Embedded plant recognition: a benchmark for low footprint deep neural networks
Mohammed El Amine Sehaba, Carlos Crispim-Junior, Laure Tougne Rodet

To cite this version:

HAL Id: hal-04236928
https://hal.science/hal-04236928
Submitted on 11 Oct 2023
Embedded plant recognition: a benchmark for low footprint deep neural networks

Mohammed El Amine Sehaba
Univ Lyon, Lyon 2, LIRIS, F-69676 Lyon, France
mohammed.sehaba@liris.cnrs.fr

Carlos Crispim-Junior
Univ Lyon, Lyon 2, LIRIS, F-69676 Lyon, France
carlos.crispimjunior@liris.cnrs.fr

Laure Tougne Rodet
Univ Lyon, Lyon 2, LIRIS, F-69676 Lyon, France
laure.tougne@liris.cnrs.fr

Abstract

Plant recognition is a challenging task due to the following elements: many classes, the variability of organs within a species, the similarity of organs between species, the shooting conditions, etc. There exist many mobile applications for plant recognition but most of them require an Internet connection to send the image to a server that will compute the recognition and, send the result back. However, in nature, in the mountains or in the forest, Internet connections are very often poor or non-existent. The only embedded plant recognition application is InterFolia based on SqueezeNet network but is it the best architecture to recognize plants? In this work, we propose to compare main existing networks that can be embedded allowing the recognition of plants from their organs (leaves, flowers/fruit, barks). The aim is to study how these networks behave in the face of this difficult problem to highlight their advantages and disadvantages in this context. The elements of comparison are not only the performance of the networks but also their memory impact and their inference time on computer and smartphone. Such elements could be extended to other applications in similar contexts, such as embedded phenotyping. We also propose a dataset with 477 plant classes that we make available to the scientific community1.

1https://kaggle.com/datasets/lauretougne/interfolia-dataset-2-0

1. Introduction

Our relationship with nature has profoundly changed in recent decades. While less than a third of the world’s population lived in cities in 1950, this will be the case for more than two thirds of the population in 2050 [10]. Human is therefore moving further and further away from nature.

Paradoxically, he has understood that he must protect his environment to survive. But we can only protect what we know well. In other words, environmental education is essential. It is therefore essential to learn to know the plants that surround us. This learning can be done through computer tools and especially smartphones that everyone now has in his pocket.

There exist many plant recognition applications freely available like Pl@ntNet Identify [8], Seek [9], PictureThis [7], LeafSnap [5], Google Lens [4], etc. but all these applications require a working Internet connection since the classification task is processed on a remote server. But in the nature, in the forest, in the mountains, etc. we do not necessarily possess an Internet connection. This implies to have embedded recognition algorithms. In this context, trade-offs must be made between response accuracy, processing speed and memory footprint. The only plant recognition application currently available that works without an Internet connection is InterFolia [33].

The objective of this work is precisely to compare different possible network models for the embedded recognition of trees and shrubs on the same dataset which contains many species (up to 184 per organ) and for which it is possible to use different organs (leaf, bark, flower/fruit).

In the remainder of this paper, we will first briefly de-
scribe the state of the art that on embedded plant recogni-
tion. We will then present the dataset that we have pro-
posed to establish these performance comparisons. We will
describe the architectures of the deep neural networks we
have chosen to compare and justify this choice. The sec-
tion 5 will present the different experiments conducted and
the results obtained.

2. Related work

Automatic plant recognition is a very developed research
topic since many years [34]. It is a difficult problem for
multiple reasons: firstly, there are many plant species in
nature, which results in many classes to discriminate. It
is estimated that there are 400,000 species of plants in the
world. Second, some species are not frequent and therefore
we sometimes have very few examples per class. Third,
some species have organs that may be visually quite dif-
f erent (Figures 1 and 2). It conducts to a very important
intra-class variability. In the same way, we can have organs
different species which are very similar to each other
(Fig. 1, right image); which has for effect also a similarity
inter-class sometimes very important. In addition to this,
the images are taken in an uncontrolled environment, which
can lead to artefacts of luminosity or cast shadows that can
disturb recognition. Figure 3 shows the large variability
of shape and color within flowers/fruits.

The methods developed in the literature were initially
based on the segmentation of the object to be recognized,
extraction of handcrafted features and classification of the
resulting feature vectors [16, 18, 29, 30]. The main problem
with this type of methods was its sensitivity to segmentation
quality. Indeed, poor segmentation implied features that did
not really correspond to the object, and consequently lead
to potential misclassification.

Then, deep learning based methods took over, as in other
application fields, and are now the only ones used [26, 17,
12, 23]. However, as deep learning methods are often com-
putationally and memory-intensive, they are still very often
deployed in remote servers.

The development of smartphone applications has given
rise to numerous applications that are either dedicated to
plants (e.g. PlantNet identify), or that allow to recognize
anything and in particular plants (e.g. Google Lens applica-
tion). Initially based on the extraction of handcrafted
characteristics, most of these applications are now based
on deep learning methods. The principle is the same for
all of them: the user classification request is sent to a re-
 mote server which processes the image and sends the an-
swer back to the smartphone. This framework allows to
use “large” deep neural networks and powerful computing
servers to run the classification task. However, as in nature
we often have either very little Internet connection or no
connection at all; these applications are incompatible with
this context of use. The only existing application that does
not require an Internet connection is InterFolia [33] which
embeds SqueezeNet networks because of their small size
and fast response time [22]. More specifically, three mod-
els are on board: one for recognizing leaves, one for barks
and one for flowers and fruits. The user can take several
photos of the different organs of the same plant and ask the
application to combine the results.

But is this network architecture still the most suitable
to address the problem? Can more recent network archi-
tectures be embedded in the current smartphones? For in-
stance, there are many “small” networks architecture that
are good candidates for the task, such as MobileNet fam-
ily (V2, V3) [25, 14], EfficientNet [32], SqueezeNet [21]
or MobileViT [24]. However, given the difficulty of the
plant recognition problem, it is difficult to predict which
network would be the most suitable. Comparisons made in
other contexts, such as, manufactured objects, do not corre-
spond to the conditions we experience in the context of plant
recognition in natural environments. For these reasons, we
propose InterFolia 2.0, a novel performance benchmark for
plant recognition task.

3. Datasets

There are many datasets in the literature concerning plant
species (Table 1). Several datasets contain a single organ.
For instance, leaves [28], barks [13][1], or flowers and
fruits [6][2][3]. However, in [28], for example, the leaves
are shot flat against a white background, which is not repre-
sentative of real-life conditions. In [6], these are ornamen-
tal flowers that are photographed, not tree or shrub flowers.
There exist also datasets mixing organs such as [19]. How-
ever, most of these datasets do not contain many classes
and therefore do not cover all the trees and shrubs that can
be found on a territory like France for example.

The datasets used in the ImageCleF challenges2 are rather
complete but do not differentiate the plant organs. Even if a
user may not be able to tell the difference between a flower
and a fruit from trees and shrubs due to its similarity for
non-experts, it is easy to tell the difference between a leaf
and a bark, a leaf and a fruit, or a leaf and a flower. Thus,
aiming at a smartphone use, we can consider recognizing
specifically these types of organs. Indeed, we can easily
ask the user to specify if the picture taken is of a leaf, a
bark or a flower/fruit. The idea is that by using a model
per plant organ, one hopes to recognize plant species more
easily. Prior work has shown that using several organs of
the same plant to recognize the plant species improves the
results [22].

For this, we require a dataset that organizes the images
of each plant species per organ. However, the only dataset

https://www.imageclef.org/
known to us with plant photos classified by species and organs is the dataset used by the InterFolia application, that is currently a private dataset.

### 3.1. InterFolia dataset

The InterFolia dataset has been used to train the deep neural network models used by the InterFolia application. It is based on the ImageCLEF dataset in which the examples

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of images</th>
<th># of classes</th>
<th>Barks</th>
<th>Leaves</th>
<th>Flowers/Fruits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bark-101 [1]</td>
<td>2.5k</td>
<td>101</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BarkNet [15]</td>
<td>23k</td>
<td>23</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BarkID [31]</td>
<td>18k</td>
<td>10</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Flavia [28]</td>
<td>1.9k</td>
<td>32</td>
<td>-</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Leafsnap [5]</td>
<td>28k</td>
<td>184</td>
<td>-</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>InterFolia [33]</td>
<td>9k</td>
<td>477</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 1. Existing plant datasets. The letter “X” denotes available information.
Table 2. Composition of the InterFolia dataset

<table>
<thead>
<tr>
<th>Organs</th>
<th>Barks</th>
<th>Leaves</th>
<th>Flowers/Fruits</th>
</tr>
</thead>
<tbody>
<tr>
<td># Images</td>
<td>1,236</td>
<td>2,536</td>
<td>5,245</td>
</tr>
<tr>
<td># Species (classes)</td>
<td>148</td>
<td>178</td>
<td>184</td>
</tr>
<tr>
<td># classes with 2-10 samples</td>
<td>118</td>
<td>107</td>
<td>51</td>
</tr>
<tr>
<td># classes with 11-50 samples</td>
<td>25</td>
<td>66</td>
<td>107</td>
</tr>
<tr>
<td># classes with 51-311 samples</td>
<td>5</td>
<td>5</td>
<td>26</td>
</tr>
</tbody>
</table>

Figure 4. Distribution of classes in InterFolia dataset 2.0.

have been classified by organ, in addition to species. Table 2 presents the distribution of the dataset for the different organs, as well as the number of species. We observe that there are fewer bark classes than leaf or flower/fruit because bark has no meaning for shrubs. We may also notice that InterFolia dataset is very unbalanced. For some classes, we have only 2 examples which is clearly insufficient to learn the variability of the organs. The left image of Figure 1 shows pictures of barks of the same species. We understand that with 2-3 photos per class (an organ of a species), a data-driven method, such as a deep neural network, is likely to face difficulty to learn a species model for accurate recognition.

3.2. InterFolia dataset 2.0

This section describes how we extended InterFolia dataset to obtain a richer version, for bark class recognition. The resulting dataset called InterFolia dataset 2.0 is available to the community on Kaggle. Firstly, we removed from the dataset the classes that contained less than 3 examples. This is done to be sure to have at least one image is available for the training, validation, and test sets.

Bark-101 dataset [27] is also based on PlantCLEF dataset and it contains 101 tree bark classes on which the
authors compared different recognition methods.

We merged both datasets to create InterFolia 2.0. This allowed us to get more classes and examples for barks. We have removed any duplicates that we found. Additionally, we have removed all shrub images in the Bark 101 dataset, to focus only on bark images of older trees. Table 3 presents the number of classes and samples per organ of the created dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># classes</th>
<th># samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaves</td>
<td>167</td>
<td>2,514</td>
</tr>
<tr>
<td>Flowers/Fruits</td>
<td>181</td>
<td>5,239</td>
</tr>
<tr>
<td>Barks</td>
<td>129</td>
<td>2,305</td>
</tr>
</tbody>
</table>

Table 3. Number of images for each organ.

We can see that we have a dataset with 477 classes (167 leaves, 181 flowers/fruits, 129 barks) and 10058 examples. This is already an interesting challenge in terms of classification when we know the nature of the objects to recognize, and the material constraints imposed by the smartphone. Figure 4 shows the class distribution for each organ type. We can see that the classes are also imbalanced, and that this remains a major challenge to overcome to achieve good recognition rates.

Finally, all the three new datasets were divided into train, validation, and test sets. At first, we split the dataset into 70% of images for training and 30% for testing. Then, the train set is divided again into 70% for training and 30% for validation. In the following, we focus on the architectures we compared and their training conditions.

4. Architectures and pretrained models

Using embedded networks necessarily involves trade-offs between performance, network size and inference time.

We first chose networks that are “light” in terms of memory impact but efficient. We also considered that it was important to compare them with “trendy” networks, like visual Transformers. Finally, we also added in our comparison the SqueezeNet network which is the network architecture currently used in the InterFolia application.

Table 4 compares the selected networks in terms of number of parameters, model size and efficiency on the ImageNet dataset. Notice that the ‘TOP-5’ error rate is the fraction of test images for which the correct label is amongst this top 5, and the ‘TOP-1’ error rate is the fraction of test images for which the correct label is the one judged most likely by the model.

Inference time of the four selected pre-trained models is counted in milliseconds for data augmentation. For both leaf and flower datasets, we train our models for 50 epochs, with a batch size of 32, and a learning rate of 1e-4 that decreases by 0.1 every 7 epochs. We apply categorical cross-entropy loss function and Adam optimizer. For the Flowers/Fruits dataset, we use the same training hyperparameters of other organ models, except that we train models for 10 epochs, with a batch size of 128.

We evaluate our models on several criteria. First evaluation criterion is the efficiency of the network, for that, we calculate the TOP-1 and TOP-5 accuracy. Secondly, since the target phones are limited in hardware resources, we take into consideration the memory size of the model, to ensure that the 3 networks (one per organ) can be embedded in addition to the elements necessary for the proper functioning of the application (example of images to be shown to the user for instance). The third criterion is the inference time of the network to ensure that the user will not have to wait too long for the result of his request.

5. Experiments and results

Note that all models are based on PyTorch framework, and they were pre-trained on ImageNet. For the evaluation of models on the smartphone, we converted all models using CoreML framework.

5.1. Training details and evaluation criteria

For the three sub-datasets, we use the software library Albumentations for data augmentation. For both leaf and bark datasets, we train our models for 50 epochs, with a batch size of 32, and a learning rate of 1e-4 that decreases by 0.1 every 7 epochs. We apply categorical cross-entropy loss function and Adam optimizer. For the Flowers/Fruits dataset, we use the same training hyperparameters of other organ models, except that we train models for 10 epochs, with a batch size of 128.

We evaluate our models on several criteria. First evaluation criterion is the efficiency of the network, for that, we calculate the TOP-1 and TOP-5 accuracy. Secondly, since the target phones are limited in hardware resources, we take into consideration the memory size of the model, to ensure that the 3 networks (one per organ) can be embedded in addition to the elements necessary for the proper functioning of the application (example of images to be shown to the user for instance). The third criterion is the inference time of the network to ensure that the user will not have to wait too long for the result of his request.

5.2. Results and discussion

First, we present the results obtained using PyTorch models on a MacBook computer. The MacBook configuration is not relevant because our goal here is to establish a baseline in terms of memory size and accuracy, that will
be later compared with the embedded version of the models running on the device.

Then, we convert those models to be compatible with an iPhone 12 Pro. This smartphone model has an Apple A14 Bionic processor equipped with 6 GB of RAM. The Apple A14 Bionic has a six-core 64-bit processor implementing ARMv8 ISA and incorporates a four-core @1000 MHz graphics processor designed by Apple. We convert all trained models to "neuralnetwork" type using CoreML-Tools, and allow it to use all compute units available on the target device.

The aim here is to compare the converted models in terms of memory size, accuracy, and inference time.

### 5.2.1 Results on the computer

Tables 5, 6 and 7 present the models’ results, respectively, on Leaves, Flowers/Fruits and Barks datasets. We compare models of the following architectures: MobileNet versions 2 (MV2) and 3 (MV3), EfficientNet version B0 (EB0), Mobile ViT (MViT) and SqueezeNet (SQ).

<table>
<thead>
<tr>
<th>Model</th>
<th>Memory size</th>
<th>TOP-1</th>
<th>TOP-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV2</td>
<td>10.0 MB</td>
<td>43.58%</td>
<td>63.44%</td>
</tr>
<tr>
<td>MV3</td>
<td>17.9 MB</td>
<td>47.81%</td>
<td>66.49%</td>
</tr>
<tr>
<td>EB0</td>
<td>17.2 MB</td>
<td>48.21%</td>
<td>66.75%</td>
</tr>
<tr>
<td>MViT</td>
<td>20.5 MB</td>
<td>48.21%</td>
<td>66.89%</td>
</tr>
<tr>
<td>SQ</td>
<td>3.3 MB</td>
<td>21.85%</td>
<td>33.11%</td>
</tr>
</tbody>
</table>

Table 5. Leaves - Results using models on MacBook.

<table>
<thead>
<tr>
<th>Model</th>
<th>Memory size</th>
<th>TOP-1</th>
<th>TOP-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV2</td>
<td>9.8 MB</td>
<td>34.99%</td>
<td>52.86%</td>
</tr>
<tr>
<td>MV3</td>
<td>17.7 MB</td>
<td>36.31%</td>
<td>54.17%</td>
</tr>
<tr>
<td>EB0</td>
<td>17.0 MB</td>
<td>31.19%</td>
<td>48.32%</td>
</tr>
<tr>
<td>MViT</td>
<td>20.4MB</td>
<td>35.29%</td>
<td>52.56%</td>
</tr>
<tr>
<td>SQ</td>
<td>3.2 MB</td>
<td>24.16%</td>
<td>37.92%</td>
</tr>
</tbody>
</table>

Table 7. Bark - Results using models on MacBook.

First, bark recognition is much more difficult than leaf or flower/fruit recognition. This is mainly because the shape of leaves is very characteristic. For flower/fruit, color is an important piece of information in addition to shape. For bark, color is not very discriminating, and only texture is. Moreover, texture varies over time with the age of the plant.

On Leaves dataset, the three networks MobileNet V3, EfficientNet-B0 and MobileViT-S are equally effective, but MVIT is slightly larger. MobileNet V2 has a lower efficiency than the other 3 architecture but it is almost half their size. SqueezeNet achieves very low performance.

On Flowers/Fruits dataset, EfficientNet-B0 is the best while MobileNet V3 and MobileViT-S are equivalent. SqueezeNet’s performance is still well below other networks.

Finally, on Barks dataset, MobileNet V3 is the best networks with MobileViT-S not far away but still with a larger size. EfficientNet-B0 has a little low performance on barks. SqueezeNet is still well below in terms of performance.

To conclude on these first tests, EfficientNet-B0, MobileNet V3 and MobileViT-S are the best performing methods; MobileViT-S remains the largest with the most parameters.

### 5.2.2 Results on the smartphone

Tables 8, 9 and 10 show the obtained results using the models converted to CoreML and running on a smartphone. We did not provide results for MobileViT model here because the other architectures we tested presented a higher performance with a smaller number of parameters.

First, we note a small overall loss of performance for all models on the flowers/fruits and barks datasets. On the leaves dataset, performance is slightly better on the smartphone, but we did not run the experiment many times. We therefore conclude that the results are broadly equivalent to those obtained on the computer.

Concerning the Leaves dataset, the best network is
clearly EfficientNet-B0 both in terms of performance and inference time. MobileNet V3 has a much higher inference time. MobileNet V2 has an interesting inference time, and a lower memory impact but it achieves a lower performance. On flowers/fruits dataset, EfficientNet-B0 remains the best compromise, as we observed during the experiment on the computer. Finally, it should be noted that MobileNet V2 has a very good behavior on the barks dataset which is a difficult dataset.

To conclude from these tests, the first element that comes out is that the bark dataset is very challenging. MobileNet V2 is recommended in this context for its performance, its inference time, and its memory impact.

For the other two sets (flowers/fruits and leaves), EfficientNet-B0 is the best compromise. SqueezeNet really does not seem to be the better solution for this problem, since its performances are very low.

### 6. Conclusion and future Work

In this paper, we presented a benchmark of model comparisons for embedded plant recognition.

We first proposed a dataset organized by species and organs that will be made available to the scientific community. This dataset is very challenging as recognizing plants in natural environment is. It also contains lots of classes for each organ and imbalanced classes. Efficiently combining the results from 2 or 3 organs of the same plant remains a challenge too in this context. The effectiveness of new methods can therefore be tested on this difficult dataset.

Secondly, we compared most interesting models of the literature to solve the problem of embedded plant recognition. The results show that depending on the type of organ to recognize, the same network is not always recommended. For organs that can be discriminated by their shape and color like leaves or flowers/fruits, EfficientNet-B0 is a good solution. For barks, which are more difficult to discriminate and based on texture, MobileNet V2 is interesting.

In any case, the SqueezeNet network currently used in the InterFolia application is not the one that achieves the best results. It might be worthwhile to change the networks embedded in InterFolia and study the benefit in terms of recognition.

It should be noted that these results obtained on tree and shrub organs found in metropolitan France are exportable to organs of other plant types. They could thus serve as a basis for embedded phenotyping research in other contexts.

In future work, research can be undertaken on the most efficient use of the results of the different classifiers. It will also be necessary to work on methods to overcome the lack of data on certain classes to improve the results.

### References


