Participation of LIRMM / Inria to the GeoLifeCLEF 2020 challenge

Benjamin Deneu[1,2], Maximilien Servajean[3], Pierre Bonnet[4], François Munoz[5], Alexis Joly[1]

[1] INRIA, Zenith Team, UMR LIRMM, Univ Montpellier, France.
[2] AMAP, Univ Montpellier, CIRAD, CNRS, INRAE, IRD, Montpellier, France.
[3] LIRMM, Université Paul Valéry, Univ Montpellier, CNRS, Montpellier, France
[4] CIRAD, UMR AMAP, F-34398 Montpellier, France.
[5] Université Grenoble Alpes, 38400 Saint-Martin-d'Hères, France,

Abstract. This paper describes the methods that we have implemented in the context of the GeoLifeCLEF 2020 machine learning challenge. The goal of this challenge is to advance the state-of-the-art in location-based species recommendation on a very large dataset of 1.9 million species observations, paired with high-resolution remote sensing imagery, land cover data, and altitude. We provide a detailed description of the algorithms and methodology, developed by the LIRMM / Inria team, in order to facilitate the understanding and reproducibility of the obtained results.

Keywords: LifeCLEF, biodiversity, environmental data, species distribution, evaluation, benchmark, Species Distribution Models, methods comparison, presence-only data, model performance, prediction, predictive power

1 The GeoLifeCLEF 2020 challenge

Predicting a list of the most likely species present at a given location is a key tool in biodiversity inventories as well as in the involvement of non-expert nature observers by highlighting candidate species available at a particular location. In addition, it could be used for educational purposes through biodiversity discovery applications providing innovative features such as contextualized educational pathways. Since several years, the development of mobile apps, whether for professional (such as land management, research or educational activities) or personal activities (for entertainment for example), which provide access to information characterising the biodiversity of a given site according to the user' geolocation, has been growing steadily. They make it possible, among other things, to increase the user' experience and to facilitate their contribution. Users produce much higher quality data, by avoiding for example the records of species

in a geographical area where they could not develop. The growing demand of geolocation-based biodiversity service, has led the development of the GeoLifeCLEF challenge [6, 4], in the context of the LifeCLEF[] evaluation campaign.

This challenge is linked to a particular type of species distribution models (SDM) where the objective is to predict the species most likely to be present at a given point. The models do not predict presence or probability of presence, nor densities, but a relative probability of presence of the species in relation to each other.

In the recent years and in previous participations to the GeoLifeCLEF challenge [7, 5], convolutional neural networks have shown to outperform more traditional strategies by producing better predictions, especially on rare species. Their performance seems to be due to two key points: (i) first, the possibility of processing large datasets that are inaccessible to other approaches; (ii) second, by learning a representation space common to all species, stabilizing predictions from one species to another, especially for the less represented ones.

In this paper, we detail the participation of LIRMM / Inria at the GeoLifeCLEF challenge 2020 [3, 2]. The particularity of this year challenge is to focus on a large common high resolution dataset covering France and USA. In addition to have for the first time an international dataset with the addition of USA, this year, the dataset contains for each occurrence a high resolution tensor including remote sensing imagery, elevation and land cover at one meter per pixel. An other improvement of this year edition is the evaluation protocol. The metric has been adapted for more flexibility and the split between train and test is now spatially done to avoid biases due to spatial auto-correlation.

A detailed description of the challenge methodology, data and results is provided in [2, 1]. Figure 1 provides an overview of the results of the challenge. In a nutshell, LIRMM / Inria submitted three categories of runs. A first one based on a random forest (RF) (trained on environmental vectors). A second one with a convolutional neural network (CNN) only based on high resolutions patches (this CNN has been trained using a cross entropy and cross-validated using a top-1, top-10 and top-30 metric). Finally a fusion model from the output of the convolutional neural network and the random forest has been submitted.

The following sections of this manuscript have been organized as follows : Section 2 gives an overview of the various data we used to build our model and of the official metric used in the challenge. Section 3 provides the detailed description of our implemented methods. Section 4 discusses the results.

2 Data and metric

This section briefly introduces the data used by our models. A more detailed presentation of the dataset used for the GeoLifeCLEF challenge 2020 is available at [1].

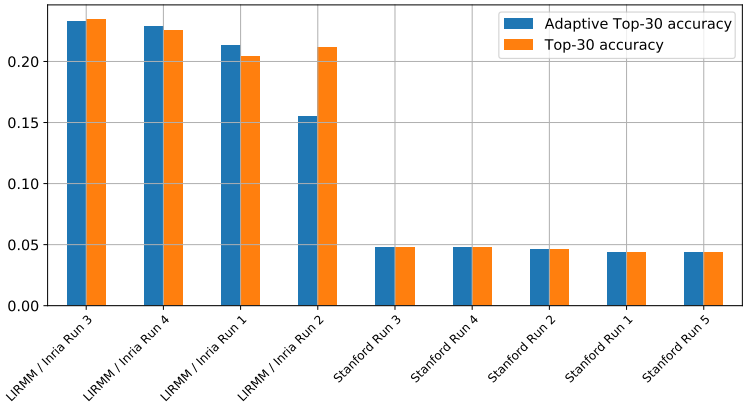

Fig. 1: Adaptive top-30 accuracy and top-30 accuracy per run and participant on GeoLifeCLEF 2020 task.

2.1 Occurrences

The occurrence dataset is a large presence only dataset containing 1.9M occurrences of plants and animals covering USA and France. The occurrences come from two citizen science programs iNaturalist and Pl@ntNet. All the 1.1M american occurrences come from iNaturalist (animals and plants) as well as the french animal ones. The french plant occurrences come from Pl@ntNet platform [8]. The split between train and test sets was processed through a spatial block holdout. To do so, all occurrences were considered on a grid of 5 by 5km and all occurrences falling in 2.5% randomly selected quadrats were kept for the test set.

2.2 High resolution patches

High resolution patches are composed of 6 layers of 256 by 256 pixels at a resolution of 1 meter per pixel. These layers are 4 layers of aerial view (Red, Green, Blue and Near-IR channels of remote sensing imagery), 1 layer for altitude and 1 layer for land cover. Both altitude and land cover had a lower raw resolution and have been re-sampled. For the remote sensing imagery, US raw data where already at 1m/px and french ones where down-sampled from 0.5m/px to 1m/px. Patches where extracted for each train and test occurrences and directly accessible for participants. Land cover is a categorical layer with 34 categories, for practical use it must be unstack to 34 binary layers each coding the presence or absence of the corresponding category. A dataset code was also given to facilitate finding and reading patches. This code also include the possibility to unstack the land cover categorical layer. More details about these patches are given in [1].

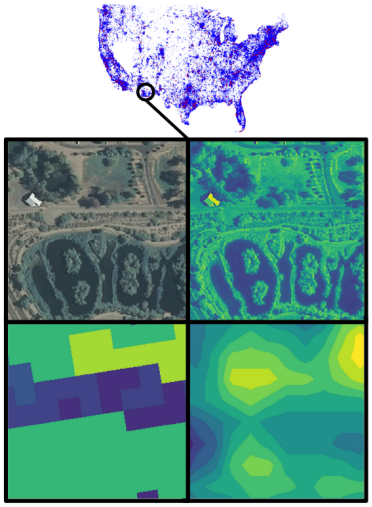

Fig. 2: Caption

2.3 Environmental rasters

Twenty seven environmental rasters were also provided to the participants. This dataset combined 19 bioclimatic rasters from WorldClim and 8 pedologic rasters from SoilGrids covering France and USA. Their spatial resolution is approximately of 1 kilometer. This dataset allows to learn more classical environmental models in order to compare them with deep learning approaches. An extraction code was provided to the participants at `https://github.com/maximiliense/GLC`.

2.4 Metric

The main metric used to evaluate runs of participants, is an adaptive top-30 (although a classical top-30 is also displayed for information to the participants).

The adaptive top-30 is constructed as follows. First, let us note

$$\mathcal{P} = \{s_i^{(1)}, ..., s_i^{(K)}\}_{i \leq N}$$

the set of N predictions over K classes. The first step of the adaptive top-30 is the definition of a threshold t such that:

$$t = \arg\min_{\kappa \in \mathbb{R}} \kappa, \;\; s.t. \;\; \frac{1}{N} \sum_i \sum_j \delta_{s_i^{(j)} > \kappa} \leq 30$$

where $\delta_{s_i^{(j)} > t}$ equals 1 if the inequality is verified and 0 otherwise. In other words, t defines the smallest threshold such that in average, at most 30 classes have a score above t.

Finally, the score is computed as:

$$\frac{1}{N}\sum_{i}\delta_{s_i^{(y_i)}>t}.$$

Such a metric permits to adapt the number of species to consider at a given location depending on the confidence returned by the model.

3 Implemented Methods

For reproducibility purposes, we share a repository containing the trained models and their parameters, available at the following web address : `https://gitlab.inria.fr/bdeneu/glc20-participation`.

3.1 Run 1 - Random forest trained on environmental feature vectors

This method was used for the runs entitled *LIRMM / Inria Run 1* in Figure 1. This model has been trained with environmental rasters (bio_1-bio_19, orcdrc, phihox, cecsol, bdticm, clyppt, sltppt, sndppt, bldfie). For each occurrence an environmental vector is extracted from the rasters at the occurrence location (vector of size 27). The model was learnt on all occurrences (France and US) keeping 0.5% for validation (note that the final model has not been re-trained on these occurrences). In addition, few occurrences ended up outside of our environmental rasters and were put aside. The implementation used is the random forest classifier from scikit-learn [10] with a number of trees (n_estimators) of 100 and a maximum depth of 10 (all other parameters are kept at the default value).

3.2 Runs 2/3 - Convolution neural networks trained on high-resolution image covariates

The method described here was used for the runs entitled *LIRMM / Inria Run 3* and *LIRMM / Inria Run 2* in Figure 1. Both runs correspond to the same model (architecture and training). They differ only by the prediction procedure on the test set. The model has been trained on the following patches: red imagery, green imagery, blue imagery, near-infrared imagery, land cover and altitude. Thus, all patches have shape $(c \times w \times h) = 6 \times 256 \times 256$ where c stands for channel.

The 1.9 million occurrences have been grouped together in order to train the model on all samples. In addition, the occurrences have been split in three sets: train, validation and test. Train occurrences represent 90% of all occurrences, validation ones represent 5% of the total and tests occurrences represent 5% of the total as well. The validation set is used to select the best model while the test set is used once at the end to have an estimate of its performances. The split is done completely randomly, unlike the challenge test occurrences which split used spatial information.

The model used in both runs is an Inception V3 [9] that has been a little bit customized. Lower layers have been adapted for our tensor shape. In particular, the first two layers have been constructed as follows in `Pytorch`[6]:

```
self.Conv2d_1a_3x3 = BasicConv2d(n_input, 32, kernel_size=3,
                                 stride=1, padding=1)
self.Conv2d_2a_3x3 = BasicConv2d(32, 32, kernel_size=3,
                                 stride=1, padding=1)
```

In addition, Inception v3 models typically have an auxiliary output that we have not used for simplicity reasons. In more details, the training procedure is as follows:

- Batch size: 128,
- Dropout: 0.5,
- Learning rate: 0.1,
- 180 epochs with a learning rate decay ($\gamma = 0.1$) at $90, 130, 150, 170, 180$,
- One validation every 5 epochs where top-1, top-10 and top-30 accuracy are evaluated.

The representation layer has width 2048 and is followed by a softmax:

$$p_i = \exp(p_i)/(\sum_k \exp(p_k)).$$

Additionally the loss is the cross-entropy, also known as the negative log-likelihood when interpreting the output of the softmax layer as probabilities:

$$\ell(y, p) = -\sum_k y_k \log(p_k),$$

where y is the one-hot encoding of the sample label and p the output of the model interpreted as a probability. The model has been trained on a machine with 190GB of memory and 8 GPUs V100 with 32GB. Due to a lack of time, the model has not been trained on 100% of the occurrences before predicting on the test set. Run 2 was used with dropout during its prediction, whereas run 3 was used in prediction mode, without dropout. The latter obtained the best score in the challenge.

3.3 Run 4 - Fusion

This method was used for the run entitled *LIRMM / Inria Run 4* in Figure 1. The late fusion is based on the prediction of the CNN and of the random forest on the test set. Predictions normalized as probabilities of both model have been averaged. Then predicted species of each occurrence have been reranked.

Unfortunately a bug during the decompression of US patches has affected this run leading to a degraded prediction of the CNN using them. This late fusion is in consequence not representative of the potential maximum score of this method.

[6] `http://pytorch.org`

4 Results analysis

As expected, CNNs beat the random forest which confirm previous results comparing these two methods [7, 5]. Here the comparison went further by not comparing the two methods on the same data. Random forest has been learned on an environmental vector as it is the case in most classical SDM approaches while the neural network was learned on high-resolution tensors at 1m per pixel containing the aerial views in R, G, B and near-IR, altitude and land cover. The neural network does not use environmental raster data and therefore does not use explicit environmental data. This result suggests that convolutional neural networks are capable of extracting rich ecological information from high-resolution aerial view data. This result is particularly interesting from the point of view of producing high spatial resolution SDMs. The comparison of the two models can also be made from a practical point of view, as both have advantages and disadvantages. Firstly, as the neural network uses large dimensional data with fine resolution, the volume of data to be manipulated is very large (near 1TB). This can be a blocking point depending on the volume of computational resources available. Moreover, learning the model on such a volume of data requires access to large computing resources over a long period of time (several weeks). In contrast, random forests take less than 15 minutes to be learnt and require a smaller amount of resources. The limiting point is the amount of RAM required (in this case near 50GB). If we now look at the learned model, the neural network is much lighter than the random forest (656MB vs. 41GB). This is an important positive point for the CNN because, combined with its other advantages such as high spatial resolution and the possibility of transfer learning, it makes it a lightweight, reusable model with a fine resolution high predictive power. It is important to note, however, that methods using environmental rasters (available on a territorial scale) such as RF allow predictions to be made at any point by easily extracting the environmental vector. This is much more complicated for the CNN, for which it is necessary to first extract the tensors (especially aerial views) at the point where the prediction is to be made. This extraction is time-consuming as it is very difficult to quickly access this information over the entire territory (the volume of data being far too large).

Regarding the late fusion, the idea was to merge a more classical environmental model (the RF) with the CNN at a fine resolution to study the complementary of methods. Unfortunately the extraction of the US patches from the archives on the machine where was performed the late fusion was corrupted, resulting on near half of the patches (the majority of the US ones) being damaged. We did not have time to re-download and extract the patches in the allocated time. The results of this fusion are therefore not exploitable as such. We can however note that the performance, despite the number of patches concerned, remains high and close to the CNN (above the RF) which would seem to indicate probable gain in the case where there would not have been these problems.

Rank	RunId	RunName	top30	AdaptiveTop30	Participant
1	67895	Run 3	0.2346	0.2333	LIRMM/Inria
2	68516	Run 4	0.2260	0.2290	LIRMM/Inria
3	68222	Run 1	0.2043	0.2130	LIRMM/Inria
4	67853	Run 2	0.2115	0.1549	LIRMM/Inria
5	67853	Run 3	0.0482	0.0482	Stanford
6	67853	Run 4	0.0480	0.0480	Stanford
7	67853	Run 2	0.0462	0.0462	Stanford
8	67853	Run 1	0.0439	0.0439	Stanford
9	67853	Run 5	0.0435	0.0435	Stanford

Table 1: Submissions

5 Conclusion and future works

Our results show that the method achieving the best prediction of species in the context of the GeoLifeCLEF 2020 challenge was a convolutional neural network trained solely on the high-resolution covariates (RGB-IR imagery, land cover, and altitude). It did outperform the more classical species distribution modelling approach based solely on punctual environmental variables at a coarse resolution. This suggests two things: (i) important information explaining the species composition is contained in the high-resolution covariates and (ii), convolutional neural networks are able to capture this information. An important following question would be to know whether the information captured by the high-resolution CNN is complementary to the one captured from the bioclimatic and soil variables. This was the purpose of one of the method we implemented but unfortunately was not really conclusive because of a corruption in the data.

6 Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 863463 (Cos4Cloud project), the support of #DigitAG.

References

[1] Elijah Cole, Benjamin Deneu, Titouan Lorieul, Maximilien Servajean, Christophe Botella, Dan Morris, Nebojsa Jojic, Pierre Bonnet, and Alexis Joly. "The GeoLifeCLEF 2020 Dataset". In: *arXiv preprint arXiv:2004.04192* (2020).

[2] Benjamin Deneu, Titouan Lorieul, Elijah Cole, Maximilien Servajean, Christophe Botella, Dan Morris, Nebojsa Jojic, Pierre Bonnet, and Alexis Joly. "Overview of LifeCLEF location-based species prediction task 2020 (GeoLifeCLEF)". In: *CLEF task overview 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece.* 2020.

[3] Alexis Joly, Hervé Goëau, Stefan Kahl, Benjamin Deneu, Maximilien Servajean, Elijah Cole, Lukáš Picek, Rafael Ruiz De Castañeda, é, Titouan Lorieul, Christophe Botella, Hervé Glotin, Julien Champ, Willem-Pier Vellinga, Fabian-Robert Stöter, Andrew Dorso, Pierre Bonnet, Ivan Eggel, and Henning Müller. "Overview of LifeCLEF 2020: a System-oriented Evaluation of Automated Species Identification and Species Distribution Prediction". In: *Proceedings of CLEF 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece.* 2020.

[4] Christophe Botella, Maximilien Servajean, Pierre Bonnet, and Alexis Joly. "Overview of GeoLifeCLEF 2019: plant species prediction using environment and animal occurrences". In: *CLEF task overview 2019, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2019, Lugano, Switzerland.* (2019).

[5] Mathilde Negri, Maximilien Servajean, Benjamin Deneu, and Alexis Joly. "Location-Based Plant Species Prediction Using A CNN Model Trained On Several Kingdoms-Best Method Of GeoLifeCLEF 2019 Challenge". In: *CLEF working notes 2019, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2019, Lugano, Switzerland.* 2019.

[6] Christophe Botella, Pierre Bonnet, Francois Munoz, Pascal Monestiez, and Alexis Joly. "Overview of GeoLifeCLEF 2018: location-based species recommendation". In: *CLEF task overview 2018, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2018, Avignon, France.* (2018).

[7] Christophe Botella, Alexis Joly, Pierre Bonnet, Pascal Monestiez, and François Munoz. "A deep learning approach to species distribution modelling". In: *Multimedia Tools and Applications for Environmental & Biodiversity Informatics*. Springer, 2018, pp. 169–199.

[8] Antoine Affouard, Jean-Christophe Lombardo, Herve Goeau, Pierre Bonnet, and Alexis Joly. "Pl@ntnet app in the era of deep learning". In: *ICLR 2017 Workshop Track-5th International Conference on Learning Representations* (2017).

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python ". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.