A comparative study of fine-grained classification methods in the context of the LifeCLEF plant identification challenge 2015

Julien Champ[1,2], Titouan Lorieul[1,2], Maximilien Servajean[1,2], and Alexis Joly[1,2]

[1] Inria ZENITH team, France, name.surname@inria.fr
[2] LIRMM, Montpellier, France

Abstract. This paper describes the participation of Inria to the plant identification task of the LifeCLEF 2015 challenge. The aim of the task was to produce a list of relevant species for a large set of plant observations related to 1000 species of trees, herbs and ferns living in Western Europe. Each plant observation contained several annotated pictures with organ/view tags: Flower, Leaf, Fruit, Stem, Branch, Entire, Scan (exclusively of leaf). To address this challenge, we experimented two popular families of classification techniques, i.e. convolutional neural networks (CNN) on one side and fisher vectors-based discriminant models on the other side. Our results show that the CNN approach achieves much better performance than the fisher vectors. Beyond, we show that the fusion of both techniques, based on a Bayesian inference using the confusion matrix of each classifier, did not improve the results of the CNN alone.

Keywords: LifeCLEF, plant, leaves, leaf, flower, fruit, bark, stem, branch, species, retrieval, images, collection, species identification, citizen-science, fine-grained classification, evaluation, benchmark

1 Introduction

Content-based image retrieval and computer vision approaches are considered as one of the most promising solutions to help bridging the taxonomic gap, as discussed in [5,1,36,34,17]. We therefore see an increasing interest in this trans-disciplinary challenge in the multimedia community (e.g. in [26,10,2,25,20,12]. Beyond the raw identification performances achievable by state-of-the-art computer vision algorithms, recent visual search paradigms actually offer much more efficient and interactive ways of browsing large flora than standard field guides or online web catalogs ([3]). Smartphone applications relying on such image-based identification services are particularly promising for setting-up massive ecological monitoring systems, involving thousands of contributors at a very low cost. A first step in this way has been achieved by the US consortium behind LeafSnap[3], an i-phone application allowing the identification of 184 common american

[3] http://leafsnap.com/

plant species based on pictures of cut leaves on an uniform background (see [23] for more details). Then, the French consortium supporting Pl@ntNet ([17]) went one step beyond by building an interactive image-based plant identification application that is continuously enriched by the members of a social network specialized in botany. Inspired by the principles of citizen sciences and participatory sensing, this project quickly met a large public with more than 300K downloads of the mobile applications ([8,7]). A related initiative is the plant identification evaluation task organized since 2011 in the context of the international evaluation forum CLEF[4] and that is based on the data collected within Pl@ntNet. This paper presents the participation of Inria ZENITH team to the 2015-edition of this challenge [9,19].

2 Related work

From a computer vision and technological perspective, our work is more generally related to *image classification*. Most popular methods for this problem are typically based on the pooling of local visual features into global image representations and the use of powerful classifiers in the resulting high-dimensional embedded space such as linear support vector machines ([24,28]). The Bag-of-word representation (BoW) notably remains a key concept although the raw initial scheme of ([33]) is now outperformed by several alternative new schemes ([24,16,27,6,14]). Its principle is to first train a so called visual vocabulary thanks to an unsupervised clustering algorithm computed on a given training set of local features. The produced partition is then used to quantize the visual features of a given new image into *visual words* that are aggregated within a single high-dimensional histogram. Partial geometry can be embedded in the image representation by using the Spatial Pyramid Matching scheme of ([24]). As it relies on vector quantization, the BoW representation is however affected by quantization errors. Very similar visual features might be split across distinct clusters whereas more dissimilar ones might be affected to the same visual word. This results in both mismatches and potentially irrelevant matches. To alleviate this problem, several improvements have been proposed in the literature. The first one consists in expanding the assignment of a given local feature to its nearest visual words ([16,29,6,14]). This allows reducing the number of mismatches without degrading much the encoding time. Other researchers have investigated alternative ways to avoid the vector quantization step, using sparse coding ([38]) or locality-constrained linear coding ([37]). Such methods optimize the affectation of a given local feature to a small number of visual words thanks to sparsity or locality constraints on the global representation. Another alternative is to use aggregation-based models such as the improved Fisher Vector of [27] or the VLAD encoding scheme ([14]). Such methods do not only encode the number of occurrences of each visual word but also encode additional information about

[4] http://www.clef-initiative.eu/

the distribution of the descriptors by aggregating the component-wise differences. When used with discriminative linear classifiers, such high-dimensional representations benefit of both generative and discrimination approaches leading to state-of-the-art classification performances on fine-grained classification benchmarks ([11]).
A radically different approach to image classification is the use of *deep convolutional neural networks*. Rather than extracting the features according to hand-tuned or psycho-vision oriented filters, such methods directly work on the image signal. The weights learned by the first convolutional layers allows to automatically build relevant image filters whereas the intermediate layers are in charge of pooling these raw responses into high-level visual patterns. The last fully connected layers work more traditionally as any discriminative classifier on the image representation resulting from the previous layers. Deep convolutional neural networks have been recently proved to achieve better results on large-scale image classification datasets such as ImageNet ([22]) and do attract more and more interest in the computer and multimedia vision communities. A known drawback of Deep Convolutional Neural Networks is however that they require a lot of training data mainly because of the huge number of parameters to be learned. Their performances on fine-grained classification are consequently more controversial and they are still often outperformed by local features based approaches, as shown in our experiments. Besides, it is important to notice that they inspire the investigation of new deep learning models making use of more traditional visual features embedding methods (e.g. [31]).

3 Experimented fine-grained image classification systems

We did experiment two families of image classification techniques that are known to provide state-of-the-art classification performances, in particular in fine-grained recognition challenges ([11,18]).

3.1 Convolutional neural networks

Convolutional Neural Networks (CNN) have been mainly used since the 90's for their performances in digit classification. But since a few years, they appear to have now surpassed all state of the art methods for large-scale image classification [22]. In this experimentation, we have used Caffe [15], a Deep Learning Framework, allowing us to use CNN architectures and models from the literature. We have chosen in the Caffe model Zoo the "GoogLeNet GPU implementation" model, based on Google winning architecture in the ImageNet 2014 Challenge [35], and we fine-tuned this model on the LifeCLEF datasets.

The GoogLeNet architecture consists of a 22 layers deep network with a softmax loss as the top classifier. It is composed of three "inception modules" stacked on top of each other. Each intermediate inception module is connected to an auxiliary classifier during training, so as to encourage discrimination in the lower stages of the classifier, increase the gradient signal that gets propagated back, and provide additional regularization. These auxiliary classifiers are only used during the training part, and then discarded.

Experiments Setup The previously described GoogLeNet CNN uses square images as input. For each image in the training and test sets, we therefore cropped the largest square in the center, and re-sized it to 256x256 pixels. Instead of starting to train our CNN from scratch only on plant images, and as it was authorized in this year's challenge, we started with a CNN trained on the popular generalist ImageNet dataset. We only removed its top layers (the fully connected ones), changed the number of outputs, and trained this new model using the desired dataset. As it was implemented within Caffe library, it makes also use of a simple data augmentation technique, consisting in cropping randomly a 224x224 pixels image, and eventually mirroring it horizontally.

During our preliminary experiments, we have tried several training strategies that are presented are presented in Table 1.

Table 1. Various approaches using CNNs.

Name	**# CNNs**	**Data Augmentation**
CNN1	1 CNN with all images	No
CNN2	1 CNN with all images	Yes
CNN3	7 CNNs (1 for each view)	No

We have tested all these configurations using the PlantCLEF 2014 data and groundtruth (500 species, 47815 train images and 13146 test images). CNN1 configuration was the simplest and the first that we have tested, but finally also the one providing the best results. The Data Augmentation method proposed for CNN2 configuration increased significantly the number of train images as we generated 8 new images by applying rotations, and a set of colorimetric transformations with randomized parameters, i.e. brightness & saturation modulation in the HSL color space (multiplier factor randomized between 0.8 and 1.2), and contrast modulation (multiplier factor randomized between 0.7 and 1.3). Even with additional iterations to train the CNN, results remained nearly the same than those for CNN1. The CNN3 configuration consisted in training several CNNs, one for each view type (thanks to the tags provided in the meta-data). On one hand, as some species haven't images for all views, the number of output for each CNN is lower than 1000 and that could help to obtain better results because of the reduction of the confusion risk. On the other hand, some images from a given view (Branch for example) can really help to identify some images tagged with another view (Entire for example). Results were slightly lower for the Branch, Entire, Leaf, Fruit, and Flower views than what was obtained with the standalone CNN. This could be explained by a less important number of images to train the network, and proves that images from a given view can help when identifying an image tagged with another view. This conclusion is not true for the Stem and LeafScan views. The reason is probably that the LeafScan view is specific, very different from other views, and does not contain background information, and as the Stem tag identifies a closeup view of the plant which is not really apparent on other images.

Training parameters As a reminder, here are the most important parameters for Caffe to obtain our submitted run (CNN1). The base learning rate parameter was set to 10^{-5}. The learning rate is divided by 10 every 60k iterations. After 150k iterations the training is over, and the batch size was fixed to 32. All other parameters were unchanged.

3.2 Fisher vectors & Logistic Regression

Fisher vectors (FV) were first introduced in image classification by [27] and proved to be very efficient in fine-grained classification tasks later on ([11]). According to recent surveys such as [13], it is the best performing pooling strategy currently available. We will only recall here the main steps used to extract Fisher vectors, for detailed explanations of the theoretical derivation and for performance analysis we redirect the readers to [30]. The pipeline for computing the Fisher vector describing an image consists in:

1. *Dense extraction of local features*: descriptors, often SIFT descriptors, are extracted on densely sampled overlapping patches at several scales.
2. *PCA transformation*: the descriptors are then de-correlated and compressed using a Principal Component Analysis.
3. *Feature space density estimation*: the distribution of features is modeled as a Gaussian Mixture Model (GMM) that is learned using the popular Expectation-Maximisation (EM) algorithm. We thus obtain a probability distribution of the form of $u(x) = \sum_{k=1}^{K} w_k u_k(x)$ where u_k follows a Gaussian distribution of mean μ_k and covariance matrix Σ_k, $u_k \sim \mathcal{N}(\mu_k, \Sigma_k)$, with Σ_k being diagonal because the features are decorrelated, and where w_k is the weight of the k-th Gaussian, these weights satisfy $\sum_k w_k = 1$.
4. *Encoding and pooling*: the features are encoded and pooled using

$$\mathcal{G}_{\mu_k} = \frac{1}{\sqrt{w_k}} \sum_{i=1}^{N} \gamma_k(x_i) \frac{x_i - \mu_k}{\sigma_k}$$

$$\mathcal{G}_{\sigma_k} = \frac{1}{\sqrt{w_k}} \sum_{i=1}^{N} \frac{\gamma_k(x_i)}{\sqrt{2}} \Big((\frac{x_i - \mu_k}{\sigma_k})^2 - 1 \Big)$$

   where all the divisions and squaring are element-wise operations and where $\gamma_k(x) = \frac{w_k u_k(x)}{\sum_{k'=1}^{K} w_{k'} u_{k'}(x)}$. Theses $2K$ vectors are concatenated to produce the final representation of dimension $2dK$.

5. *Post-processing*: the vectors are L2-normalized and element-wise square-rooted using $x \mapsto sign(x).\sqrt{|x|}$.

Usually, the classification of Fisher Vectors is performed using a linear classifier as it has been shown that using kernelization techniques on such high-dimensional spaces does not improve significantly the performances. In our experiments, we used the Logistic Regression classifier implemented within the LibLinear library ([4]). This method was preferred over Support Vectors Machine because it directly outputs probabilities which then can be used for fusion purposes.

Here, we used two types of Fisher Vectors with two different types of descriptors. The first system was built with RootSIFT descriptors, l2-normalized and square-rooted SIFT descriptors, of 128 dimensions which are then reduced to 80 dimensions through PCA. The second one was based on some complementary descriptors used in the Pl@ntNet application [17]. It consists in the concatenation of several basic descriptors such as Fourier histograms, Edge Orientation Histograms (EOH), HSV histograms and Hough transform histograms. This concatenation was then compressed and de-correlated using PCA. The association of descriptors used depends on the organ, for *Branch*, *Entire*, *Leaf*, *LeafScan*, *Stem* only Fourier, EOH and Hough histograms are used resulting in 44-dimension final descriptors compressed to 14 dimensions after PCA while *Flower* and *Fruit* add HSV histograms giving descriptors of dimension 74 reduced to 38 after compression. In both systems, the GMM used to estimate the probability distribution of the features learns a codebook of 128 words.

4 Fusion methods

Combining multiple classifiers or even multiple results (*i.e.* several images of a single observation) from a single classifier is a way to increase the classification quality. This section presents three main approaches we used to merge the various results from our classifiers.

4.1 Max and Borda

Maximum and Borda Count are two approaches used to merge top-k lists. While the maximum relies on the score of each class with the lists, Borda Count uses their rank.

More precisely, the maximum based approach associates to each class the maximum score it reaches among the different lists. In the Borda Count approach, we have associated each class within a list to a score decreasing while the rank increases. In more details, since we only retrieve the top-K most likely classes, the score of a given species s is computed as follows:

$$score(s) = \sum_{c \in C} K - r_c(s) \tag{1}$$

where $r_c(s)$ is the ranking of species s returned by the classifier c.

4.2 Bayesian inference

Framework presentation This fusion method is inspired by what is done in crowdsourcing multi-labeled classification tasks [21,32]. For this purpose we used the Bayesian inference framework described in Figure 1.

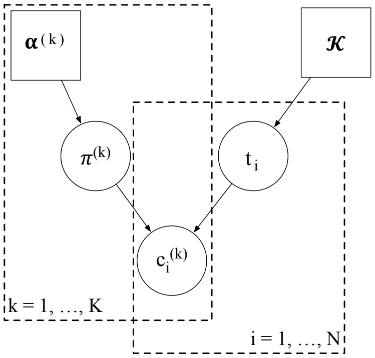

Fig. 1. A Bayesian network to merge multiple classifiers identifications.

In such inference framework, we are given a set of classifiers $k \in 1, ..., K$ and a confusion matrix $\pi^{(k)}$ is assigned to each one of them. Such matrix enables to evaluate the classification quality of each classifier. In a more precise way, $\pi^{(k)}_{i,j}$ refers to the probability that the classifier k, given an image, will answer class j while the right class is i. The set of all confusion matrices is noted $\varPi$. Notice that, as presented in Figure 1, the confusion matrix $\pi^{(k)}$ is directly derived from the parameters matrix $\alpha^{(k)}$. The set of all parameters matrices is noted A. In parallel, each observation (*i.e.* set of images corresponding to a single plant) is associated to a distribution probability, noted t_i for the i^{th} observation. This probability depends on the proportion of each species in the database, and we note κ the vector referring to this proportion. Finally, based on the probabilities t_i and on the confusion matrix of a given classifier k, we can infer the probability of the classifier's answer for the i^{th} observation, noted $c_i^{(k)}$.

Therefore, the joint probability of this Bayesian framework follows Equation 2.

$$p(\varPi, t, c|A, \kappa) = \prod_{i=1}^{N} \{\kappa_{t_i} \prod_{k=1}^{K} \pi^{(k)}_{t_i, c_i^{(k)}}\} p(\varPi|A) \tag{2}$$

Once the classifiers answers (*i.e.* the set of answers $c_i^{(k)}$ for all k and i) are known, the probabilities of $A, \varPi, \kappa$ and t can be updated, thus inferring the correct class of each observation (*i.e.* the one with the highest probability in t_i). In the following, we suppose κ known thanks to the very large size of the training set.

Addressing the large dimensionality Generally, in the state of the art solutions, several approaches are proposed to compute the posterior probabilities such as Gibbs sampling [21] or Variational Bayes [32]. In our experiments we had to face the very large dimension of the problem: each confusion matrix being of size 1000 × 1000. Classical method are therefore intractable in our context. To address this challenge, we used a single-shot approach: only $p(t_i = j|rest)$ is computed and used to update A and π – recall that κ is known and does not need to be updated. Thus, the confusion matrix of each classifier evolves while the number of identifications increases and the quality of inference is refined more and more.

Experiments Setup In this subsection, we present three aspects of the setup: parameters initialization, parameters refinement and classifier's confusion refinement.

An important part of the fusion is to learn the confusion matrix (and its parameters). To do so, we have initialized each parameters matrix A with a value of S in the diagonal and $S/(dimension - 1)$ in the other cells, meaning that there is a 50% probability that the classifier will be correct and that given the correct class and a wrong one, it is more likely that the classifier will return the correct one. In our experiments the value of S has been fixed to 5 (best choice among several runs).

Then, we tried to enhance the confusion matrix quality based on the training data. For each image of the set, we asked the classifiers to re-propose a top-30 classification, and, given the correct class i, we have added in each cell $a_{i,j}$ of the matrices A a value inversely proportional to the species rank in the top-30: $\frac{1}{rank}$.

Finally, to be as fine-grained as possible, each classifier was associated to several confusion matrices corresponding to each plants organs. Thus, the system knows the confusion of each classifier for all possible organs. In a way, we consider each couple $\{organ, classifier\}$ as a single classifier.

5 Official Results

5.1 Runs details

3 runs were finally submitted to the LifeCLEF 2015 plant challenge:

- *INRIA Zenith Run 1* is based on the results provided by the single Convolutionnal Neural Network finetuned using all provided data (CNN1), and described in 3.1. Observations composed of several images, are combined using a Max function to provide Observation Results.
- *INRIA Zenith Run 2* is based on Fisher Vectors described in 3.2. To obtain Observation Results we used the Borda Count Algorithm.
- *INRIA Zenith Run 3* is the combination of the results obtained by previous methods (CNN and Fisher Vectors) using the Bayesian inference method described in 4.2.

5.2 Results

Table 2 summarizes the scores of the 11 best submitted runs out of a total of 18 runs. Figure 2 gives a complementary graphical overview of all results obtained by the participants.

Table 2. PlantCLEF 2015 scores of the 11 best runs.

Name	Score
SNUMED INFO run4	0.667
SNUMED INFO run3	0.663
QUT RV run2	0.633
QUT RV run3	0.624
SNUMED INFO run2	0.611
INRIA ZENITH run1	**0.609**
SNUMED INFO run1	0.604
INRIA ZENITH run3	**0.592**
QUT RV run1	0.563
ECOUAN run1	0.487
INRIA ZENITH run2	**0.300**

If we compare the best runs of each team, the *INRIA Zenith Run 1*, the one using CNN, is ranked 3rd regarding to observation results. We can note that all the 4 best teams used Deep Neural Networks. Our second run, *INRIA Zenith Run 2*, the one using Fisher Vectors, is disappointingly distanced by the CNN runs: its final score is two times lower (0.3 instead of 0.609 for *INRIA Zenith Run 1*). In LifeCLEF 2014, the best performances were obtained by Fisher Vectors, but the use of external training data was not allowed which explains why CNN were not performing better.

Our final run, *INRIA Zenith Run 3*, is the Bayesian inference fusion method using previous runs. It was made in order to benefit from both technologies. Unfortunately, the results obtained are a little bit lower than the standalone CNN of *INRIA Zenith Run 1* (0.592 instead of 0.609). Two main reasons can be highlighted to explain this quality loss. First, the two classifiers are not necessarily independent, thus, there combination does not enable to obtain quality gain. Second, building a confusion matrix for such high dimension problems (*i.e.* 1000×1000) is very challenging and the size of the test set is not enough to learn an accurate confusion.

6 Conclusion

Inria Zenith team submitted 3 runs, using different strategies. The first run was based on the well-known GoogLeNet CNN architecture, finetuned over Imagenet dataset, and using a max method to fuse image results to observation results. Our second run did not used external data, and was based on fisher vectors which

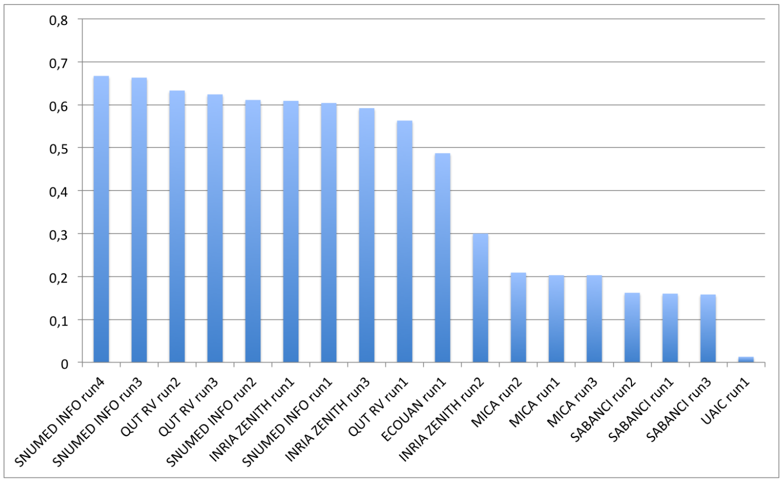

Fig. 2. Official results

was last year winning technology. The conclusion is that Deep Neural Networks outperforms fisher vectors for such classification tasks, particularly with an important number of classes, and when you have large training datasets. Our last run consisted in trying a new fusion method, based on Bayesian inference, to merge results of the two previous runs. However results were not as good as expected, probably because the first run is already two times better than the second one.

7 Appendix: Complementary Results

Table 3. Results for individual images

Methods	**Score**
FV (color + basic texture)	0.184
FV (SIFT)	0.267
CNN	0.581

References

1. Cai, J., Ee, D., Pham, B., Roe, P., Zhang, J.: Sensor network for the monitoring of ecosystem: Bird species recognition. In: Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on. pp. 293–298 (Dec 2007)

Table 4. Fusion results for observations

Descriptors	Maximum	Borda	Bayesian
FV (color + basic texture)	0.196	0.197	0.186
FV (SIFT)	0.286	0.283	0.259
FV (color + basic texture + SIFT)	0.285	0.300	0.293
CNN	0.609	0.607	0.598
Global fusion	0.599	0.487	0.592

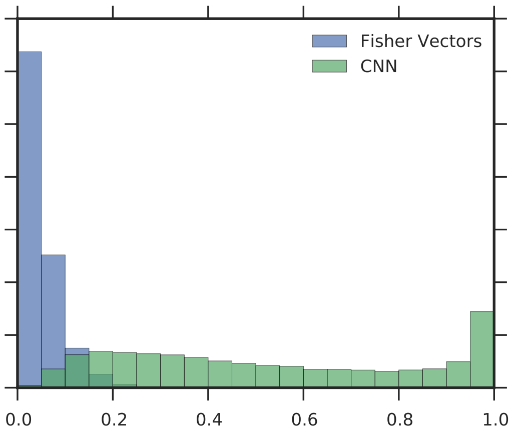

Fig. 3. Distribution of the top 1 probabilities returned by the CNN and the Fisher Vectors with Logistic Regression.

2. Cerutti, G., Tougne, L., Vacavant, A., Coquin, D.: A Parametric Active Polygon for Leaf Segmentation and Shape Estimation. In: 7th International Symposium on Visual Computing. p. 1. Las Vegas, United States (Sep 2011), https://hal.archives-ouvertes.fr/hal-00622269
3. Ellison, A.M., Farnsworth, E.J., Chu, M., Kress, W.J., Neill, A.K., Best, J.H., Pickering, J., Stevenson, R.D., Courtney, G.W., VanDyk, J.K.: Next-generation field guides (2013)
4. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. The Journal of Machine Learning Research 9, 1871–1874 (2008)
5. Gaston, K.J., O'Neill, M.A.: Automated species identification: why not? Philosophical Transactions of the Royal Society of London B: Biological Sciences 359(1444), 655–667 (2004)
6. van Gemert, J.C., Veenman, C.J., Smeulders, A.W., Geusebroek, J.M.: Visual word ambiguity. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32(7), 1271–1283 (2010)
7. Goëau, H., Bonnet, P., Joly, A., Affouard, A., Bakic, V., Barbe, J., Dufour, S., Selmi, S., Yahiaoui, I., Vignau, C., et al.: Pl@ ntnet mobile 2014: Android port and new features. In: Proceedings of International Conference on Multimedia Retrieval. p. 527. ACM (2014)
8. Goëau, H., Bonnet, P., Joly, A., Bakic, V., Barbe, J., Yahiaoui, I., Selmi, S., Carré, J., Barthélémy, D., Boujemaa, N., et al.: Plantnet mobile app. In: Proceedings of the 21st ACM international conference on Multimedia. pp. 423–424. ACM (2013)
9. Goëau, H., Joly, A., Bonnet, P.: Lifeclef plant identification task 2015. In: CLEF working notes 2015 (2015)
10. Goëau, H., Joly, A., Selmi, S., Bonnet, P., Mouysset, E., Joyeux, L.: Visual-based plant species identification from crowdsourced data. In: MM'11 - ACM Multimedia 2011. pp. 0–0. ACM, Scottsdale, United States (Nov 2011), https://hal.inria.fr/hal-00642236
11. Gosselin, P.H., Murray, N., Jégou, H., Perronnin, F.: Revisiting the fisher vector for fine-grained classification. Pattern Recognition Letters 49, 92–98 (2014)
12. Hsu, T.H., Lee, C.H., Chen, L.H.: An interactive flower image recognition system. Multimedia Tools Appl. 53(1), 53–73 (May 2011), http://dx.doi.org/10.1007/s11042-010-0490-6
13. Huang, Y., Wu, Z., Wang, L., Tan, T.: Feature coding in image classification: A comprehensive study. Pattern Analysis and Machine Intelligence, IEEE Transactions on 36(3), 493–506 (2014)
14. Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C.: Aggregating local image descriptors into compact codes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(9), 1704–1716 (2012)
15. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
16. Jiang, Y.G., Ngo, C.W., Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: Proceedings of the 6th ACM international conference on Image and video retrieval. pp. 494–501. ACM (2007)
17. Joly, A., Goëau, H., Bonnet, P., Bakić, V., Barbe, J., Selmi, S., Yahiaoui, I., Carré, J., Mouysset, E., Molino, J.F., et al.: Interactive plant identification based on social image data. Ecological Informatics 23, 22–34 (2014)

18. Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.P., Planque, R., Rauber, A., Fisher, R., Müller, H.: Lifeclef 2014: multimedia life species identification challenges. In: Information Access Evaluation. Multilinguality, Multimodality, and Interaction, pp. 229–249. Springer (2014)
19. Joly, A., Müller, H., Goëau, H., Glotin, H., Spampinato, C., Rauber, A., Bonnet, P., Vellinga, W.P., Fisher, B.: Lifeclef 2015: multimedia life species identification challenges
20. Kebapci, H., Yanikoglu, B., Unal, G.: Plant image retrieval using color, shape and texture features. Comput. J. 54(9), 1475–1490 (Sep 2011), http://dx.doi.org/10.1093/comjnl/bxq037
21. Kim, H.C., Ghahramani, Z.: Bayesian classifier combination. In: International conference on artificial intelligence and statistics. pp. 619–627 (2012)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
23. Kumar, N., Belhumeur, P.N., Biswas, A., Jacobs, D.W., Kress, W.J., Lopez, I.C., Soares, J.V.: Leafsnap: A computer vision system for automatic plant species identification. In: Computer Vision–ECCV 2012, pp. 502–516. Springer (2012)
24. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. vol. 2, pp. 2169–2178. IEEE (2006)
25. Mouine, S., Yahiaoui, I., Verroust-Blondet, A.: Advanced shape context for plant species identification using leaf image retrieval. In: Ip, H.H.S., Rui, Y. (eds.) ICMR '12 - 2nd ACM International Conference on Multimedia Retrieval. ACM, Hong Kong, China (Jun 2012), https://hal.inria.fr/hal-00726785
26. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Computer Vision, Graphics Image Processing, 2008. ICVGIP '08. Sixth Indian Conference on. pp. 722–729 (Dec 2008)
27. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. pp. 1–8. IEEE (2007)
28. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Computer Vision–ECCV 2010, pp. 143–156. Springer (2010)
29. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008)
30. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: Theory and practice. International journal of computer vision 105(3), 222–245 (2013)
31. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep Fisher networks for large-scale image classification. In: Advances in Neural Information Processing Systems (2013)
32. Simpson, E., Roberts, S., Psorakis, I., Smith, A.: Dynamic Bayesian Combination of Multiple Imperfect Classiers. In: Decision Making with Imperfect Decision Makers Springer (2012)
33. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. pp. 1470–1477. IEEE (2003)

34. Spampinato, C., Mezaris, V., van Ossenbruggen, J.: Multimedia analysis for ecological data. In: Proceedings of the 20th ACM international conference on Multimedia. pp. 1507–1508. ACM (2012)
35. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. CoRR abs/1409.4842 (2014), `http://arxiv.org/abs/1409.4842`
36. Trifa, V.M., Kirschel, A.N.G., Taylor, C.E., Vallejo, E.E.: Automated species recognition of antbirds in a Mexican rainforest using hidden Markov models. Journal of The Acoustical Society of America 123 (2008)
37. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3360–3367. IEEE (2010)
38. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 1794–1801. IEEE (2009)