

# PL@NTNET APP IN THE ERA OF DEEP LEARNING

**Antoine Affouard, Hervé Goeau, Pierre Bonnet**

Pl@ntNet project, AMAP joint research unit, France

{antoine.affouard, herve.goeau, pierre.bonnet}@cirad.fr

**Jean-Christophe Lombardo, Alexis Joly**

Pl@ntNet project, Inria, ZENITH team, France

{jean-christophe.lombardo, alexis.joly}@inria.fr

## ABSTRACT

Pl@ntNet is a large-scale participatory platform and information system dedicated to the production of botanical data through image-based plant identification. In June 2015, Pl@ntNet mobile front-ends moved from classical hand-crafted visual features to deep-learning based image representations. This paper gives an overview of today's Pl@ntNet architecture and discusses how the introduction of convolutional neural networks did improve the whole workflow along the years.

## 1 INTRODUCTION

Pl@ntNet is a large-scale participatory platform and information system dedicated to the production of botanical data through image-based plant identification. It offers 3 main front-ends, an Android app (the most advanced and the most used one), an iOS app (being currently re-developed) and a web interface<sup>1</sup>, each allowing to submit one or several pictures of a plant in order to get a list of the most likely species in return. As illustrated by Figure 2 in the appendix, the application is becoming more and more popular especially during spring and summer. Nowadays, the application is translated in eleven languages and has been downloaded by more than 3 millions users in about 170 countries (it is often referred to as the *shazam of plants*). One of the main strength of Pl@ntNet is that the training set enabling the automated recognition is collaboratively enriched and revised. Indeed, users can share their observation with the community (whether they have identified it or not). Several collaborative tools are then available for data validation:

- a web platform called *IdentiPlante*<sup>2</sup> that is hosted by one of the largest network of amateur and expert botanists in the world (TelaBotanica, about 30K French speaking members).
- A gamified web application called *ThePlantGame*<sup>3</sup> that is based on a set of data-driven algorithms allowing to (i) actively train annotators, and (ii) evaluate the quality of contributors answers on new test items in order to optimize predictions.
- embedded validation mechanisms in the Android and Web front-ends themselves

All these validation tools allow the app to cover a growing number of floras and species. It was initially restricted to a fraction of the European flora (in 2013) and has then been extended to the Indian ocean flora and south American one (in 2015), and last year to the north African flora (2016). Numbers of species covered by the app in January 2017 are provided in Table 4 of the appendix.

## 2 PL@NTNET ARCHITECTURE

Figure 1 presents an overview of Pl@ntNet architecture (at the beginning of 2017). We give hereafter some details of the main modules:

---

<sup>1</sup>identify.plantnet-project.org

<sup>2</sup>www.tela-botanica.org/appli:identiplante

<sup>3</sup>www.theplantgame.com

**Plant Observations Storage:** Plant observations are the basic data unit of PI@ntNet information system. An observation documents evidence of a living individual plant. It includes one or several images of the plant tagged with a view type (flower, leaf, fruit, stem or entire plant) as well as some provenance data (device, author, date, etc.). Each observation is optionally associated to a geo-location and to one or more determinations, *i.e.* possible species names proposed by the author himself, and/or by other annotators, and/or by automated classifiers. A determination is tagged as valid when its confidence score (computed through different inference algorithms) exceeds some threshold. Observations are stored within a NoSQL document storage called CouchDb.

**Image representation learning and extraction:** Image representations are computed by a convolutional neural network (CNN) that is periodically trained in a supervised manner on the observations with a valid determination name and an additional rejection class (containing non-plant pictures taken by PI@ntNet users, *e.g.* faces, animals, manufactured objects, etc.). At the time of writing, the used CNN architecture is the inception model Szegedy et al. (2015) extended with batch normalization Ioffe & Szegedy (2015). The network is pre-trained on ImageNet dataset Deng et al. (2009) and periodically fine-tuned on PI@ntNet data. The number of species (*i.e.* classes) in January 2017 was about 10K and the number of training images about 332K.

**Species prediction:** As an observation might be composed of several pictures of the observed plant, the predictions of the CNN for each searched picture need to be fused. We therefore use a weighted average of the SOFTMAX probability vectors. The weight of each picture only depends on its view type (*e.g.* the pictures tagged as *flower* are more weighted than the pictures tagged as *leaf* because flowers are much more discriminant than leaves). The values of the weights have been optimized empirically. After this fusion process, a species filtering is applied based on the *checklist* of species activated within the app (*e.g.* West Europe, North Africa, South America, etc.). By default, this checklist is automatically chosen according to the geo-location of the mobile device, but the user also has the possibility to select another one manually. Note that all the checklists share some common species. The degree of overlap depends on environmental the factors of the covered areas.

**Similarity Search:** In addition to the most probable species, PI@ntNet's search engine returns the images of the dataset that are the most similar to the queried observation. This allows improving the interactive identification process by illustrating the candidate species with images consistent with the observed individual plant (the variability within the same species can indeed be high as well as the shooting protocol). This content-based image retrieval is performed through a hashing-based approximate nearest neighbors search algorithm applied on top of the 1024-dimensional feature vectors extracted by the last hidden layer of the fine-tuned CNN. The feature vector of each image is compressed into a compact binary code thanks to an unsupervised hashing method (RMMH: Joly & Buisson (2011)) and its approximate k-nearest neighbors are searched by probing neighboring buckets in the hash table (using the multi-probe algorithm described in Joly & Buisson (2008)). Note that a separate hash table is used for each checklist to focus the search on the right species and facilitate the distribution of the whole index. If a candidate species does not have any representative image in the returned top-k, it is not displayed to the user (actually, this adds a distance rejection criterion to the recognition process).

**Web API and front-ends:** Data exchanges between the search engine (server side) and the front-ends (client side) are managed through a REST-full web API using json data format and jpeg images. The list of candidate species predicted for a searched plant observation is displayed in decreasing order of confidence and each species is illustrated by the most similar picture of that species. If the user selects a species, the other retrieved images are displayed by decreasing similarity scores (grouped by type of view). At this stage, the user can access to richer information about the species such as all the available illustrations in the dataset (allowing to refine the identification) or descriptive web pages of the species providing useful information about the plant.

### 3 THE CONTRIBUTION OF IMAGE REPRESENTATIONS LEARNING

PI@ntNet's visual search engine started working with learned image representations in June 2015 (thanks to the integration of CNNs). Before that date, both the species prediction and the similarity search were based on a combination of hand-crafted visual features as described in Goëau et al.

(2014). We provide hereafter three evaluation dimensions showing the benefit of having moved to learned representations.

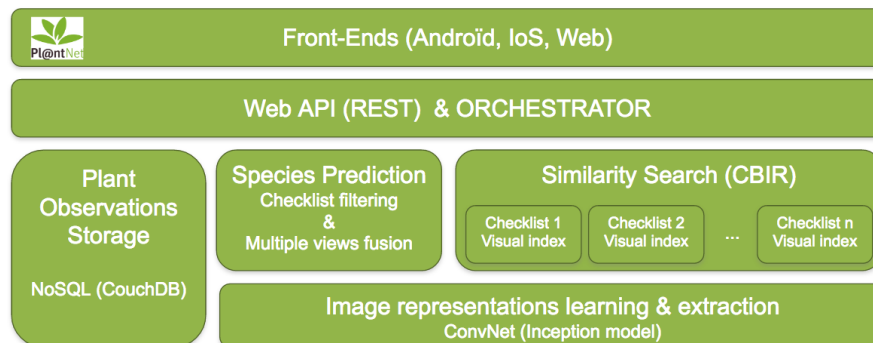


Figure 1: PI@ntNet system architecture

### Recognition performance improvement

Table 4 in the appendix gives an overview of the identification performance of PI@ntNet system along the years, as measured within the yearly system-oriented evaluation campaign LifeCLEF<sup>4</sup> (see references in the table for details about the metric, dataset, evaluation protocol and overall results of each year). The table shows that, despite the increasing difficulty of the task (more species, more variety and more noise along the years), a huge performance gap was observed in 2015 with the introduction of CNN's. This gap was also observed for other regular participants and, in 2016, 14 of the 15 participants were using deep learning in their system as described in Goëau et al. (2016).

### User's perception

Figure 3 in the appendix gives an overview of PI@ntNet user's ratings collected through the digital media store *Google Play store* (where the Android front-end is distributed). It shows that the monthly average rating gradually increased from about 3.2 to about 4.1 after the deep-learning based image representations were integrated in June 2015. It is likely that other improvements of the application also contributed to a better perception (actually, the ratings were already increasing beforehand). But the integration of the CNN clearly accelerated the increase and is one of the main explanatory factor.

### Qualitative evidences

Figure 4 in the appendix provides some qualitative evidences of the benefit of using CNN-based image representations within PI@ntNet application. For 4 observations made in March 2015, it provides the results returned by the mobile app at that time (using on hand-crafted visual features) compared to the results returned today (using the architecture described in section 2). It shows that beyond the raw increase of the identification performance, the search engine now returns results that are much more consistent with the query image. Even for the wrong species predictions, the returned similar images share easily interpretable visual patterns with the observed plant and they often correspond to species with known confusion with the true one. In practice, this makes the application a much more user-friendly and pedagogical tool.

## 4 CONCLUSION

This paper describes for the time the architecture of the PI@ntNet information system since convolutional neural networks were introduced in June 2015. It illustrates in what way moving from hand-crafted image representations to the use of deep learning in an information system can have a consistent societal impact. Today, PI@ntNet is largely used in various professional contexts (agriculture, education, ecotourism, etc.). It contributes to develop an increasing interest of a large part of the society to their environment as well as new forms of educational trainings in botany and ecology.

<sup>4</sup>[www.lifeclef.org](http://www.lifeclef.org)

## REFERENCES

- Vera Bakic, Itheri Yahiaoui, Sofiene Mouine, Saloua Litayem Ouertani, Wajih Ouertani, Anne Verroust-Blondet, Hervé Goëau, and Alexis Joly. Inria imedia2's participation at imageclef 2012 plant identification task. In *CLEF working notes*, 2012.
- Vera Bakic, Sofiene Mouine, Saloua Ouertani-Litayem, Anne Verroust-Blondet, Itheri Yahiaoui, Hervé Goëau, and Alexis Joly. Inria's participation at imageclef 2013 plant identification task. In *CLEF working notes*, 2013.
- Julien Champ, Titouan Lorieul, Maximilien Servajean, and Alexis Joly. A comparative study of fine-grained classification methods in the context of the lifeclef plant identification challenge 2015. In *CLEF working notes*, volume 1391, 2015.
- Julien Champ, Hervé Goëau, and Alexis Joly. Floristic participation at lifeclef 2016 plant identification task. In *CLEF working notes*, pp. 450–458, 2016.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. IEEE, 2009.
- Hervé Goëau, Alexis Joly, Itheri Yahiaoui, Vera Bakić, Anne Verroust-Blondet, Pierre Bonnet, Daniel Barthélémy, Nozha Boujemaa, and Jean-François Molino. Plantnet participation at lifeclef2014 plant identification task. In *CLEF2014 Working Notes. Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, pp. 724–737. CEUR-WS, 2014.
- Hervé Goëau, Pierre Bonnet, and Alexis Joly. Plant identification in an open-world (lifeclef 2016). *CLEF working notes*, 2016, 2016.
- Joly A. Yahiaoui I. Bonnet P. Mouysset E. Goëau, H. Participation of inria pl@ntnet to imageclef 2011 plant images classification task. In *CLEF working notes*, 2011.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- Alexis Joly and Olivier Buisson. A posteriori multi-probe locality sensitive hashing. In *Proceedings of the 16th ACM international conference on Multimedia*, pp. 209–218. ACM, 2008.
- Alexis Joly and Olivier Buisson. Random maximum margin hashing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 873–880. IEEE, 2011.
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

## APPENDIX

height Floras	#Species cover by PI@ntNet	#Flora species	#Percentage
Western Europe	6220	7268	85.58
Indian Ocean	1204	2541	47.38
French Guyana	928	8047	11.53
North Africa	2742	8305	33.01

Table 1: Synthesis of the number of species covered by PI@ntNet

Years	#Species	#Images	Score (mAP)	Used image representations	Reference
2011	70	5,436	<b>0.197</b>	Hand-crafted local features	Goëau (2011)
2012	126	11,572	<b>0.22</b>	Hand-crafted local features	Bakic et al. (2012)
2013	250	26,077	<b>0.385</b>	Hand-crafted local features	Bakic et al. (2013)
2014	500	60,961	<b>0.289</b>	Hand-crafted local features	Goëau et al. (2014)
2015	1,000	113,205	<b>0.609</b>	Learned through CNN	Champ et al. (2015)
2016	+1,000	121,205	<b>0.627</b>	Learned through CNN	Champ et al. (2016)

Table 2: Synthesis of the performance of PI@ntNet system at LifeCLEF plant challenge

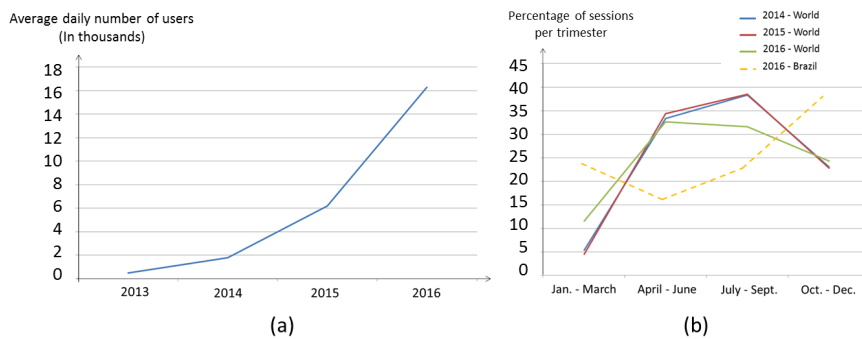


Figure 2: (a) Average daily number of users (b) Percentage of sessions per trimester

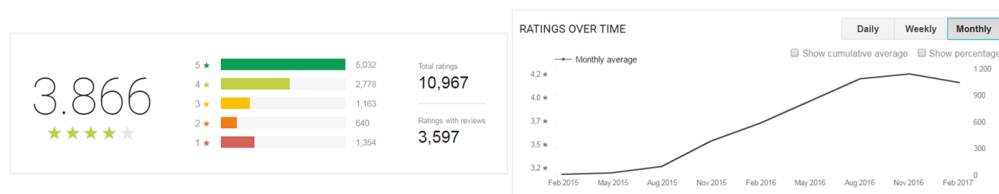


Figure 3: User's ratings of PI@ntNet android app





































Query		Top 1 species	Top 2 species	Top 3 species	Top 4 species	Top 5 species
	hand-crafted	 Syringa vilgaris	 Cercis siliquastrum	 Allium sphaerocephalon	 Amorpha fruticosa	 Malva sylvestris
	CNN	 Malva sylvestris	/			
	hand-crafted	 Aphyllanthes monspeliensis	 Cichorium intybus	 Primula farinosa	 Primula latifolia	 Pulmonaria angustifolia
	CNN	 Plumbago auriculata	 Plumbago europaea	/		
	hand-crafted	 Quercus pubescens	 Potentilla indica	 Acer campestre	 Acer saccharinum	 Asplenium viride
	CNN	 Quercus pubescens	 Quercus robur	 Quercus pyrenaica	 Quercus coccifera	/
	hand-crafted	 Reynoutria japonica	 Allium schoenoprasum	 Torilis arvensis	 Fumaria reuteri	 Prunus dulcis
	CNN	 Tamarix gallica	 Salvia leucantha	 Himantoglossum robertianum	 Phytolacca americana	 Phyteuma spicatum

Figure 4: Results returned by Pl@ntNet application before and after the introduction of CNN's (the species in red corresponds to the true label of the query)