

Chapter 9

Automated identification of herbarium specimens at different taxonomic levels

Jose Carranza-Rojas, Alexis Joly, Herve Goeau, Erick Mata-Montero, and Pierre Bonnet

Abstract The estimated number of flowering plant species on Earth is around 400,000. In order to classify all known species via automated image-based approaches, current datasets of plant images will have to become considerably larger. To achieve this, some authors have explored the possibility of using herbarium sheet images. As the plant datasets grow and start reaching the tens of thousands of classes, unbalanced datasets become a hard problem. This causes models to be inaccurate for certain species due to intra- and inter-specific similarities. Additionally, automatic plant identification is intrinsically hierarchical. In order to tackle this problem of unbalanced datasets, we need ways to classify and calculate the loss of the model by taking into account the taxonomy, for example, by grouping species at higher taxon levels. In this research we compare several architectures for automatic plant identification, taking into account the plant taxonomy to classify not only at the species level, but also at higher levels, such as genus and family.

Jose Carranza-Rojas
Costa Rica Institute of Technology, Cartago, Costa Rica, e-mail: jcarranza@itcr.ac.cr,

Alexis Joly,
Inria ZENITH team, France, e-mail: alexis.joly@inria.fr,

Hervé Goëau
CIRAD, UMR AMAP, F-34398 Montpellier, France — AMAP, Univ Montpellier, CIRAD, CNRS,
INRA, IRD, Montpellier, France, e-mail: herve.goeau@cirad.fr,

Erick Mata-Montero
Costa Rica Institute of Technology, Cartago, Costa Rica, e-mail: emata@itcr.ac.cr,

Pierre Bonnet
CIRAD, UMR AMAP, F-34398 Montpellier, France — AMAP, Univ Montpellier, CIRAD, CNRS,
INRA, IRD, Montpellier, France, e-mail: pierre.bonnet@cirad.fr

9.1 Introduction

In general, Deep Learning classification has focused mostly on flat classification, i.e., hierarchies and knowledge associated with higher levels are usually not taken into account. However, in the biological domain, the approach traditionally followed by taxonomists is intrinsically hierarchical. Single-access and multiple-access identification keys are an example of such an approach [3]. They are used to identify organisms mostly at the species level but sometimes at the genus and family levels too. To our knowledge, most of the research on image-based automated plant identifications classify plant images into species and do not exploit knowledge about other taxonomic levels.

Very few studies also have attempted to use herbarium images for plant identification. With new deep learning methods, large datasets of herbarium images such as those published by iDigBio¹ [1] [2], which comprises millions of images of thousands of species from around the globe, become very useful. These datasets are suitable for deep learning approaches and include as metadata all levels of the associated taxonomy. In [4] a GoogleNet model with modifications is used to classify species from the Mediterranean region and Costa Rica. It shows promising results in terms of accuracy when training and testing with herbarium sheet images, as well as when doing transfer learning from the Mediterranean region to Costa Rica. However, classifications are conducted only at the species level and do not use additional knowledge related to other taxonomic levels.

Herbaria normally hold many samples that have not been identified at the species level [5] but they make an effort to at least have them identified at the genus or family level. It is therefore important to help streamline the identification process with tools that support identifications at multiple levels (probably with different levels of accuracy).

One of the biggest issues in plant identification is the lack of balanced datasets. At the species level, most available datasets are unbalanced due to taxonomically uneven nature of sample collection processes [3]. So, an expected intuition in this domain is to exploit higher levels of the taxonomy in order to have more images of a single class and use that knowledge to help the classification at lower levels of the taxonomy, such as the species at the bottom. In other words, the unbalanced dataset issue could be tackled by using a class hierarchy and doing classifications from top to bottom.

In this work we compare several deep learning architectures to do herbarium specimen identification at not only species level, but also other taxonomic levels such as genus and family. We explore architectures that do several taxonomic level classifications at the same time by sharing parameters, as well as separated flat classifiers, independent from each other.

The rest of this manuscript is organized as follows: Section 9.2 presents relevant related work. Section 9.3 and Section 9.4 cover methodological aspects and exper-

¹ <https://www.idigbio.org/>

iment design, respectively. Section 9.5 describes the results obtained. Section 9.6 presents the conclusions and, finally, Section 9.7 summarizes future work.

9.2 Related Work

PlantCLEF is the largest and best known plant identification challenge [6]. It has helped to create bigger datasets each year as well as allowed participants to gradually improve the techniques (mostly deep learning based models) to achieve better accuracy. So far, PlantCLEF has focused on species level identifications only.

The same situation happens with apps for automated image-based plant identification such as LeafSnap [7] and Pl@ntNet [8]. These apps are also focused on classification only at the species level; however, it would be useful in cases where the accuracy is low, to have predictions at other taxonomic levels such as genus and family.

Very few studies have tackled the problem of hierarchical classification. Silla et al. [9] present a very comprehensive survey about different techniques used for hierarchical classification. Wu et al. [10] discuss how there are no even proper standards to evaluate hierarchical classification systems, and use Naive Bayes approach on text data. Both studies are focused on traditional machine learning, not deep learning.

Shahbaba et al. [11] create a new method using a Bayesian form of the softmax function, adding a prior that introduces correlations between the parameters of nearby classes of the hierarchy. This approach was also developed for traditional machine learning and not deep learning. Nevertheless it could be easily adjustable to deep learning. This approach is also useful when there is a prior knowledge of the class hierarchy.

Yan et al. [12] create a new architecture named Hierarchical Deep CNN (HD-CNN), which uses 2 levels of classification. The first level is more general and then the second level is composed by several smaller classifiers per each class. This means that the amount of classifiers grows after the first classification. Furthermore, an error during the first classification will lead to error propagation to the second layer of classifiers.

There have been a lot of studies where the hierarchy is learned via unsupervised learning. In this paper, we focus on an already defined hierarchy which is a plant taxonomy. It is the result of decades if not centuries of work in the field of taxonomy. Based on that result, we don't calculate automatically the class hierarchy.

In particular, to our knowledge, no plant identification system or study has been proposed that actually exploits the class hierarchy using the plant taxonomy. In [13] authors do analyze the accuracy per species but also per genera and families, to see which species are better identified and also to evaluate if the amount of images per species has a direct impact over the accuracy obtained per class. They conclude that species with around 100 images are very well identified with some exceptions. Indeed, a few species are very well identified even with a very small number of images. They also provide accuracy per genus and family.

Additionally, ImageNet weights were used to pre-train the deep learning model. We only used the weights of a pre-trained model on ImageNet, not the dataset itself. Details of the used datasets are presented below:

- Herbaria1K (H1K): this dataset covers 1,191 species, 918 of which are included in the 1,000 species of the PlantCLEF dataset from 2015 [14]. Obtained through iDigBio, the dataset contains 202,445 images for training and 51,288 images for testing. All images have been resized to a width of 1,024 pixels and their height proportionally, given the huge resolutions used in herbarium images. This is an unbalanced dataset, as explained in next sections of this manuscript. In terms of genera, it contains a total of 498 genera, and regarding families it has a total of 124 families.
- ImageNet is the most widely used dataset by the machine learning research community. It contains 1,000 generalist classes and more than a million images [15]. It is the *de facto* standard for pre-training deep learning models. We use only the weights of a trained model with this dataset for transfer learning proposes.

Table 9.1 Datasets used in this research

Name	Acronym	Source	Type	# of Images	# of Species/Classes
Herbarium1K	H1K	iDigBio	Herbarium Sheets	253,733	1,191
ImageNet	ImageNet (I)	ImageNet Challenge	Generic Images	1M	1000

9.3.2 Unbalanced dataset

Figure 9.2 shows how unbalanced the H1K dataset is. According to the work in [13], the H1K dataset allows high identification rates with their deep learning model after 100 images per species. As shown in the figure, around 60% of the species have more than 100 images, and 40% less than that. Some species have lots of images, for example 324 species have more than 300 images, but in contrast, 311 species have less than 11 images in total for both training and testing.

9.3.3 Architectures

The basis architecture used in our study is an extension of the GoogleNet architecture with batch norm [16], as used in [4] for plant identification on herbarium specimens. The main difference is at the last fully connected layer. Table 9.2 shows the modified GoogleNet network used in this research, taken from [4]. The network was implemented in Lasagne [17], using Theano [18].

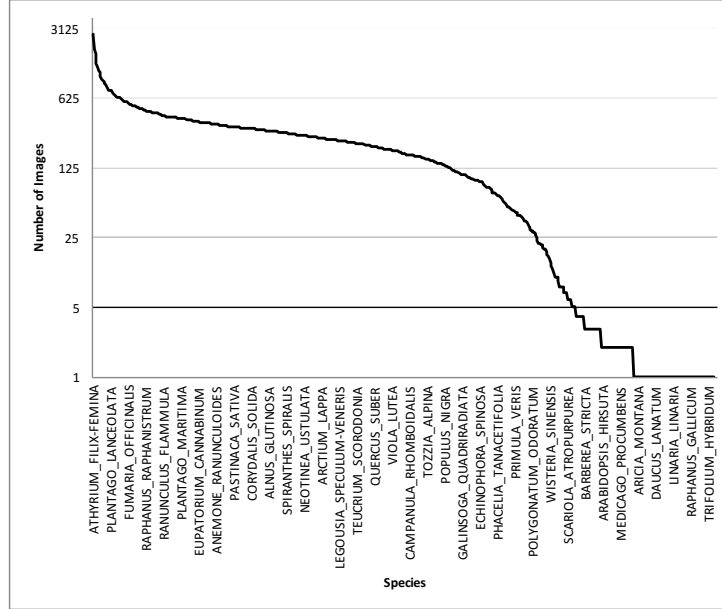


Fig. 9.2 Image per class distribution showing the unbalanced H1K dataset

Table 9.2 GoogleNet architecture modified with Batch Normalization, taken from [4]

Type	Patch size / Stride	Output Size	Depth	Params	Ops
convolution	7x7/2	112x112x64	1	2.7K	34M
max pool	3x3/2	56x56x64	0		
batch norm		56x56x64	0		
LRN		56x56x64	0		
convolution	3x3/1	56x56x192	2	112K	360M
max pool	3x3/2	28x28x192	0		
batch norm		28x28x192	0		
LRN		28x28x192	0		
inception (3a)		28x28x256	2	159K	128M
inception (3b)		28x28x480	2	380K	304M
max pool	3x3/2	14x14x480	0		
batch norm		14x14x480	0		
inception (4a)		14x14x512	2	364K	73M
inception (4b)		14x14x512	2	437K	88M
inception (4c)		14x14x512	2	463K	100M
inception (4d)		14x14x528	2	580K	119M
inception (4e)		14x14x832	2	840K	170M
max pool	3x3/2	7x7x832	0		
batch norm		7x7x832	0		
inception (5a)		7x7x832	2	1072K	54M
inception (5b)		7x7x1024	2	1388K	71M
avg pool	7x7/1	1x1x1024	0		
batch norm		1x1x1024	0		
linear		1x1x10000	1	1000K	1M
softmax		1x1x10000	0		

9.3.3.1 Baseline: Flat Classification Model (FCM)

In order to evaluate the performance of adding hierarchies to the architecture classification, a first base line is set based on a FCM. Since we are classifying not only species but also genera and families, the flat approach requires 3 different instances of the same model, with different number of outputs on the last dense layer and softmax, according to the dataset label size for each taxonomic level. Figure 9.3 shows the 3 main building blocks that will be used on the next sections with information about the models. For species we have a total of 1191 outputs, for genera 498 and for families 124. These output sizes are the same across all architectures.

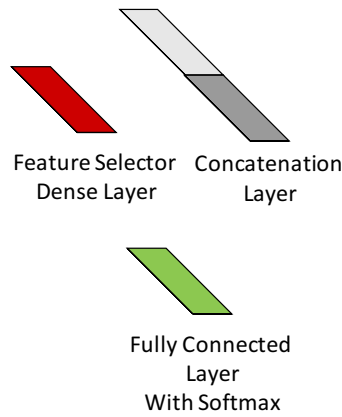


Fig. 9.3 Representation of some building blocks of the different architectures.

Figure 9.4 shows how the flat model looks like. The model is basically a GoogleNet [19] model, modified with Parametric REctified Linear Unit (PRELU) and batch normalization for faster convergence. A total of 3 different flat models were deployed: one for species, one for genera, and one for families. The 3 models are completely independent and do not share any parameters. They also have their own training and parameter update process.

9.3.3.2 Multi-Task Classification Model (MCM)

Another approach to calculate accuracy at different taxonomic levels is by using a model where the different classifiers share the same deep network. Multi-Task Classification Model (MCM) implements one classifier per taxonomic level, in this case 3 classifiers, one for species, one for genera and one for families. However, each classifier is connected to the last pooling layer of the GoogleNet model, allowing to do 3 classifications at the same time and sharing the same parameters of the model

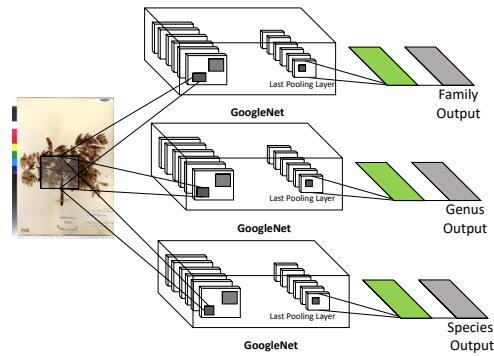


Fig. 9.4 Separated Flat Classification Model (FCM) for species, genera and family

instead of having 3 separate models with their own parameters. The intuition behind is that the network will learn features from the 3 taxonomic levels at the same time. Figure 9.5 shows how a single main GoogleNet model is shared between 3 different classifiers, each one assigned to classifying a different taxonomic level. This model is inspired by the work of [20], where the authors identify multi-digit numbers from houses, using a classifier per digit.

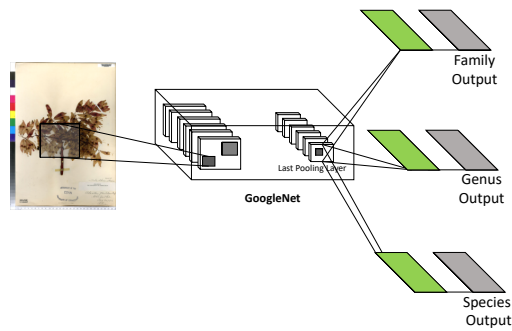


Fig. 9.5 A Multi-Task Classification Model (MCM) for species, genera and family. Parameters are shared between the 3 taxonomic levels, similar to the work of [20] for multi-digit identification of house numbers.

9.3.3.3 TaxonNet: Hierarchical Classification

We present the following architecture that attempts to capture features at several levels of the plant taxonomy. We call this architecture TaxonNet, as it takes into account several levels of the plant taxonomy as the hierarchy, and uses knowledge of the previous taxonomic level classification for the next one, as shown in Figure 9.6.

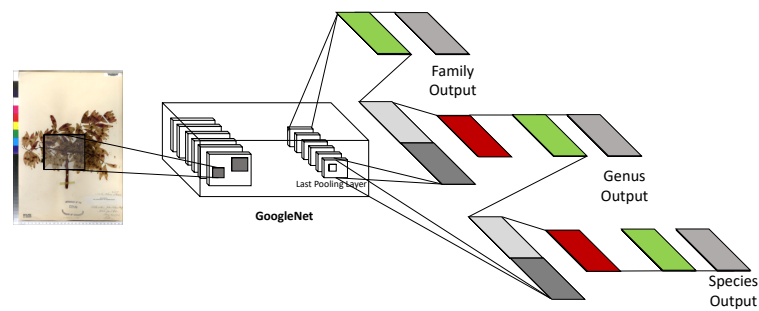


Fig. 9.6 TaxonNet used to identify species, genera and family. The architecture allows to take into account previous classification of another taxonomic level for the next one

We modified the GoogleNet model in the following fashion: the last fully connected layer which was used normally for a flat species classification is now used for the higher taxonomic level, in this case the Family. The loss of this fully connected layer will be calculated based on family labels of each image. Just before the softmax, the feature vector of the family fully connected layer output is concatenated with the last pooling layer feature vector. The idea behind this, is to add a new fully connected layer for the genus, which will base its computations on both the family fully connected feature vector, and the raw feature vector coming from the Convolutional Neural Network (CNN). In other words, the features learned to recognize the family will be used as a context to learn new complementary features at the genus level. Finally, we apply the same concept with the species: we add a new fully connected layer for species, which takes as input the concatenation of the genus fully connected layer output plus the last pooling layer feature vector from the CNN. In all cases, there is a middle feature selector layer in red, as shown in Figure 9.3, which allows the model to learn which features to take into account either from the original GoogleNet or from the previous taxonomic classification.

It is important to notice that the design allows the model to make mistakes at higher levels of the taxonomy, such as family or genus, but it can have good accuracy at the species level since it handles the raw feature vectors coming from the CNN. In other words, an error at higher levels of the taxonomy won't necessarily cause an error propagation to lower levels. It also allows to do classification at all taxonomic levels, thus, each one has its own loss which is back-propagated to the whole network. Our intuition is that the whole network learns features at all taxonomic levels, instead of having several complete CNN for each level. This allows to share parameters between levels, for a smaller network.

9.4 Experiments

By using the previous explained models we ran several experiments to measure the effect of taking into account different taxonomic levels for the classification.

In all cases the used learning rate was 0.0075 and weight decay of 0.0002. The total number of training iterations was 6300 with a mini-batch size of 32 images, with 5 epochs. The number of validation iterations was 1500, the same as for testing iterations.

9.4.1 Baseline Experiments: Flat Classification Model (FCM)

The first experiments are based on running the separated models for species, genus and family without sharing any type of parameters. This is considered as the baseline, as there are no hierarchical characteristics at all, but just 3 models completely independent from each other.

9.4.2 Architecture Comparison Experiment

This experiment consists on a comparison of the different architectures at the different taxonomic levels. The experiment compares the MCM approach, where parameters are shared between the different classifiers, with the intention to see how the accuracy and loss behaves as the model is trained, compared to separated model per taxonomic level. The TaxonNet architecture is also compared with the separated models and the MCM approach.

9.5 Results

9.5.1 FCM Baseline Results

First experiments consisted on running 3 separated FCM models to explore the loss and accuracy behavior at each taxonomic level. We consider this as the baseline results, as they are flat classifiers that do not share any hierarchical characteristics in terms of classification.

The results for all the FCM are shown in Figures 9.7, 9.8 and 9.9. In particular, FCM for species gets Top-1 63.02% and Top-5 is about 82.93% as shown in Figure 9.7. In case of the FCM for genus the accuracy goes up to Top-1 of 70.51% and Top-5 of 87.85%, as shown by Figure 9.8. For the family, Figure 9.9 shows the best results for both Top-1 and Top-5, with 75.55% and 93.43% respectively.

It is important to notice that both genus and family show an improvement compared to the species. This makes sense as genus and family have more images per class and also both models have less classes, 498 for genus and 124 for families. This compensates the difficulty of having a higher intra-class variability at that levels.

9.5.2 Architecture Comparison Results

9.5.2.1 MCM Top-1 and Top-5 behavior

In case of the MCM architecture, for Top-1 accuracy the results show 64.32% for species, 75.95% for genus, and for family 88.17%, as shown by Figure 9.10. The parameter sharing allows the model to predict the family with a very high accuracy. In case of Top-5 accuracy, MCM results in 71.66% for species, 83.23% for genus, and 92.99% for family. Family classification is again the best among the 3, as shown by Figure 9.11.

9.5.2.2 TaxonNet Top-1 and Top-5 behavior

In Figure 9.12, TaxonNet architecture shows for Top-1 accuracy 62.39%, 76.23%, 86.92% for species, genus and family, respectively. Again, similarly to MCM, the parameter sharing allows the model to predict the genus and family with a very high accuracy. For Top-5 accuracy, as shown by Figure 9.13, TaxonNet results in 70.20%, 82.36% and 92.80% for species, genus and family, respectively, again being the family classification the best among the 3.

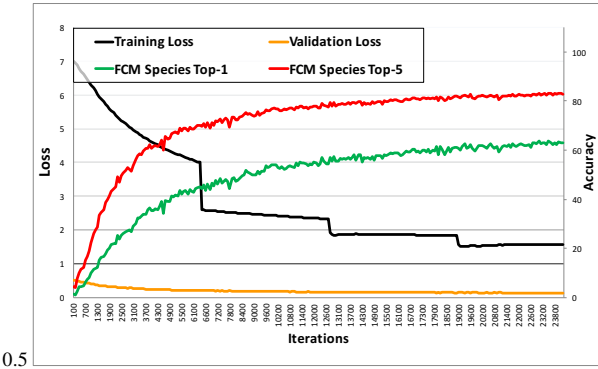


Fig. 9.7 FCM for species showing Top-1 and Top-5 accuracy and losses.

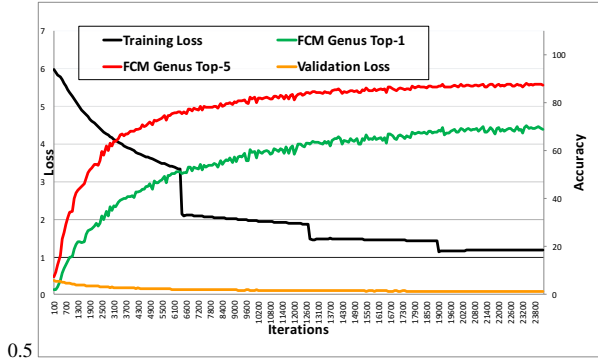


Fig. 9.8 FCM for genera showing Top-1 and Top-5 accuracy and losses.

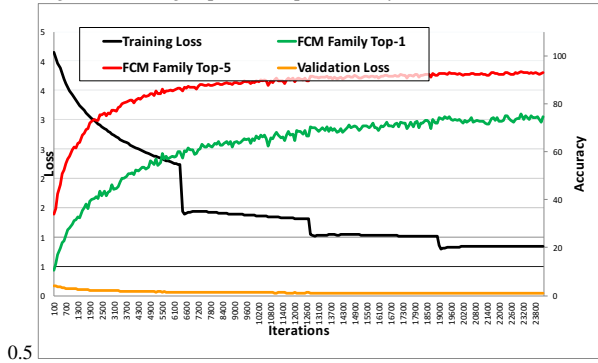


Fig. 9.9 FCM for family showing Top-1 and Top-5 accuracy and losses.

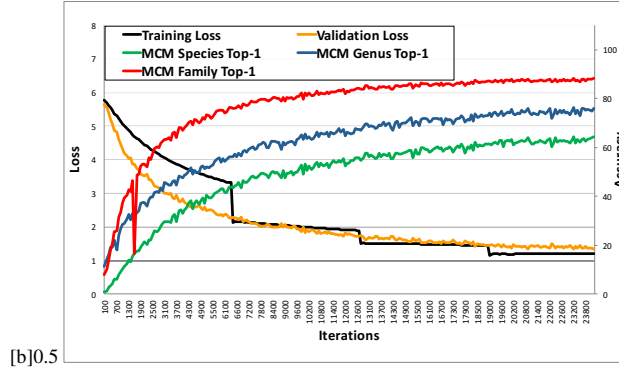


Fig. 9.10 MCM results for species, genus and family on Top-1 accuracy

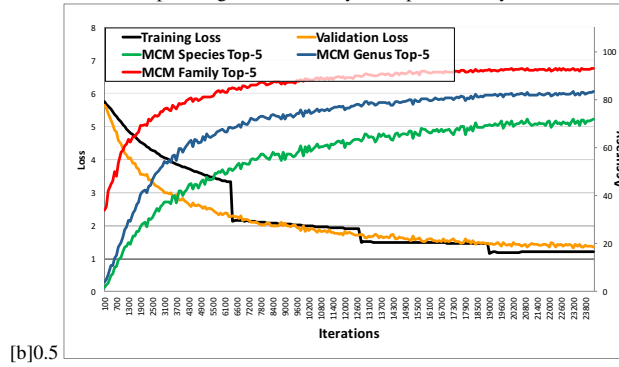


Fig. 9.11 MCM results for species, genus and family on Top-5 accuracy

9.5.2.3 Architecture Comparisons

Our results demonstrate that for species, Top-1 accuracy is 63.02%, 64.32%, 62.39% for FCM, MCM and TaxonNet, respectively, showing the best results on the MCM architecture by a margin of 1% approximately.

Regarding genus, our result shows a Top-1 accuracy is 70.51%, 75.95%, 76.23% for FCM, MCM and TaxonNet, respectively. In this case, the degradation of the flat classifier for the genus is improved significantly by both hierarchical architectures, with the TaxonNet being the best one.

Finally, for family, our result shows a Top-1 accuracy is 75.55%, 88.17%, and 86.92% for FCM, MCM and TaxonNet. Here the improvement by both hierarchical architectures is very important compared to the flat classifier on Top-1.

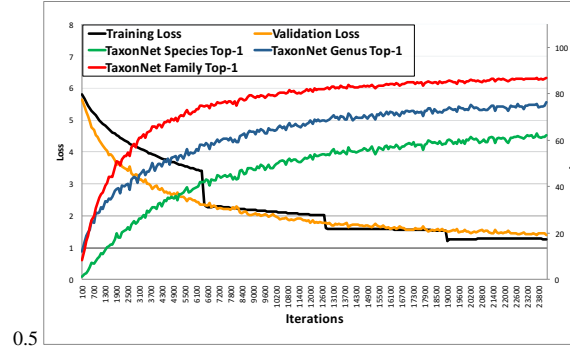


Fig. 9.12 TaxonNet results for species, genus and family on Top-1 accuracy

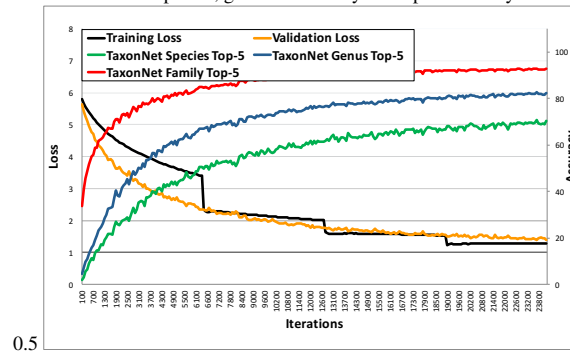


Fig. 9.13 TaxonNet results for species, genus and family on Top-5 accuracy

9.6 Conclusions

The best accuracy results for species and genus are provided by the independent Flat Classification Model (FCM), but at the cost of 3 times more GPU consumption as well as 3 times more parameters. In case of the family, both the Multi-Task Classification Model (MCM) and TaxonNet architectures provide similar results to the flat model.

In general, whatever architecture is used, the classification accuracy increases significantly with the taxonomic level and reaches high classification accuracy at the family level whereas such groups are very heterogeneous visually as they include a lot of species.

The goal of this research was to study several architectures for automatic plant identification, taking into account the plant taxonomy to classify not only at the species level, but also at higher levels, in particular genus and family. In this regard, we introduced two architectures on top of the GoogleNet basis convolutional neural

network: one multi-task classifier dedicated to do 3 classifications at the same time, and one hierarchical architecture (called TaxonNet) aimed at capturing features at several levels of the plant taxonomy. Our experiments did show that the multi-task network as well as the hierarchical one allow considerable improvements compared to having separate flat classifiers, in particular for predicting the genus and the family.

9.7 Future Work

This work uses knowledge of higher levels of taxonomy for species classification, and allows to classify at higher levels of the taxonomy such as genus and family. However, it uses traditional fully connected layers with traditional cross entropy loss and softmax calculations. Next steps include exploiting the class hierarchy to calculate a different loss functions using Bayesian approaches of hierarchical softmax functions. Furthermore, hierarchical regularization terms could be defined to regularize the loss calculation using the class hierarchy. Interesting future experiments include understanding how using the taxonomy impacts the classification of new, unseen classes, at higher taxon levels. For instance, a species may not have been included during training but the genus related to that species may be, thus, allowing the system to provide an identification at that level could be of a strong interest. Additional architectures are also needed to be explored such as Long Short-Term Memory (LSTM) based architectures for the taxonomy.

Acknowledgements Special thanks to the Colaboratorio Nacional de Computacin Avanzada (CCNA) in Costa Rica for sharing their Tesla K40-based cluster and providing technical support for this research. We also thank the Costa Rica Institute of Technology for the financial support for this research.

References

1. Matsunaga, A., Thompson, A., Figueiredo, R. J., Germain-Aubrey, C. C., Collins, M., Beaman, R. S., ... Fortes, J. A. (2013, October). A computational-and storage-cloud for integration of biodiversity collections. In *eScience (eScience)*, 2013 IEEE 9th International Conference on (pp. 78-87). IEEE.
2. Page, L. M., MacFadden, B. J., Fortes, J. A., Soltis, P. S., Riccardi, G. (2015). Digitization of biodiversity collections reveals biggest data on biodiversity. *BioScience*, 65(9), 841-842.
3. E. Mata-Montero and J. Carranza-Rojas, *Automated Plant Species Identification: Challenges and Opportunities*. Springer International Publishing, 2016, pp. 26–36.
4. J. Carranza-Rojas, H. Goeau, P. Bonnet, E. Mata-Montero, and A. Joly, "Going deeper in the automated identification of herbarium specimens," *BMC Evolutionary Biology*, vol. 17, no. 1, p. 181, Aug 2017.
5. D. P. Bebber, M. A. Carine, J. R. Wood, A. H. Wortley, D. J. Harris, G. T. Prance, G. Davidse, J. Paige, T. D. Pennington, N. K. Robson *et al.*, "Herbaria are a major frontier for species

- discovery,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 51, pp. 22 169–22 171, 2010.
6. A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, J. Champ, R. Planqué, S. Palazzo, and H. Müller, *LifeCLEF 2016: Multimedia Life Species Identification Challenges*. Cham: Springer International Publishing, 2016, pp. 286–310.
 7. 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.
 8. A. Joly, P. Bonnet, H. Goëau, J. Barbe, S. Selmi, J. Champ, S. Dufour-Kowalski, A. Affouard, J. Carré, J.-F. Molino *et al.*, “A look inside the pl@ntnet experience,” *Multimedia Systems*, vol. 22, no. 6, pp. 751–766, 2016.
 9. C. N. Silla and A. A. Freitas, “A survey of hierarchical classification across different application domains,” *Data Min Knowl Disc*, vol. 22, pp. 31–72, 2011.
 10. F. Wu, J. Zhang, and V. Honavar, “Learning classifiers using hierarchically structured class taxonomies,” in *Proceedings of the 6th International Conference on Abstraction, Reformulation and Approximation*, ser. SARA’05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 313–320. [Online]. Available: http://dx.doi.org/10.1007/11527862_24
 11. B. Shabbaba and R. M. Neal, “Improving classification when a class hierarchy is available using a hierarchy-based prior,” *Bayesian Anal.*, vol. 2, no. 1, pp. 221–237, 03 2007. [Online]. Available: <http://dx.doi.org/10.1214/07-BA209>
 12. Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu, “Hd-cnn: Hierarchical deep convolutional neural network for large scale visual recognition,” in *ICCV’15: Proc. IEEE 15th International Conf. on Computer Vision*, 2015.
 13. J. Carranza-Rojas, A. A.J.Joly, P. Bonnet, H. H.G.Goau, and E. Mata-Montero, “Automated herbarium specimen identification using deep learning,” *Biodiversity Information Science and Standards*, vol. 1, p. e20302, 2017.
 14. H. Goëau, P. Bonnet, and A. Joly, “LifeCLEF Plant Identification Task 2015,” in *CLEF: Conference and Labs of the Evaluation forum*, ser. CLEF2015 Working notes, CEUR-WS, Ed., vol. 1391, Toulouse, France, Sep. 2015. [Online]. Available: <https://hal.inria.fr/hal-01182795>
 15. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
 16. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: <http://arxiv.org/abs/1502.03167>
 17. S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri *et al.*, “Lasagne: First release.” Aug. 2015. [Online]. Available: <http://dx.doi.org/10.5281/zenodo.27878>
 18. Theano Development Team, “Theano: A Python framework for fast computation of mathematical expressions,” *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: <http://arxiv.org/abs/1605.02688>
 19. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1–9, 2015.
 20. I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, “Multi-digit number recognition from street view imagery using deep convolutional neural networks,” 2014. [Online]. Available: <https://arxiv.org/pdf/1312.6082.pdf>