one or more organs of a plant. Each state of the network stores the information of one plant view.

the inputs upon all other states information for the plant structural modeling, $P(r_t|\mathbf{I_t}, \{r_d\}_{d\neq t})$. The reason is that, states in our context are analogous to the collections of plant views captured from a similar plant, so the relationships between these states are interrelated. Henceforth, to tackle this challenge, we design the Plant-StructNet in such a way that it would be able to iteratively classify images of a plant while conjointly operate on all of its related instances. In particular, we build a bidirectional states modeling mechanism where the forward neuron activations $\overrightarrow{\mathbf{h}}$ models $P_{fw} = P(r_t|\mathbf{I_t}, r_1, ..., r_{t-1})$ and the backward neuron activations $\overleftarrow{\mathbf{h}}$ models $P_{bw} = P(r_t|\mathbf{I_t}, r_{t+1}, ..., r_T)$. Then, we put in correspondence between both neurons for every state and train them upon the respective species classes as shown in Fig 4. In this manner, each state t can be considered as condition upon the collections of the related plant images from states $1, ..., t-1, t+1, ..., T$. To model both the P_{fw} and P_{bw}, we adopt GRU [60] as one of the RNN gating mechanisms.

During training, given an array of plant images acquired from a similar plant $\mathbf{I_t} \in \{\mathbf{I_1}, \mathbf{I_2}, ..., \mathbf{I_T}\}$, we first compute their feature representation using CNN, $\boldsymbol{\delta_t} \in \{\boldsymbol{\delta_1}, \boldsymbol{\delta_2}, ..., \boldsymbol{\delta_T}\}$. Then, we feed them sequentially to each state $t = 1, ..., T$ of the Plant-StructNet. The forward activation function of the GRU, $\overrightarrow{\mathbf{h}}_{\mathbf{t}}$ at state t is a linear interpolation between the previous activation $\overrightarrow{\mathbf{h}}_{\mathbf{t-1}}$ and the candidate activation $\overrightarrow{\tilde{\mathbf{h}}}_{\mathbf{t}}$:

$$\overrightarrow{\mathbf{h}}_{\mathbf{t}} = (1 - \overrightarrow{\mathbf{z}}_{\mathbf{t}})\overrightarrow{\mathbf{h}}_{\mathbf{t-1}} + \overrightarrow{\mathbf{z}}_{\mathbf{t}}\overrightarrow{\tilde{\mathbf{h}}}_{\mathbf{t}} \tag{3}$$

where $\overrightarrow{\mathbf{z}}_{\mathbf{t}}$ is the update gate that decides how much of the unit updates its activation. It is computed as follows:

$$\overrightarrow{\mathbf{z}}_{\mathbf{t}} = \sigma(\mathbf{W_{z1}}\boldsymbol{\delta_t} + \mathbf{W_{z2}}\overrightarrow{\mathbf{h}}_{\mathbf{t-1}}) \tag{4}$$

The candidate activation $\overrightarrow{\tilde{\mathbf{h}}}_{\mathbf{t}}$ is computed as follows:

$$\overrightarrow{\tilde{\mathbf{h}}}_{\mathbf{t}} = \tanh(\mathbf{W_{h1}}\boldsymbol{\delta_t} + \mathbf{W_{h2}}(\overrightarrow{\mathbf{v}}_{\mathbf{t}} \odot \overrightarrow{\mathbf{h}}_{\mathbf{t-1}})) \tag{5}$$

where $\overrightarrow{\mathbf{v}}_{\mathbf{t}}$ is the reset gate and $\odot$ is an element-wise multiplication operator, $\overrightarrow{\mathbf{v}}_{\mathbf{t}}$ is computed as:

$$\overrightarrow{\mathbf{v}}_{\mathbf{t}} = \sigma(\mathbf{W_{v1}}\boldsymbol{\delta_t} + \mathbf{W_{v2}}\overrightarrow{\mathbf{h}}_{\mathbf{t-1}}) \tag{6}$$

All the various $\mathbf{W}$ matrices are trained parameters. Specifically, GRU has two gating units to modulate the flow of information inside the unit. The update gate $\overrightarrow{\mathbf{z}}_{\mathbf{t}}$ decides how much of the previous state should be kept around, while the reset gate $\overrightarrow{\mathbf{v}}_{\mathbf{t}}$ determines to which extent the new input should be combined with the previous state. When the reset gate $\overrightarrow{\mathbf{v}}_{\mathbf{t}}$ is off ($\overrightarrow{\mathbf{v}}_{\mathbf{t}}$ close to 0), it allows the unit to forget the previous computed state. To compute the backward activation $\overleftarrow{\mathbf{h}}_{\mathbf{t}}$, we formulate it as for the $\overrightarrow{\mathbf{h}}_{\mathbf{t}}$ but in a reverse direction as shown in Fig 4. In order to correlate between both states, the output activations of the forward and backward GRU are cascaded as follows:

$$\mathbf{h_t} = [\overrightarrow{\mathbf{h}}_{\mathbf{t}}, \overleftarrow{\mathbf{h}}_{\mathbf{t}}] \tag{7}$$

Then, we multiply $\mathbf{h_t}$ with a class embedding matrix, $\mathbf{W_{em}}$, which is $\mathbf{s}(\mathbf{I_t}) = \mathbf{W_{em}}\mathbf{h_t}$ before normalizing it with a softmax function:

$$P(r_t|\mathbf{I_t}, \{r_d\}_{d\neq t}) = \frac{e^{s_r(\mathbf{I_t})}}{\sum_{m=1}^{M} e^{s_m(\mathbf{I_t})}} \tag{8}$$

We perform the softmax operation for every state t preceding the computation of the overall cross entropy function: $L_{psn} = \frac{1}{T}\sum_{t=1}^{T} L_t$, where L_t is mentioned at eqn. (2).

During prediction, the species label for the t-th plant image can be calculated by first simply cascading the output activations of the forward and backward GRU as mentioned in eqn. 7. The output $\mathbf{h_t}$ is then multiplied with the class embedding matrix, $\mathbf{W_{em}}$, before going through the softmax function (eqn. (8)).

V. Datasets and Evaluation metrics

Dataset. The PlantClef2015 dataset was used. It has 1000 plant species classes. Training and testing data comprise 91759 and 21446 images respectively. Each image is associated with a single organ type (branch, entire, flower, fruit, leaf, stem or leaf scan).

Evaluation metrics. Two evaluation metrics are employed: the *image-centered* and the *observation* score [15]. The purpose of the observation score is to evaluate the ability of a model predicting correct species labels for all the users. The observation score calculates the mean of the average classification rate per user as defined:

$$S_{obs} = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{P_u}\sum_{p=1}^{P_u} S_{u,p} \tag{9}$$

where U: represents the number of users, P_u: the number of individual plants observed by the u-th user, $S_{u,p}$: the score between 0 and 1 as the inverse of the rank of the correct species (for the p-th plant observed by the u-th user). Each query observation is composed of multiple images. To compute $S_{u,p}$, we adopt the Borda count (BD) and the majority voting (MAV) based approaches to combine the scores of multiple images:

$$BD = \frac{1}{n}\sum_{k=1}^{n} score_k \tag{10}$$

$$MAV = \max_{1 \leq k \leq n} score_k \tag{11}$$

where n: the total number of images per query observation. $score$: is the softmax output score, which describes the ranking of the species.

Next, the image-centered score evaluates the ability of a system to provide the correct species labels based on a single plant observation. It calculates the average classification rate for each individual plant defined as:

$$S_{img} = \frac{1}{U}\sum_{u=1}^{U} \frac{1}{P_u}\sum_{p=1}^{P_u} \frac{1}{N_{u,p}}\sum_{n=1}^{N_{u,p}} S_{u,p,n} \tag{12}$$

where U and P_u are explained earlier in the text. $N_{u,p}$ is the number of pictures taken from the p-th plant observed by the u-th user, $S_{u,p,n}$ is the score between 0 and 1 equal to the inverse of the rank of the correct species (for the n-th picture taken from the p-th plant observed by the u-th user). We compute the rank of the correct species based on its softmax scores.

VI. Experiments with the HGO-CNN

We train our HGO-CNN model using the *Caffe* library [61]. The networks are trained with back-propagation, using stochastic gradient descent [62]. For the training parameter setting, we employed the fixed learning policy. We set the learning rate to 0.01, and then decrease it by a factor of 10 until the validation set accuracy stops improving. The momentum is set to 0.9 and the weight decay to 0.0001. In all experiments, we use a mini-batch size of 60. We improve the generalization of the model by randomly cropping and mirroring the input image during training. We run the experiments using an NVIDIA K40 graphics card.

A. Performance Evaluation

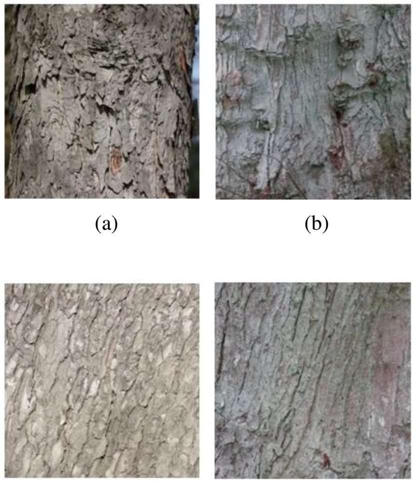
We compare our HGO-CNN with the best plant identification systems evaluated in the previous LifeClef2015 challenge [12], [13], [31]. We also compare with the VGG-16 net [63], which is fine tuned and trained purely on species labels using the PlantClef2015 dataset. This is to measure the contribution of correlation between organ and generic components in the plant species classification. Table I shows the comparison results. We observe that the HGO-CNN model achieves a higher score compared to the VGG-16 net. This confirms the importance of organ features used to discriminate between plant species compared to using solely generic information for plant classification. Apart from that, by applying the multi-scaling technique mentioned in Sec. III-B, the multi-scale HGO-CNN, abbreviated M-S HGO-CNN, outperforms all the previous methods.

TABLE I: Performance comparison with other best plant identification systems evaluated in the LifeClef2015 challenge. Note that, M-S = Multi-scale.

Method	S_{obs}	S_{img}
GoogLeNet + Fisher Vectors (BD) [13]	0.592	-
GoogLeNet (MAV) [13]	0.609	0.581
GoogLeNet (content+ domain) [31]	0.633	-
GoogLeNet + softmax normalization [31]	0.624	0.590
5-fold GoogLeNet (MAV) [12]	0.667	0.652
5-fold GoogLeNet (BD) [12]	0.663	0.652
VGG-16 net(MAV)	0.663	0.638
VGG-16 net(BD)	0.664	0.638
HGO-CNN(MAV)	0.671	0.647
HGO-CNN(BD)	0.673	0.647
M-S HGO-CNN(MAV)	**0.715**	**0.690**
M-S HGO-CNN(BD)	**0.717**	**0.690**

TABLE II: Classification performance comparison of each content based on S_{img}.

Method	Branch	Entire	Flower	Fruit	Leaf	LeafScan	Stem
Choi [12]	0.498	0.531	0.784	0.602	0.600	0.766	0.326
Ge et al. [31]	0.416	0.448	0.738	0.558	0.524	0.694	0.291
Champ et al. [13]	0.398	0.453	0.723	0.559	0.501	0.713	0.302
Le et al. [4]	0.051	0.084	0.207	0.125	0.342	0.737	0.164
VGG-16 net	0.491	0.522	0.777	0.585	0.591	0.747	0.337
HGO-CNN	0.522	0.532	0.779	0.604	0.607	0.690	0.326
M-S HGO-CNN	**0.568**	**0.603**	**0.798**	**0.653**	**0.652**	**0.803**	**0.411**



(a) (b)

(c) (d)

Fig. 5: Species of stem images:(a)Acer pseudoplatanus L.(b) and (d)Acer saccharinum L.(c) Aesculus hippocastanum L.

B. Detailed Scores for Each Plant Organ

In this section, we analyse the classification performance for each organ based on the image-centered score, S_{img}. Instead of calculating the average classification rate for each individual plant based on all the n-th picture taken from the p-th plant, the S_{img} considers only pictures illustrating a dedicated plant view (organ). Table II illustrates the comparison results. We observe that both of our proposed models, HGO-CNN and M-S HGO-CNN show that scanned leaf and flower are the most effective organs compared to others for plant identification. This is similar to the results reported in [15]. Our HGO-CNN shows a higher identification score for the 'Flower' category compared

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2836321, IEEE Transactions on Image Processing

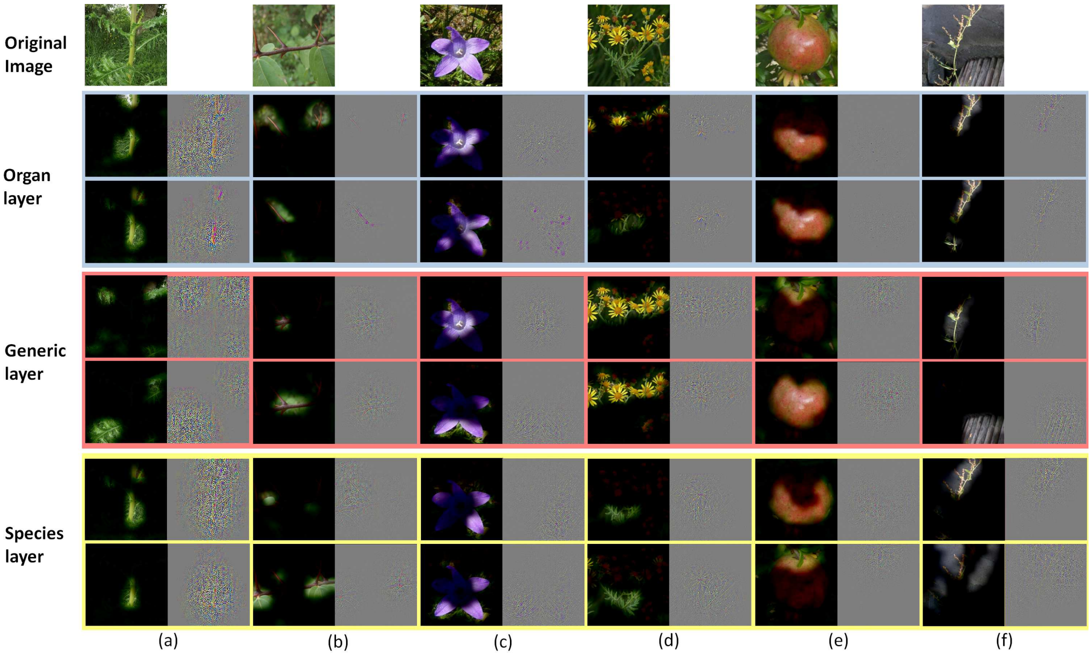


Fig. 6: Visualization of the last convolution of generic, organ and species layer for the test images. Color contrast is digitally enhanced. Figure is best viewed in electronic form.

to 'LeafScan'. In addition, using multi-scale training, M-S HGO-CNN shows a major improvement in the 'LeafScan' category. This indicates that multi-scale training data could further improve the feature representation for multi-organ plant images. Overall, our M-S HGO-CNN achieves the highest S_{img} compared to other SOTAs. Although M-S HGO-CNN leads to a better result for 'Stem', the organ is still considered as the least informative one compared to other organs. This might be due to the intra and interspecies diversity of plants in nature, resulting in a stem not vivid enough for species identification. For example, Fig. 5 illustrates the confusion risk for identifying plant species when only stem information is used. These stem organs are considered hard to disambiguate even by the botanists.

C. Qualitative Analysis

Besides the quantitative analysis, we go deeper into exploring, analyzing and understanding the learned features. We use the deconvolution approach [16] to find which features have been learned in each layer and observe their differences. We subsample the top 2 activation feature maps in the last convolution of each layer and reconstruct them back to image pixels. Fig. 6 shows the learned activation maps as well as the deconvolved images. For example, in the Fig. 6(a), we observe that both organ and generic-based features show complementary information, in which the organ layer mainly focuses on the tree branch, while generic layer stimulates at the twig. The species layer encapsulates both information and reveals the portions that best represent the plant image. Through this visualisation, it is clear that features learned in both layers are not overlapped, but extracting complementary information that could drive the network to better characterize a plant species. Apart from that, we observe that the generic layer erroneously considers non-plant object as one of the discriminative features to distinguish species. For example, in the Fig. 6(f), we observe that the generic features focus at the irrelevant features of the stairs instead of the plant structures. This indicates that although generic features can identify plant species, it might learn irrelevant features that are inappropriate. For this reason, in our work, we propose adding an additional organ features to explicitly force the network to focus on the organ regions in order to boost the species discrimination.

D. Model Improvement

In the previous experiments, we have proven the importance of organ information in plant species predictive modeling. We also have documented the significance of multi-scale training in classifying different organs plant images. Then, we extended and improved the generalization of the model using various enhancement techniques. In this section, we analyze how these techniques improve the previous model. In Table III, we summarize their performance and compare them to our best model (i.e. M-S HGO-CNN(BD)), obtained in the previous set of experiments. In these experiments, besides S_{img} and S_{obs}, we also compute the top-1 classification result to infer the robustness of the system:

$$Acc = T_r / T_s \tag{13}$$

where T_r is the number of true species prediction, T_s represents total number of testing data

Full model finetuning (FMF): In the original set of experiments, we fine tuned only the generic layer in the species layer training, leaving the organ layer unaltered. Although we could extract the pre-initialised organ information and combine it with the generic information in the species layer training, organ and generic layer were not jointly trained. This process constrains the network to learn the co-adaptation of features between two components. In fact, in [64], it has been proven that fixing the weights of the higher level layers of a network will deteriorate the interaction between neurons, subsequently affecting the network optimization.

To further enhance the model, we consider adding finetuning on the organ layer together with the generic layer during the species layer training. We set the learning rate to be 10 times lower than the newly assigned species layer, so that, the organ layer weights are not altered too much. By doing so, the organ layer can be optimized so that it provides a better connection with the generic layer and, at the same time, retains its organ features. This improves the result by 1.7% for the top-1 classification measure compared to the baseline provided by M-S HGO-CNN.

Feature space normalization (FSN): To improve the training of M-S HGO-CNN, we employ a data layer normalization technique – batch normalization (BN) [65] that offers extra flexibility in learning the input distribution. We found that by adding BN before and after the fusion of organ and generic components helps enhancing the learning capability of the network, boosting its representation ability. The improvement achieved by this was 3.2% in the top-1 measure of the classification result compared to the baseline M-S HGO-CNN. Noted that, owing to data memory constraint, we only add BN starting from the last convolution layer (conv5_3_O and conv5_3) of each individual component up to the fully connected layers (conv6, conv7, fc6 and fc7). During species layer training, these layers have learning rate 10 times higher than the preceding layers.

Enhanced feature fusion (EFF): To enhance the performance of M-S HGO-CNN, we tested a new fusion function, $g_{sum} : \mathbf{x_{org}}, \mathbf{x_{gen}} \rightarrow \mathbf{y_{sum}}$. It is employed at stage L (after the last convolutional layer for both components as shown in Fig. 3(a)). Note that, unlike the g_{cat}, the function g_{sum} performs summation of the $\mathbf{x_{org}}$ and $\mathbf{x_{gen}}$ features: $\mathbf{y_{sum}}_{i,j,k} = \mathbf{x_{org}}_{i,j,k} + \mathbf{x_{gen}}_{i,j,k}$, where $1 \leq i \leq H$, $1 \leq j \leq W$, $1 \leq k \leq Z$. The feature maps $\mathbf{y_{sum}}$ will then go through convolution layers to learn the combined representation of generic and organ features. We found that by incorporating the two aforementioned techniques with this new feature fusion, the improvement gained is 3.6% for the top-1 measure of the classification result compared to the baseline M-S HGO-CNN. This shows that with fusion through summation can better amplify the important features for the network.

TABLE III: Evaluation of the improvement strategies for M-S HGO-CNN.

Method	Acc	S_{img}	S_{obs}	
			BD	MAV
Baseline	0.599	0.690	0.717	0.715
Full model finetuning	0.616	0.690	0.718	0.716
Feature space normalisation	0.631	0.704	0.730	0.730
Enhanced feature fusion	**0.635**	**0.710**	**0.737**	**0.736**

TABLE IV: Classification performance comparison of each content based on S_{img} for the enhanced model.

Method	Branch	Entire	Flower	Fruit	Leaf	LeafScan	Stem
Baseline	0.568	0.603	0.798	0.657	0.652	0.803	0.411
FMF	0.589	0.613	0.803	0.657	0.650	0.792	0.377
FSN	0.605	0.621	0.813	0.687	0.655	0.753	0.411
EFF	0.593	0.635	0.816	0.691	0.669	0.768	0.402

Table III clearly shows that all model enhancement techniques achieve a notable improvement in the top-1 accuracy as well as the S_{img} and S_{obs} results. We further evaluate the classification performance of each organ and document it in Table IV. We observe that, although ‘Branch’ and ‘Entire’ organs show a significant improvement using the enhanced feature fusion approach, other organs such as ’LeafScan’ or ’Stem’ show either not much improvement on or lower classification rate. This suggests that M-S HGO-CNN which operates on each plant image individually is not robust enough to predict all plant organs, especially those which are highly influenced by intra and interspecies diversity. Henceforth, this motivates us to explore the more sophisticated RNN architecture, which exploits the dependencies between plant views capturing one or more organs of a plant.

VII. Experiments on the Plant-StructNet

Before initiating the training of the Plant-StructNet, we firstly group the training and testing images into their respective observation ID. Note that, each observation ID consists of T number of plant images captured from a p-th plant observed by a u-th user. By doing so, we have 27907 and 13887 numbers of observation IDs for training and testing respectively. We apply the similar multi-scaling image augmentation to these plant images, and, extract their representation through the enhanced M-S HGO-CNN which obtained by the EFF approach mentioned in Sec. VI-D, we abbreviate it the E-CNN. During training, the extracted features are fed sequentially to each state $t = 1, ..., T$ of the Plant-StructNet. We fix the order of the plant images presented to the network based on the following sequence: branch, entire, flower, fruit, leaf, leafscan and stem. We test the performance of the Plant-StructNet using different levels of image abstraction representation extracted from $conv7$, $fc6$, and $fc8$ layers of E-CNN.

The Plant-StructNet is trained using the Tensorflow library [66]. We use the ADAM optimizer [67] with the parameters $\alpha = 1e-08$, $\beta1$ =0.9 and $\beta2$ =0.999. We set the learning rate to 0.0001. In all experiments, we use a mini-batch size of 30. We evaluate the Plant-StructNet on the same PlantClef2015 dataset using the same evaluation metrics.

A. Performance Evaluation

In this experiment, we compare the performance of the Plant-StructNet with E-CNN. We present a comparative performance evaluation of the Plant-StructNet based on different levels of image abstraction representation ($conv\_7$, $fc6$ and $fc7$). We also evaluate the performance of the Plant-StructNet when only forward directional states modeling is taken into consideration. For testing, we evaluate the architectures using

TABLE V: Performance comparison between the Plant-StructNet and the E-CNN. Note that, attn is the attention mechanism, ns is the number of stages of the attention module and Fsm is the forward states modeling.

Method	Acc	S_{img}
E-CNN	0.601	0.679
Fsm Plant-StructNet (conv7, 1 ns) + attn	0.514	0.587
Fsm Plant-StructNet (conv7, 3 ns) + attn	0.508	0.621
Fsm Plant-StructNet (fc6)	0.588	0.664
Fsm Plant-StructNet (fc7)	0.611	0.665
Plant-StructNet (fc6)	0.622	0.669
Plant-StructNet (fc7)	**0.641**	**0.680**

plant images that are isotropically rescaled to 256 × 256 pixels as explained in the Sec. III-B. Table V shows the performance comparison results.

To train the Plant-StructNet based on $conv\_7$, we incorporate an attention mechanism [68] to enhance the representation of the visual input. Such attention mechanism allows the model to look for the most pertinent local features of a plant image in each state. In some respect, it forces an explicit additional step in the reasoning process, identifying salient regions in a plant image by assigning different importance to features from different image regions. The attention mechanism is introduced by the $\boldsymbol{\epsilon_t}$ term, the weighted average of convolutional features that depends on the previous activation:

$$\boldsymbol{\zeta_t} = \mathbf{W_a}^T \tanh(\mathbf{W_{a1}}\boldsymbol{\delta_t} + \mathbf{W_{a2}}\overrightarrow{\mathbf{h_{t-1}}}) \quad (14)$$

$$\boldsymbol{\lambda_t} = softmax(\boldsymbol{\zeta_t}) \quad (15)$$

$$\boldsymbol{\epsilon_t} = \boldsymbol{\lambda}_{\boldsymbol{t}}^T \boldsymbol{\delta_t} \quad (16)$$

In this case, the attention term $\boldsymbol{\lambda_t}$ controls the contribution of each convolutional feature at the t-th state. Large values in $\boldsymbol{\lambda_t}$ indicate more importance of the corresponding region to the target species class. Note that, for language modeling tasks, based on images [69], [70], a similar image is refined by attention model across all the steps of the RNN. However, in our work, every step of the Plant-StructNet takes in different plant views that do not have a form of sequences. The spatial information captured by the attention mechanism in the current state, will not be relevant for the next state input. To address this issue, we introduce a multi-stage attention mechanism. Specifically, we add in a cascaded attention module that can refine every plant image in each state before proceeding to the subsequent state. For example, the 3-stage cascaded attention module as shown in Fig. 7. The new refined convolution features for the first and second stages are generated through: $\boldsymbol{\delta}_{\boldsymbol{t_{j+1}}} = \boldsymbol{\lambda}_{\boldsymbol{t_j}} \odot \boldsymbol{\delta}_{\boldsymbol{t_j}}$, where $j = 1, 2$. The computations performed by a GRU with attention mechanism are described as follows:

$$\overrightarrow{\mathbf{z_t}} = \sigma(\mathbf{w_{z\epsilon}}\boldsymbol{\epsilon_t} + \mathbf{w_{z2}}\overrightarrow{\mathbf{h_{t-1}}}) \quad (17)$$

$$\overrightarrow{\mathbf{v_t}} = \sigma(\mathbf{w_{v\epsilon}}\boldsymbol{\epsilon_t} + \mathbf{w_{v2}}\overrightarrow{\mathbf{h_{t-1}}}) \quad (18)$$

$$\overrightarrow{\mathbf{h_t}} = (1 - \overrightarrow{\mathbf{z_t}})\overrightarrow{\mathbf{h_{t-1}}} + \overrightarrow{\mathbf{z_t}}\overrightarrow{\tilde{\mathbf{h}}_{\mathbf{t}}} \quad (19)$$

$$\overrightarrow{\tilde{\mathbf{h}}_{\mathbf{t}}} = \tanh(\mathbf{w_{h\epsilon}}\boldsymbol{\epsilon_t} + \mathbf{w_{h2}}(\overrightarrow{\mathbf{v_t}} \odot \overrightarrow{\mathbf{h_{t-1}}})) \quad (20)$$

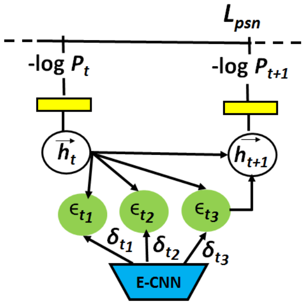


Fig. 7: The 3-stage cascaded attention module

TABLE VI: Comparison of top-1 classification accuracy for different categories of observation ID. Note that, Category A = number of images < 2 per observation ID; Category B = number of images ≥ 2 per observation ID

Category	A	B
Total number of testing images for each category	9905	11541
E-CNN	0.592	0.609
Plant-StructNet	**0.542**	**0.745**

TABLE VII: Classification performance comparison of each content based on S_{img}.

Method	Branch	Entire	Flower	Fruit	Leaf	LeafScan	Stem
E-CNN	0.564	0.573	0.801	0.657	0.666	0.759	0.384
Plant-StructNet	**0.596**	**0.582**	**0.798**	**0.673**	**0.714**	**0.762**	**0.540**

In Table V, we can see that using uni-directional states modeling, the Plant-StructNet trained on either the $fc6$ or $fc7$ features achieves much better results compared to the $conv\_7$ with attention mechanism. We think the reason for fc layers to perform better than the $conv$ layer is that fc layers hold more class-specific features which are less complex to be trained compared to the $conv$ layer. Next, using bi-directional states modeling is obviously better compared to uni-directional state modeling. This is understandable as bi-directional state modeling enables prediction of an image based on the holistic collection of data extracted from a same plant as explain in Sec. IV-A. We observe that the Plant-StructNet trained on $fc7$ can boost up the performance significantly, achieving a highest top-1 accuracy of 0.641 (improvement of 4% compared to the E-CNN). However, we found that its S_{img} does not seem to show different or better results. We then explore the cause and observe that most of the misclassification occurs when there is only one testing image per observation ID.

Table VI shows that there is a total of 9905 observation IDs that contains only one image, nearly 47% of the testing set. It is noticeable that the Plant-StructNet performs better for category B than A (top-1 accuracy of 0.75 compared to 0.54), while E-CNN performs almost equally in all cases for category A and B (top-1 accuracy of 0.59 and 0.61). This can be explained from the characteristic of both RNN and CNN based models used in this context. To recognize a plant image, the CNN based model is trained to look for similar patterns on all different subfields of an image, while the RNN based model is trained to look for higher level features modeling the dependencies between series of images. Based on these findings, we therefore deduce that the poor performance of

TABLE IX: Performance comparison with SOTA based on S_{img}.

Method	Branch	Entire	Flower	Fruit	Leaf	LeafScan	Stem	Overall
GoogLeNet + VGGNet [71]	0.621 0.666*	0.632 0.624*	0.828 0.861*	0.711 0.723*	0.652 0.745*	0.788 0.848*	0.411 0.429*	0.715 0.752*
Plant-StructNet (fc7) + E-CNN	**0.642**	**0.670**	**0.829**	**0.725**	**0.734**	**0.789**	**0.540**	**0.732**

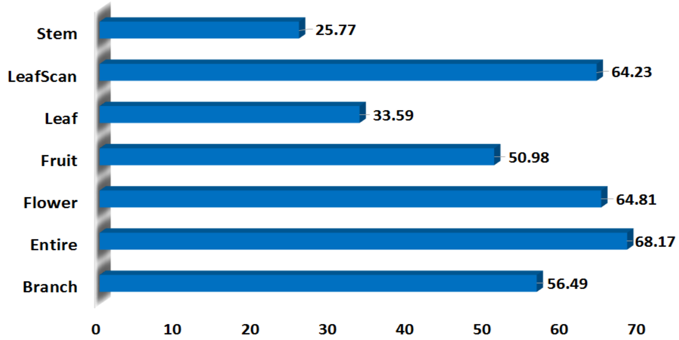


Fig. 8: Percentage of images that fall under category A for each organ category (%).

the Plant-StructNet is mainly due to the inadequate samples of plants given one observation ID.

Next, we compare the classification performance for each organ based on the image-centered score, S_{img}. We observe that the Plant-StructNet can essentially improve the recognition performance of each organ, especially the 'stem' organ as shown in Table VII. The percentage increase is 40.65% which is considerably significant compared to other organs. This improvement is explained by the fact that the stem organ has the least number of images falling under category A (as shown in Fig. 8). That is the majority of stem images co-exists with other plant images in one observation ID. For this reason, we can see that although the stem organ is considered as the least informative one compared to other organs, using the Plant-StructNet, we can successfully boost its classification performance.

We qualitatively evaluate the features learned in both the Plant-StructNet and E-CNN by projecting them into a two-dimensional space using t-SNE [17]. To ease the visualization for it is impractical to show all 1000 classes within a limited space, we have randomly selected 39 to clearly show in Fig. 9 and 10. We extract the $fc7$ and h_t features of the testing images from both the E-CNN and Plant-StructNet respectively. Fig. 9 visualizes the feature embedding of Plant-StructNet features while the Fig. 10 visualizes the feature embedding of E-CNN features. Note that, for Fig. 9(b) and Fig. 10(b), each point depicts the learned feature, and, it is represented by different color and symbol to distinguish different species classes. We observe that the Plant-StructNet features are semantically separable compared to E-CNN. This indicates that the features learned in Plant-StructNet are more discriminative compared to these of the E-CNN.

B. Assessing Performance in the Absence of Sequence

To model plant images in a sequential manner using the Plant-StructNet, we initially fix the order of the plant images presented to the network based on the sequence: branch, entire, flower, fruit, leaf, leafscan and stem. However, we notice that in reality, during their field work, botanists usually observe and study simultaneously a plant from different vantage points, as a whole and also analyse different organs. This drives us to extend the analysis of the Plant-StructNet on its capability of modeling plant views images irrespective of the order. Hence, we train the Plant-StructNet using the $fc7$ features based on random sequence, disregarding the order of the plant images fed into the network. Indeed, we found that training the Plant-StructNet by disregarding the order does not affect much the performance. The Acc and S_{img} obtained are 0.643 and 0.675 respectively which are comparable to the $Acc = 0.641$ and the $S_{img} = 0.680$ obtained from the model with sequence as shown in Table V. This finding again shows that the Plant-StructNet is able to process the complex structural dependencies between plant views/organ images despite the absence of sequence.

TABLE VIII: Evaluation of the ensemble models

Method	Acc	S_{img}	S_{obs}	
			BD	MAV
E-CNN	0.635	0.710	0.737	0.736
Plant-StructNet (fc6)	0.662	0.708	0.720	0.721
Plant-StructNet (fc7)	0.683	0.717	0.726	0.724
Plant-StructNet (fc6) + E-CNN	0.671	0.726	0.746	0.744
Plant-StructNet (fc7) + E-CNN	**0.685**	**0.732**	**0.747**	**0.746**

C. Ensemble Models

In this experiment, we introduce an aggregate of ensemble models to increase the performance of multi-organ plant classification. We incorporate the decisions of our proposed deep networks, E-CNN as well as the Plant-StructNet. We combine their softmax scores using an average fusion method. For testing, we use all the scaled images mentioned in Sec. III-B. Based on the experimental results shown in Table VIII, the ensemble architectures can essentially boost the performance of the individual E-CNN and Plant-StructNet, achieving the highest metric scores of S_{img} and S_{obs}.

Finally, we compare our best model with the latest SOTA [71] that proposed fine-tuning of pre-trained GoogLeNet [72] and VGGNet [63] models using the PlantClef2015 dataset. To make a fair comparison between our proposed method and the latest SOTA [71], we train and test both the VGGNet and GoogLeNet based on our proposed augmented multi-scaled plant images (Sec. III-B). We train both models using the reported training scheme in [71]. For testing, we first obtain the prediction results for each model, and finally combine them using their presented fusion technique.

Table IX shows the performance comparison results. Note that, values without $(*)$ are the results generated using our data augmentation, while values with $(*)$ are the results originally reported in [71]. We observe that our best model outperforms the SOTA with an overall S_{img} of 0.732 compared to 0.715, and, when comparing using the top-1 accuracy, our best model achieves 0.685 compared to 0.647. One intriguing finding is that, we essentially improve the classification performance of each plant organ, especially the 'stem' organ. This suggests the importance of modeling the correspondence between plant views (or organs) to further boost the discriminative power of the plant classification system.

Apart from that, compared to the original proposed overall S_{img} [71], it is noticeable using their data augmentation technique that extracting and scaling random patches from the original image, and subsequently augmenting them with image rotation can better characterize the plant data. Henceforth, we deduce that it is possible the classification performance of our best model would improve if we further enhanced the diversity of the plant dataset.

VIII. Conclusion

We have presented two plant classification frameworks: (1) the HGO-CNN which uses an end-to-end deep neural network to integrate both organ and generic features, and, capture the correlation of these complementary information for species classification; (2) the Plant-StructNet which offers extra flexibility in learning the relationship between plant views and supports classification based on varying number of plant images captured from a same plant. It is worth noting that using multi-scale training can further boost the discriminative power of the HGO-CNN model. We have also presented and analyzed in this paper various improvements we have made to our basic HGO-CNN and described the evaluation results which shown using enhanced feature fusion can better improve the model performance.

Based on our findings, it is clear that using the Plant-StructNet can essentially improve the classification performance, especially for the less distinctive 'stem' organ. This suggests the importance of learning the correspondence between plant views to boost the overall species recognition rate. Experiments on the PlantClef 2015 benchmark show the robustness of the ensemble models of the E-CNN and Plant-StructNet in classifying different plant organ images. With the help of feature visualisation, we further confirmed the effectiveness of our model. In the future, it would be interesting to consider integration of both CNN and RNN based models in order to simultaneously handle rich visual representation learning and context dependencies modeling within a fully end-to-end deep network.

Acknowledgment

References

[1] J. Wäldchen and P. Mäder, "Plant species identification using computer vision techniques: A systematic literature review," *Archives of Computational Methods in Engineering*, pp. 1–37, 2017.

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

[3] A. Joly, H. Goëau, P. Bonnet, V. Bakić, J. Barbe, S. Selmi, I. Yahiaoui, J. Carré, E. Mouysset, J.-F. Molino *et al.*, "Interactive plant identification based on social image data," *Ecological Informatics*, vol. 23, pp. 22–34, 2014.

[4] T.-L. Le, D. Dng, H. Vu, and T.-N. Nguyen, "Mica at lifeclef 2015: Multi-organ plant identification," in *Working notes of CLEF 2015 conference*, 2015.

[5] I. Dimitrovski, G. Madjarov, D. Kocev, and P. Lameski, "Maestra at lifeclef 2014 plant task: Plant identification using visual data." in *CLEF (Working Notes)*, 2014, pp. 705–714.

[6] H. Goëau, A. Joly, I. Yahiaoui, V. Bakic, A. Verroust-Blondet, P. Bonnet, D. Barthélémy, N. Boujemaa, and J.-F. Molino, "Plantnet participation at lifeclef2014 plant identification task," in *CLEF2014 Working Notes. Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.* CEUR-WS, 2014, pp. 724–737.

[7] B. Yanikoglu, Y. Tolga, C. Tirkaz, and E. FuenCaglartes, "Sabanci-okan system at lifeclef 2014 plant identification competition," in *Working notes of CLEF 2014 conference*, 2014.

[8] G. Szűcs, D. Papp, and D. Lovas, "Viewpoints combined classification method in image-based plant identification task," 2014.

[9] D. Paczolay, A. Bánhalmi, L. Nyúl, V. Bilicki, and Á. Sárosi, "Wlab of university of szeged at lifeclef 2014 plant identification task," in *Working notes of CLEF 2014 conference*, 2014.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] A. K. Reyes, J. C. Caicedo, and J. E. Camargo, "Fine-tuning deep convolutional networks for plant recognition," in *Working notes of CLEF 2015 conference*, 2015.

[12] S. Choi, "Plant identification with deep convolutional neural network: Snumedinfo at lifeclef plant identification task 2015," in *Working notes of CLEF 2015 conference*, 2015.

[13] J. Champ, T. Lorieul, M. Servajean, and A. Joly, "A comparative study of fine-grained classification methods in the context of the lifeclef plant identification challenge 2015," in *CLEF 2015*, vol. 1391, 2015.

[14] E. M. M. Ghazi, B. Yanikoglu, and M. Ozdemir, "Sabanci-okan system in lifeclef 2015 plant identification competition," in *Working notes of CLEF 2015 conference*, 2015.

[15] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, Vellinga, R. Planqué, A. Rauber, S. Palazzo, B. Fisher, and H. Müller, "Lifeclef 2015: multimedia life species identification challenges," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction.* Springer, 2015, pp. 462–483.

[16] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer vision–ECCV 2014.* Springer, 2014, pp. 818–833.

[17] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[18] S. H. Lee, Y. L. Chang, C. S. Chan, and P. Remagnino, "Hgo-cnn: Hybrid generic-organ convolutional neural network for multi-organ plant classification." ICIP, 2017.

[19] C. Zhao, S. S. Chan, W.-K. Cham, and L. Chu, "Plant identification using leaf shapes—a pattern counting approach," *Pattern Recognition*, vol. 48, no. 10, pp. 3203–3215, 2015.

[20] A. R. Sfar, N. Boujemaa, and D. Geman, "Confidence sets for fine-grained categorization and plant species identification," *International Journal of Computer Vision*, vol. 111, no. 3, pp. 255–275, 2015.

[21] J. Chaki, R. Parekh, and S. Bhattacharya, "Plant leaf recognition using texture and shape features with neural classifiers," *Pattern Recognition Letters*, vol. 58, pp. 61–68, 2015.

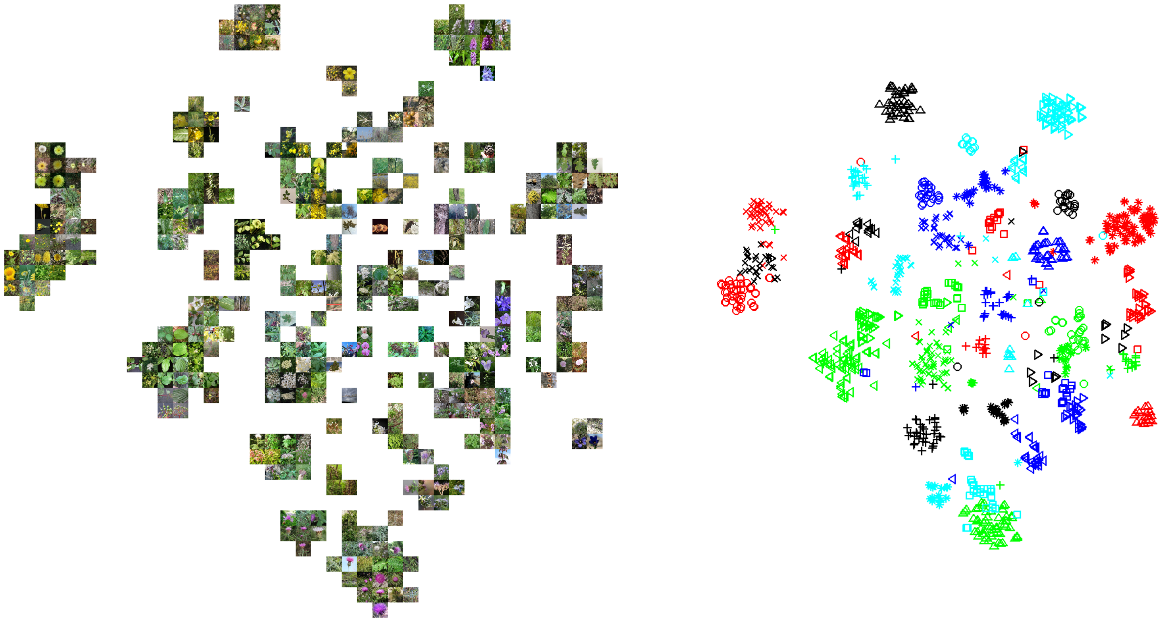
[22] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares, "Leafsnap: A computer vision system for automatic plant species identification," in *Computer Vision–ECCV 2012.* Springer, 2012, pp. 502–516.

[23] Y. Naresh and H. Nagendraswamy, "Classification of medicinal plants: An approach using modified lbp with symbolic representation," *Neurocomputing*, vol. 173, pp. 1789–1797, 2016.

[24] G. L. Grinblat, L. C. Uzal, M. G. Larese, and P. M. Granitto, "Deep learning for plant identification using vein morphological patterns," *Computers and Electronics in Agriculture*, vol. 127, pp. 418–424, 2016.

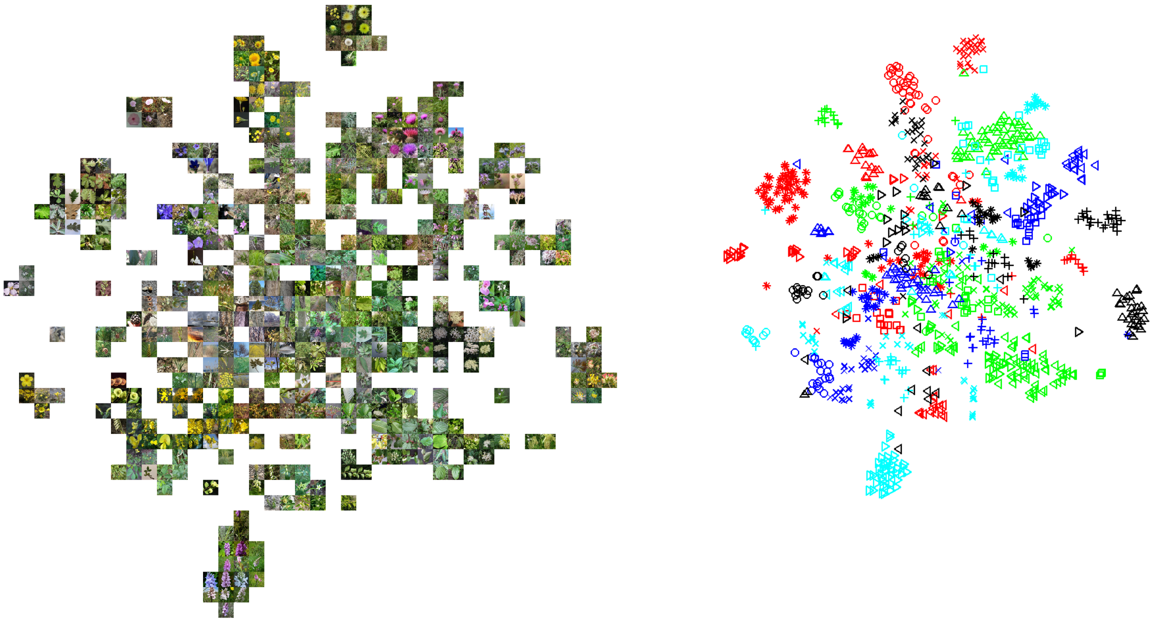
[25] J. Charters, Z. Wang, Z. Chi, A. C. Tsoi, and D. D. Feng, "Eagle: A novel descriptor for identifying plant species using leaf lamina vascular features," in *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on.* IEEE, 2014, pp. 1–6.

[26] M. G. Larese, R. Namías, R. M. Craviotto, M. R. Arango, C. Gallo, and P. M. Granitto, "Automatic classification of legumes using leaf vein image features," *Pattern Recognition*, vol. 47, no. 1, pp. 158–168, 2014.



(a) (b)

Fig. 9: Feature embedding visualizations of the Plant-StructNet using t-SNE. (a) Image visualization. (b) Scatter plot: points with the same color and symbol are the features belonging to the same species class. Best viewed in electronic form.



(a) (b)

Fig. 10: Feature embedding visualizations of the E-CNN using t-SNE. (a) Image visualization (b) Scatter plot: points with the same color and symbol are the features belonging to the same species class. Best viewed in electronic form.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2836321, IEEE Transactions on Image Processing

[27] M. Seeland, M. Rzanny, N. Alaqraa, J. Wäldchen, and P. Mäder, "Plant species classification using flower images—a comparative study of local feature representations," *PLoS One*, vol. 12, no. 2, p. e0170629, 2017.

[28] H.-H. Lee and K.-S. Hong, "Automatic recognition of flower species in the natural environment," *Image and Vision Computing*, vol. 61, pp. 98–114, 2017.

[29] T. Saitoh, K. Aoki, and T. Kaneko, "Automatic recognition of blooming flowers," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 1. IEEE, 2004, pp. 27–30.

[30] J. Fan, N. Zhou, J. Peng, and L. Gao, "Hierarchical learning of tree classifiers for large-scale plant species identification," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4172–4184, 2015.

[31] Z. Ge, C. Mccool, and P. Corke, "Content specific feature learning for fine-grained plant classification," in *Working notes of CLEF 2015 conference*, 2015.

[32] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *International Conference on Machine Learning*, 2016, pp. 1378–1387.

[33] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *International Conference on Machine Learning*, 2016, pp. 2397–2406.

[34] F. Meng, Z. Lu, Z. Tu, H. Li, and Q. Liu, "A deep memory-based architecture for sequence-to-sequence learning," *arXiv preprint arXiv:1506.06442*, 2015.

[35] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.

[36] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[37] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, "Multi-stream multi-class fusion of deep networks for video classification," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 791–800.

[38] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 652–663, 2017.

[39] K. Fu, J. Jin, R. Cui, F. Sha, and C. Zhang, "Aligning where to see and what to tell: Image captioning with region-based attention and scene-specific contexts," *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[40] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542.

[41] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4584–4593.

[42] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 21–29.

[43] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1–9.

[44] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *European Conference on Computer Vision*. Springer, 2016, pp. 451–466.

[45] F. Visin, K. Kastner, A. C. Courville, Y. Bengio, M. Matteucci, and K. Cho, "Reseg: A recurrent neural network for object segmentation," *CoRR, abs/1511.07053*, vol. 2, 2015.

[46] M. Ren and R. S. Zemel, "End-to-end instance segmentation with recurrent attention," in *CVPR*, 2017.

[47] B. Romera-Paredes and P. H. S. Torr, "Recurrent instance segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 312–329.

[48] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with lstm recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3547–3555.

[49] B. Shuai, Z. Zuo, B. Wang, and G. Wang, "Dag-recurrent neural networks for scene labeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3620–3629.

[50] H. Fan, X. Mei, D. Prokhorov, and H. Ling, "Multi-level contextual rnns with attention model for scene labeling," *arXiv preprint arXiv:1607.02537*, 2016.

[51] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[52] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2874–2883.

[53] R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-end people detection in crowded scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2325–2333.

[54] B. Ni, X. Yang, and S. Gao, "Progressively parsing interactional objects for fine grained action detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1020–1028.

[55] K. Gregor, I. Danihelka, A. Graves, D. Wierstra, and G. Deepmind, "Draw: A recurrent neural network for image generation," *CoRR*, 2015.

[56] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[57] Z. Yan, V. Jagadeesh, D. DeCoste, W. Di, and R. Piramuthu, "Hd-cnn: hierarchical deep convolutional neural network for image classification," *CoRR, abs/1410.0736*, 2014.

[58] A. Clark, "Whatever next? predictive brains, situated agents, and the future of cognitive science," *Behavioral and Brain Sciences*, vol. 36, no. 3, pp. 181–204, 2013.

[59] F. Meyniel, M. Maheu, and S. Dehaene, "Human inferences about sequences: A minimal transition probability model," *PLoS computational biology*, vol. 12, no. 12, p. e1005260, 2016.

[60] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, 2014.

[61] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.

[62] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[63] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[64] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[65] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[66] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[67] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[68] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel, "Visual question answering: A survey of methods and datasets," *Computer Vision and Image Understanding*, 2017.

[69] Y. Wang, Z. Lin, X. Shen, S. Cohen, and G. W. Cottrell, "Skeleton key: Image captioning by skeleton-attribute decomposition," *arXiv preprint arXiv:1704.06972*, 2017.

[70] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

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

[72] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.