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/fruits, 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 community[1].

[1] https://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 describe the state of the art that on embedded plant recognition. We will then present the dataset that we have proposed 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 section 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 different (Figures 1 and 2). It conducts to a very important intra-class variability. In the same way, we can have organs of 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 computationally 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 application). 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 remote server which processes the image and sends the answer 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 models 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 architectures be embedded in the current smartphones? For instance, there are many "small" networks architecture that are good candidates for the task, such as MobileNet family (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 correspond 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 representative of real-life conditions. In [6], these are ornamental flowers that are photographed, not tree or shrub flowers. There exist also datasets mixing organs such as [19]. However, 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 challenges[2] 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

[2] https://www.imageclef.org/

Figure 1. Examples of bark images from (left) the same species, (right) two different species.

Figure 2. Examples of leaf images from InterFolia dataset: the 4 images are from the same species.

Figure 3. Examples of flower/fruit images from InterFolia dataset to show variability.

Dataset	# of images	# of classes	Barks	Leaves	Flowers/Fruits
Bark-101 [1]	2.5k	101	X	-	-
BarkNet [15]	23k	23	X	-	-
BarkID [31]	18k	10	X	-	-
Flavia [28]	1.9k	32	-	X	-
Leafsnap [5]	28k	184	-	X	-
Oxford-102 [6]	8k	102	-	-	X
InterFolia [33]	9k	477	X	X	X

Table 1. Existing plant datasets. The letter "X" denotes available information.

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

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

Table 2. Composition of the InterFolia dataset

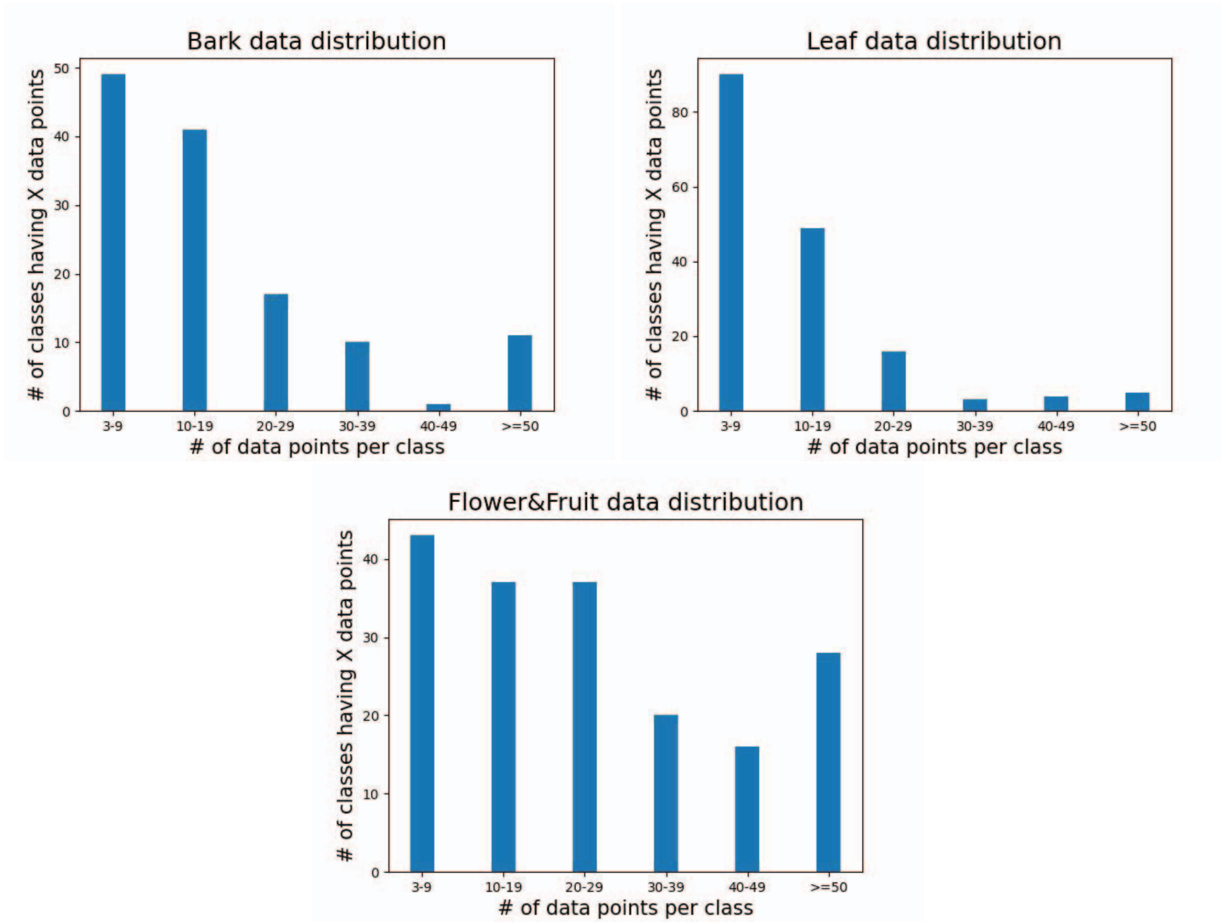

Figure 4. Distribution of classes in InterFolia dataset 2.0.

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.

Dataset	# classes	# samples
Leaves	167	2,514
Flowers/Fruits	181	5,239
Barks	129	2,305

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 [3]. 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 [20]. MobileNet V2 [25] was introduced by M Sandler et al. in 2018. This architecture is suitable for Mobile devices, or any devices with low computational power. In [14], A. Howard et al. have introduced a new version of MobileNet, that they called MobileNet V3. This model achieves a better accuracy on ImageNet than MobileNet V2. We therefore propose to compare the two architectures on our problem. In [32], T. Mingxing and Q. Le introduced a new family of neural networks called EfficientNet. EfficientNet architectures are very efficient computationally and achieves state-of-the-art results on the ImageNet dataset. Thus, we propose to compare the results obtained with the EfficientNet-B0 architecture. Finally, we include in this study a very small network (SqueezeNet [21]) which does not give very good results on ImageNet but which has the merit of not taking too much memory space. It is the architecture the authors of InterFolia have adopted. At the opposite end of the spectrum, we include a large, trendy network, MobileViT [24].

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 [11].

5.1. Training details and evaluation criteria

For the three sub-datasets, we use the software library Albumentations[4] 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

[3] https://pytorch.org/vision/stable/models.html

[4] https://albumentations.ai/docs/

Model	#params	Memory size	TOP-1	TOP-5
MV2	3.5M	13.6MB	71.87%	90.28%
MV3	5.5M	21.1MB	74.04%	91.34%
EB0	5.3M	20.5MB	77.69%	93.53%
MViT	5.6M	21.6MB	78.4%	94.1%
SQ	1.2M	4.8MB	58.09%	80.42%

Table 4. Pre-trained models using ImageNet (MV2: MobileNet V2, MV3: MobileNet V3, EB0: EfficientNet-B0, MVit: MobileViT-S, SQ: SqueezeNet).

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).

Model	Memory size	TOP-1	TOP-5
MV2	10.0 MB	43.58%	63.44%
MV3	**17.9 MB**	**47.81%**	**66.49%**
EB0	**17.2 MB**	**48.21%**	**66.75%**
MViT	**20.5 MB**	**48.21%**	**66.89%**
SQ	3.3 MB	21.85%	33.11%

Table 5. Leaves - Results using models on MacBook.

Model	Memory size	TOP-1	TOP-5
MV2	10.1 MB	45.10%	70.10%
MV3	17.9 MB	48.28%	71.18%
EB0	**17.2 MB**	**53.31%**	**75.45%**
MViT	20.6 MB	48.21%	71.82%
SQ	3.3 MB	13.36 %	27.54%

Table 6. Flowers/Fruits - 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.

Model	Memory size	TOP-1	TOP-5
MV2	9.8 MB	34.99%	52.86%
MV3	**17.7 MB**	**36.31%**	**54.17%**
EB0	17.0 MB	31.19%	48.32%
MViT	20.4MB	35.29%	52.56%
SQ	3.2 MB	24.16%	37.92%

Table 7. Bark - Results using models on MacBook.

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

Model	Mem size	Inference time	TOP-1	TOP-5
MV2	9.7MB	2.04ms	46.88%	66.22%
MV3	17.6MB	21.42ms	50.99%	69.66%
EB0	**16.8MB**	**2.62ms**	**50.59%**	**67.68%**
SQ	3.3MB	1.24ms	12.31%	23.97%

Table 8. Leaves - Results using models on iPhone 12 Pro.

Model	Mem size	Inference time	TOP-1	TOP-5
MV2	9.8MB	2.03ms	42.74%	68.23%
MV3	17.7MB	21.65ms	46.47%	70.26%
EB0	**16.9MB**	**2.67ms**	**51.50%**	**75.49%**
SQ	3.3MB	1.23ms	12.81%	27.97%

Table 9. Flowers/Fruits - Results using models on iPhone 12 Pro.

Model	Mem size	Inference time	TOP-1	TOP-5
MV2	**9.5MB**	**1.86ms**	**29.57%**	**48.02%**
MV3	17.4MB	21.56ms	30.16%	48.16%
EB0	16.6MB	2.66ms	26.20%	45.53%
SQ	3.2MB	1.23ms	16.25%	29.06%

Table 10. Bark - Results using models on iPhone 12 Pro.

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

[1] Bark-101. http://eidolon.univ-lyon2.fr/r̃emi1/Bark-101/, 2023-02-06. 2, 3

[2] Cifar-100. https ://www.cs.toronto.edu/k̃riz/cifar.html, 2023-02-06. 2

[3] Fids30. http ://www.vicos.si/Downloads/FIDS30, 2023-02-06. 2

[4] Google lens. https://play.google.com/store/apps/details?id=com.google.ar.lenshl=frgl=US, 2023-02-06. 1

[5] Leafsnap. https://play.google.com/store/apps/details?id=com.fws.plantsnap2hl=frgl=US, 2023-02-06. 1, 3

[6] Oxfordflower. https://www.robots.ox.ac.uk/ vgg/data/ flowers/, 2023-02-06. 2, 3

[7] Picturethis. https://play.google.com/store/apps/details?id=cn.danatech.xingseushl=frgl=US, 2023-02-06. 1

[8] Plantnet identification. https://play.google.com/store/apps/details?id=org.plantnethl=frgl=USpli=1, 2023-02-06. 1

[9] Seek. https://apps.apple.com/fr/app/seek-by-inaturalist/id1353224144, 2023-02-06. 1

[10] World urbanization prospects 2018. https://population.un.org/wup/, 2023-02-06. 1

[11] Coremltools. https://coremltools.readme.io/docs, 2023-02-07. 5

[12] A. Beikmohammadi et al. Leaf classification for plant recognition with deep transfer learning. *4th ICSPIS.*, 2018. 2

[13] A. Porebski et al. A new benchmark image test suite for evaluating color texture classification schemes. *Multimedia Tools and Applications Journal, vol. 70, no. 1, pp. 543–556*, 2014. 2

[14] Chu Liang-Chieh et al. Searching for mobilenetv3. *Proceedings of the IEEE/CVF ICCV*, 2019. 2, 5

[15] C. Mathieu et al. Tree species identification from bark images using convolutional neural networks. *International Conference on Intelligent Robots and Systems (IROS)*, pages 1075–1081, 10 2018. 3

[16] Dalcimar Casanova et al. Ifsc/usp at imageclef 2012 : Plant identification task. *CLEF, year=2012,*. 2

[17] D. Misra et al. Patch-based CNN evaluation for bark classification. In *Workshop on CVPPA*, Aug. 2020. 2

[18] G. Cerutti et al. Understanding leaves in natural images – a model-based approach for tree species identification. *CVIU*, 117(10):1482–1501, 2013. 2

[19] Hervé Goëau et al. The imageclef 2013 plant identification task. *CLEF*, 2013. 2

[20] Jeff Zhu et al. An improved one millisecond mobile backbone. *arXiv preprint arXiv:2206.04040*, 2022. 5

[21] Li Shen Jie Hu et al. Squeeze-and-excitation networks. *CVPR*, 2018. 2, 5

[22] M. Langlade et al. InterFolia: an embedded, educational application to identify plant species in the wild. In *CVPPA - ICCV Workshop*, Oct. 2021. 2

[23] Maha Saadeh et al. Flower classification using deep convolutional neural network. *IET Computer Vision.*, 2018. 2

[24] M. Sachin et al. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *ICLR*, 2022. 2, 5

[25] Menglong Zhu et al. Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE CVPR*, 2018. 2, 5

[26] P. Giguere et al. Tree species identification from bark images using convolutional neural networks. *In: IEEE/RSJ IROS*, 2018. 2

[27] R. Ratajczak et al. Efficient Bark Recognition in the Wild. In *VISAPP 2019*, Feb. 2019. 4

[28] Stephen Gang Wu et al. A leaf recognition algorithm for plant classification using probabilistic neural network. *Signal Processing and Information Technology*, 2007. 2, 3

[29] Thi-Lan Le et al. Complex background leaf-based plant identification me- thod based on interactive segmentation and kernel descriptor. *2nd International Workshop on EMR. ACM.*, 2015. 2

[30] Thi T.-N. Nguyen et al. A combination of deep learning and hand-designed feature for plant identification based on leaf and flower images. *ACIIDS*, 2017. 2

[31] W. Fanyou et al. Deep barkid: a portable tree bark identification system by knowledge distillation. *European Journal of Forest Research*, pages 1391–1399, 2021. 3

[32] Tan Mingxing and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *International conference on machine learning. PMLR.*, 2019. 2, 5

[33] University of Lyon. Interfolia. https://apps.apple.com/tn/app/InterFolia/id1570104475, 2023-02-06. 1, 2, 3

[34] M Rajeesh and A Ranjith Ram. Automatic plant recognition : A review. *Proceedings of the ICSEE*, 2021. 2