Towards vegetation species discrimination by using data-driven descriptors

Keiller Nogueira[1], Jefersson A. dos Santos[1], Tamires Fornazari[2], Thiago Sanna Freire Silva[2], Leonor Patricia Morellato[2], and Ricardo da S. Torres[3]

[1] Department of Computer Science, Universidade Federal de Minas Gerais – UFMG
31270-010, Belo Horizonte, MG – Brazil
Email: keillernogueira@dcc.ufmg.br, jefersson@dcc.ufmg.br

[2] São Paulo State University – UNESP
13506-900, Rio Claro, SP – Brazil
Email: tsfsilva@rc.unesp.br, pmorella@rc.unesp.br

[3] Institute of Computing, University of Campinas – UNICAMP
13083-852, Campinas, SP – Brazil
Email: rtorres@ic.unicamp.br

***Abstract*—In this paper, we analyse the use of Convolutional Neural Networks (CNNs or ConvNets) to discriminate vegetation species with few labelled samples. To the best of our knowledge, this is the first work dedicated to the investigation of the use of deep features in such task. The experimental evaluation demonstrate that deep features significantly outperform well-known feature extraction techniques. The achieved results also show that it is possible to learn and classify vegetation patterns even with few samples. This makes the use of our approach feasible for real-world mapping applications, where it is often difficult to obtain large training sets.**

***Index Terms*—Deep Learning; Remote Sensing; Feature Learning; Image Classification; Machine Learning; High-resolution Images;**

I. Introduction

Phenology is the study of the life cycles of living beings, one of the most important indicators of climate change [1], and is often applied to the analysis of plant community changes over time, monitoring alterations such as leafing, budding, and flowering, usually based on field observations. Towards this monitoring, a key issue is to **first identify plant species**, which has been recently performed based on sensor images [2]–[5], that support this kind of studies without the need of on-the-ground observations, that are typically time consuming and error prone, especially in tropical regions where a single image may include a high number of species [2]. The main objective, in these cases, is the identification of regions within the vegetation sensor images that might be associated with species of interest, i.e., those whose phenology might be useful to be observed over time. Although interesting, this task of identify specific species for phenology studies suffers from several challenges, such as difficult to establish a specific pattern for each specie, given the high intraclass variance, and also distinguish between different species, given the interclass similarity of distinct species. Therefore, in this paper, we evaluate and analyze different strategies of exploiting pre-trained ConvNets for vegetation specie discrimination.

Through the years, several approaches have been proposed to support the discrimination of individuals of particular species [6]–[10]. Works related to this task can be divided into two main groups: those based on the use of machine learning fusion approaches, and those based on time series representations. All those works use general-purpose color and texture descriptors to represent image regions, which have several drawbacks. The major one is that different descriptors may produce distinct results depending on the data. Thus, it is imperative to design a full set of experiments in order to evaluate many descriptor algorithms looking for the most suitable ones for each application [11]. This process is also expensive and, likewise, does not guarantee an effective descriptive representation, since encoding the spatial features in an efficient and robust fashion is the key to generate discriminatory models for high spatial resolution images, which require state-of-the-art methods to handle the high complexity and huge amount of information.

In order to address these limitations, a resurgent method, called deep learning, has been used to learn specific and adaptable spatial features and classifiers for the images, all at once. Deep learning [12], [13] is a branch of machine learning that refers to multi-layered interconnected neural networks. Methods related to this branch has been achieving remarkable success in several visual recognition problems [14]–[22], showing effective capacity of encoding both visual properties and their spatial distribution based on the data itself [23].

Looking for exploiting these advantages, this paper **investigates** the use of data-driven deep descriptors for vegetation specie discrimination, being totally different from the aforementioned initiatives. Specifically, two possible strategies of exploiting ConvNets are evaluated targeting the discrimination specie task: (i) pre-trained ConvNets used as feature extractors,

and (ii) fine-tuned ConvNets. Both strategies rely on pre-trained ConvNets, i.e., networks trained on different data from the data of interest. The former strategy simply uses a pre-trained ConvNet as a feature extractor, by removing the last classification layer and considering its previous layer (or layers) as feature vector of the input data. The latter strategy performs a fine-tuning of the parameters of a pre-trained ConvNet. More specifically, layer filters (weights and bias), learned using any dataset, are adjusted to encode specific features of the target dataset, in this case, the vegetation dataset. Fine-tuning is a good strategy to be evaluated for this kind of application, since datasets related to this task tend to be small, which prevent the full training (from scratch) of a ConvNet.

In practice, we can summarize the contributions of this paper as follows:

- propose of a dataset for vegetation species discrimination composed of high-resolution multispectral images;
- analysis of the generalization of deep features for vegetation species discrimination;
- evaluation of two strategies to exploit deep features considering few labelled samples; and
- comparative analysis of ConvNets and successful low- and mid-level feature descriptors in a vegetation classification task.

This is a first attempt towards vegetation specie discrimination as well as towards the investigation of CNN-driven features. It is important to highlight that there is no free available high-resolution image datasets for evaluating vegetation discrimination in the phenology context.

II. ConvNets for Vegetation Species Discrimination

As introduced, in this paper, we investigate the use of ConvNets to performed vegetation species discrimination. The main problem of this kind of application (as well as for other phenology applications) is that only a few labelled data are available, which prevents designing and training ConvNets from scratch (with random initialization), since this process requires very large datasets and a significant amount of computational power. Fortunately, it is possible to tackle such applications by using pre-trained networks either as a fixed feature extractor for the task of interest or as an initialization for fine-tuning the parameters. Both strategies were evaluated in this work for vegetation discrimination and are described next, along with their advantages and drawbacks. Section II-A explains the use of deep ConvNets as a fixed feature extractor while Section II-B presents the fine-tuning process.

A. ConvNet as a Feature Extractor

Pre-trained networks can be used as a feature extractor for any image, since features learned in earlier layers are less dependent on the final application and could be used in many tasks. Specifically, features (usually, called deep features) can be extracted from any layer of a pre-trained network and then used in a given task. Deep features trained on a dataset of everyday objects have already achieved suitable results in applications like flower categorization [14], human attribute detection [15], bird sub-categorization [16], scene retrieval [17], and remote sensing [18], [19]. Furthermore, Razavian et al. [22] suggest that features obtained from deep learning should be the primary candidate in most visual recognition tasks.

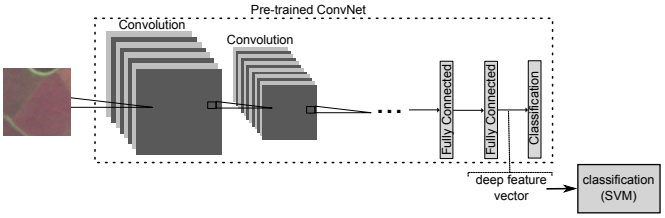


Fig. 1: Example of the use a ConvNet as feature extractor. The final classification layer is ignored and only the layer used to extract the deep features need to be defined. The figure shows the use of the features from the last layer before the classification layer, which is commonly used in the literature.

The strategy of using pre-trained ConvNets as feature extractors is very useful given its simplicity, since no retraining or tuning is necessary. Moreover, one only needs to select the layer to be used, extract the deep features, and use them combined with a machine learning technique, in case of a classification setup. According to previous works [18], [20], [22], deep features can be extracted from the last layer before the classification layer (usually, a fully-connected one) and, then, used to train a linear classifier, which is the strategy employed in this paper. Figure 1 illustrates how to use an existing ConvNet as a feature extractor.

B. Fine-tuned ConvNet

Fine-tuning is suitable for applications with reasonable but not enough large datasets to fully train a new network. It is a suitable option to extract the maximum effectiveness from pre-trained deep ConvNets, since it can significantly improve the performance of the final classifier.

The idea behind tuning deep ConvNets is based on the aforementioned fact that they tend to learn first-layer features that resemble either edges, or color blob detectors, independently of the training data. More specifically, the earlier layers of a network contain low-level filters that should be useful for many tasks. Later layers become progressively more specific to the details of the classes contained in the original dataset (i.e., the dataset in which the deep ConvNet was originally trained). Thus, initial layers can be preserved while the final layers must be adjusted to suit the application of interest.

Fine-tuning consists in performing a fine adjustment of the parameters in a pre-trained network by resuming the training of the network from a current setting of parameters but considering a new dataset of any size, aiming at accuracy improvements. It means that fine-tuning exploits the parameters learned from a previous training of the network on a specific dataset, and then adjusts the parameters from the current state

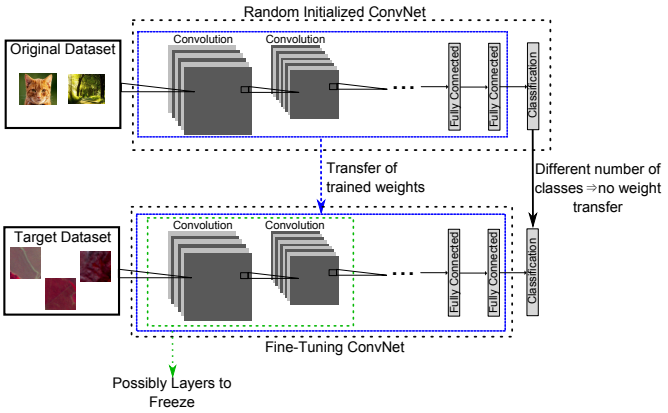


Fig. 2: Example of two options for the fine-tuning process. In one of them, all layers are fine-tuned according to the target dataset, but final layers have increased learning rates. In the other option, weights of initial layers can be frozen (green box) and only final layers are tuned.

for the new dataset, improving the performance of the final classifier.

In this paper, we exploited two approaches for fine-tuning a pre-trained network: (i) fine-tune all layers, and (ii) fine-tune only higher-level layers keeping some of the earlier layers fixed (due to overfitting concerns). It is important to emphasize that in both scenarios, the search space is bound to small variations in each step, since the learning rate is initialized with reduced value.

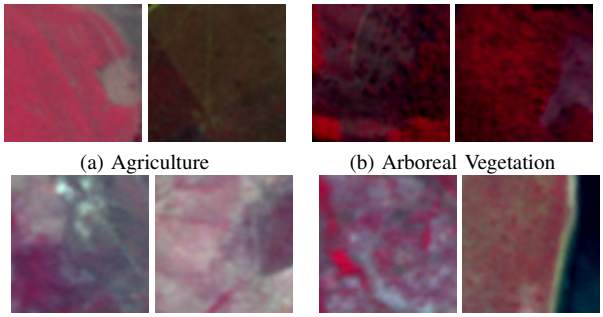
Concerning the first approach, some layers (usually the final ones, such as the classification layer, since the number of classes tend to be different) have weights ignored, being randomly initialized. These layers have the learning rate increased, so they can learn faster and converge, while the other layers may also change weights by very small variations, since they use the reduced value of the learning rate without any augmentation. Therefore, the first layers can use the information previously learned with few adjustments to the dataset of interest, and the final layers can learn based on the new dataset. In the second case, the initial layers are frozen to keep the generic features already learned, while the final layers are adjusted using the increased value of the learning rate. These two options of fine tuning are illustrated in Figure 2.

III. Experimental Protocol

In this section, we present the experimental setup, datasets and ConvNets used in this paper. The Brazilian Cerrado-Savanna Scenes Dataset is presented in Section III-A. Section III-B all baselines considered in this work, such as low-level (global) and mid-level (BoVW) descriptors. Evaluated ConvNet are presented Section III-C. Finally, Section III-D presents the protocol used in the experiments.

TABLE I: Class distribution of the dataset.

Class	#instances
Agriculture	47
Arboreal Vegetation	962
Herbaceous Vegetation	191
Shrubby Vegetation	111
Total	1,311



(a) Agriculture (b) Arboreal Vegetation

(c) Herbaceous Vegetation (d) Shrubby Vegetation

Fig. 3: Examples of the dataset.

A. Brazilian Cerrado-Savanna Scenes Dataset

The Brazilian Cerrado-Savanna Scenes Dataset, publicly released with this paper[1], is composed of 1,311 multi-spectral image segments of 64×64 pixels, extracted from images acquired by the RapidEye satellite sensors[2] over the Serra do Cipó region, a mountainous and highly biodiverse and heterogenous landscape in southern-central Brazil. These segments were labeled, by biologists and specialists, as belonging to one of four possible vegetation classes, as shown in Table I.

These images are composed of near-infrared, green, and red bands, which are the most useful and representative ones for discriminating vegetation areas, which may help in differentiating similar classes in the case of the vegetation dataset used in this work. This dataset is very challenging for several different reasons: (i) high intraclass variance, caused by different spatial configurations and densities of the same vegetation type, and (ii) high interclass similarity, given similar appearance of different types of vegetation species. Some samples showing the aforementioned challenges are shown in Figure 3.

B. Baselines

Through the years, several descriptors were successful applied to all kind of applications, including remote sensing image classification [11], texture and color image retrieval/classification [24], [25], and web image retrieval [26], [27]. Based on these works, several feature extraction tech-

[1]The dataset as well as the folds used in this paper are available for download at: www.patreo.dcc.ufmg.br/downloads/brazilian-cerrado-savanna-dataset/

[2]RapidEye system consists of a commercial constellation of five identical satellites that allow imaging of the Earth's surface with a high spatial resolution (5m per pixel), at short time intervals.

niques, which include low- and mid-level descriptors, have been selected to be evaluated as baseline of our work.

1) Low-Level descriptors: There is a myriad of descriptors can be used to represent visual elements [26] and, clearly, different ones may provide distinct information about images producing contrastive results. Thus, a diverse set of 10 low-level descriptors (based on color, texture, and shape properties) were selected to be evaluated: Auto-Correlogram Color (**ACC**) [28], which maps the spatial information of colors by pixel correlations at different distances; Border/Interior Pixel Classification (**BIC**) [29], a simple color descriptor which computes two color histograms for an image: one for border pixels and other for interior pixels; Color Coherence Vector (**CCV**) [30], a color descriptors which computes two histograms: one for coherent regions (pixel with similar neighbors) and other for incoherent areas; Global Color Histogram (**GCH**) [31], which quantizes the color space in a uniform way and scans the image computing the number of pixels belonging to each color; Local Activity Spectrum (**LAS**) [32], which captures the spatial activity of a texture in the horizontal, vertical, diagonal, and anti-diagonal directions separately; Steerable Pyramid Decomposition (**SID**) [33], which uses a set of filters sensitive to different scales and orientations to extract mean and standard deviation; Unser (**Unser**) [34], which computes measures (such as energy, contrast, and entropy) over histograms of sums and of differences.

2) Mid-Level descriptors: Some representations are called mid-level since they have one more calculation step when compared to low-level representations. Specifically, a mid-level representation uses local features built upon low-level representations, creating a new representation for an image. Bag of visual words (**BoVW**) [35] and their variations [36]–[38] are considered mid-level representations, since these methods create a codebook of visual discriminating patches (visual words), and then compute statistics (using the codebook) about the visual word occurrences in the test image. BoVW descriptors have been the state-of-the-art for several years and are still important candidates to perform well in many tasks.

In this case, based on previous works [18], [37], BoVW was tested considering SIFT [39] as the low-level descriptor, dense sampling (with grid of circles with 6 pixels of radius), hard assignment, average pooling and varying the size of visual codebook in 1,000, 5,000 and 10,000.

C. Convolutional Neural Network

As proposed, to evaluate the benefits of deep descriptors, we selected the well-regarded AlexNet, proposed by Krizhevsky et al. [40], which was the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [41] in 2012. This ConvNet has 60 million parameters and 650,000 neurons, and consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final softmax. Its final architecture can be seen in Figure 4. AlexNet was a breakthrough work as it was the first to employ non-saturating neurons, GPU implementation of the convolution operation and dropout to prevent overfitting.

In the experiments, the network was implemented in Convolutional Architecture for Fast Feature Embedding [42], or simply Caffe, a fully open-source framework that affords clear and easy implementation of deep architectures. AlexNet was used as a feature extractor network by extracting the features from the last fully-connected layer (red one in Figure 4), which results in a feature vector of 4,096 dimensions. Also, AlexNet was fine-tuned considering two strategies, as presented in Section II-B: (i) giving more importance to the final softmax layer but without freezing any layer, and (ii) freezing the first **three** layers and giving normal importance to the final ones, which participate normally in the learning process. This freezing process is based on the fact that initial layers tend to learn generic features, such as edges, or color blob detectors.

D. Experimental Setup

The experiments were carried out considering a 5-fold cross-validation protocol. The dataset was arranged into five non-overlapping folds with near-equal size, with 265, 262, 261, 261, and 261 images, respectively.

When using the ConvNets as feature extractors, four sets are used as training while the last is the test set, since linear SVM was used as the final classifier and it does not require any parameter search. Also, it is important to emphasize that there is no training when using a pre-trained ConvNet (without fine-tuning) as feature extractor, thus there are no parameters to vary.

When performing fine-tuning, at each run three folds are used as training set, one fold is used as validation (used to evaluate the current parameters of the network), and the remaining fold is used as a test set. For each run, the fine-tuning process starts from the beginning. Therefore, at the end, five different networks are obtained, one for each step of the 5-fold cross-validation process. Also, when fine-tuning, we basically preserve the original parameters of the ConvNet, varying only two: (i) number of maximum iterations, and (ii) learning rate. Both parameters were evaluated in a full set of preliminary experiments and, at the end, only the best set of parameters were selected, which are, in this case, 50,000 and 0.001 for number of iterations and learning rate, respectively.

The results are reported in terms of average accuracy and standard deviation among the 5 folds. For a given fold, we computed the accuracy for each class and then computed the average accuracy among classes. This accuracy was used to compute the final average accuracy among the 5 folds. All experiments were performed on a 64-bit Intel i7 4960X machine with 3.6GHz clock and 64GB of RAM. Two GPUs were used: a GeForce GTX770 with 4GB of internal memory and a GeForce GTX Titan X with 12GB of memory, both under a 7.5 CUDA version. Ubuntu version 14.04.3 LTS was used as operating system.

IV. Results and Discussion

In this section, we present the experimental results. In Section IV-A, we compare the performance of the deep-based

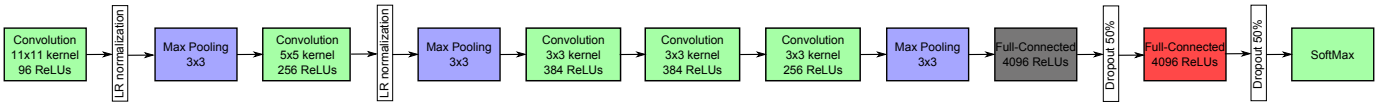


Fig. 4: Architectures of AlexNet [40]. The red box indicates the layer from where features were extracted in the case of using the ConvNet as feature extractors.

TABLE II: Comparison between different strategies to exploit pre-trained ConvNet.

Technique	Overall Accuracy (%)
Feature Extraction + SVM	88.18 ± 2.38
Fine Tuning	90.31 ± 1.74
Fine Tuning (Freeze Layers)	**90.54** ± 1.83
Fine Tuning + SVM	87.72 ± 1.09
Fine Tuning (Freeze) + SVM	87.42 ± 2.48

feature representation strategies. A comparison between the most accurate strategy against some of well-known descriptor methods are performed in Section IV-B.

A. Deep Learning Results

In this section, we compare the performance of different strategies for exploiting the benefits of deep learning: used as feature extractors and fine-tuned. Table II shows the comparison of the strategies in terms of average accuracy and standard deviation. It is important to highlight that the instances in the table with the label "SVM" refers to the ConvNet used as a feature extractor, i.e., with linear SVM as the final classifier. However, it is worth mentioning that SVM was used only after the fine-tuning, not during the training process. There are several interesting aspects. The first one is that fine-tuning was the best strategy, outperforming all others. The reason is that the fine-tuning strategy, as introduced, uses the edges and local structures learned in a dataset, but with small adjustments considering the target dataset (in this case, the Brazilian Cerrado-Savanna Scenes Dataset). This strategy allows the network to learn more specific features. Between the two fine-tuning strategies, there is no conclusion, since both achieved similar results, being statistically equal.

B. Comparison With Baselines

In this section, we compare the performance of the best ConvNets approaches (fine-tuning strategies) against baselines. As introduced, low- and mid-level feature representations were used as baselines. Table III shows the comparison in terms of average accuracy and standard deviation. The results show that all baselines were outperformed by the fine-tuning strategies, which are now the current state-of-the-art for the Brazilian Cerrado-Savanna Scenes dataset, with, approximately 90% average accuracy. Indeed, all results obtained with deep learning strategies (presented in Table II) were better than any baseline. It shows the power of feature learning inside the deep learning strategies. Besides that, two interesting aspects must be noticed. First, BIC [29] was the second best descriptor (82.53 ± 1.43) with a considerable result for a handcrafted visual descriptor. Furthermore, the Unser descriptor yielded a good result (80.32 ± 0.18) with only 32 features, being a suitable descriptor with a reduced amount of features.

TABLE III: Comparison between proposed use of pre-trained ConvNet and baselines.

Technique	Feature Vector Size	Overall Accuracy (%)
Fine Tuning		90.31 ± 1.74
Fine Tuning (Freeze Layers)		**90.54** ± 1.83
ACC [28]	256	76.43 ± 1.87
BIC [29]	128	82.53 ± 1.43
CCV [30]	128	80.56 ± 2.28
GCH [31]	64	80.10 ± 2.35
LAS [32]	256	80.02 ± 1.18
SID [33]	16	73.38 ± 0.58
Unser [34]	32	80.32 ± 0.18
SIFT+BoVW [35]	1,000	72.00 ± 2.12
SIFT+BoVW [35]	5,000	76.35 ± 2.24
SIFT+BoVW [35]	10,000	74.45 ± 0.23

V. Conclusions and Future Work

In this paper, two strategies for exploiting existing ConvNets were evaluated in the context of vegetation discrimination: used as feature extractors and fine-tuned. Specifically, these strategies were evaluated using AlexNet [40], a famous ConvNet responsible for several breakthroughs. The objective was to understand the best way to extract all feasible benefits from these state-of-the-art deep learning approaches in problems that usually are not suitable for the design and creation of new ConvNets from scratch, given the small number of labeled data. Also, a comparison between the strategies and traditional low- and mid-level descriptors considering a vegetation dataset were performed in order to point out the best methods. The results point that *fine tuning* tends to be the best strategy in different situations.

As future work, we intend to create a fine-grained and larger dataset to verify our initial conclusions obtained in this work. Also, we intend to test the evaluated strategies in different scenarios, including information-rich hyperspectral imagery (with hundreds of information channels) and images obtained by imaging Unmanned Aerial Vehicles, which are capable of yielding spatial resolutions two orders of magnitude lower than the presently evaluated RapidEye imagery, incurring in much larger intra-class heterogeneity among samples.

Acknowledgment

This work was partially financed by CNPq (grant #449638/2014-6), CAPES, Fapemig (#APQ-00768-14), and the FAPESP-Microsoft Virtual Institute (grants #2013/50169-1 and #2013/50155-0). The authors thank the support of NVIDIA Corporation, which donate a GeForce TITAN X

used for this research, and thank the support of the Brazilian Ministry of Environment by allowing free non-commercial use of the RapidEye imagery.

REFERENCES

[1] M. D. Schwartz, *Phenology: An Integrative Environmental Science*. Springer, 2013.

[2] B. Alberton, J. Almeida, R. Helm, R. da S. Torres, A. Menzel, and L. P. C. Morellato, "Using phenological cameras to track the green up in a cerrado savanna and its on-the-ground validation," *Ecological Informatics*, vol. 19, no. 0, pp. 62 – 70, January 2014.

[3] S. Nagai, T. Inoue, T. Ohtsuka, S. Yoshitake, K. N. Nasahara, and T. M. Saitoh, "Uncertainties involved in leaf fall phenology detected by digital camera," *Ecological Informatics*, vol. 30, pp. 124–132, 2015.

[4] T. Inoue, S. Nagai, H. Kobayashi, and H. Koizumi, "Utilization of ground-based digital photography for the evaluation of seasonal changes in the aboveground green biomass and foliage phenology in a grassland ecosystem," *Ecological Informatics*, vol. 25, pp. 1–9, 2015.

[5] S. Nagai, T. Ichie, A. Yoneyama, H. Kobayashi, T. Inoue, R. Ishii, R. Suzuki, and T. Itioka, "Usability of time-lapse digital camera images to detect characteristics of tree phenology in a tropical rainforest," *Ecological Informatics*, vol. 32, pp. 91–106, 2016.

[6] J. Almeida, J. A. dos Santos, B. Alberton, R. da S. Torres, and L. P. C. Morellato, "Applying machine learning based on multiscale classifiers to detect remote phenology patterns in cerrado savanna trees," *Ecological Informatics*, vol. 23, no. 0, pp. 49 – 61, 2014, special Issue on Multimedia in Ecology and Environment.

[7] F. A. Faria, J. Almeida, B. Alberton, L. P. C. Morellato, A. Rocha, and R. da Silva Torres, "Time series-based classifier fusion for fine-grained plant species recognition," *Pattern Recognition Letters*, pp. –, 2016.

[8] J. Almeida, J. A. dos Santos, B. Alberton, L. P. C. Morellato, and R. da Silva Torres, "Phenological visual rhythms: Compact representations for fine-grained plant species identification," *Pattern Recognition Letters*, pp. –, 2016.

[9] J. Almeida, J. A. dos Santos, W. O. Miranda, B. Alberton, L. P. C. Morellato, and R. da S. Torres, "Deriving vegetation indices for phenology analysis using genetic programming," *Ecological Informatics*, vol. 26, Part 3, pp. 61 – 69, 2015.

[10] F. A. Faria, J. Almeida, B. Alberton, L. P. C. Morellato, and R. da S. Torres, "Fusion of time series representations for plant recognition in phenology studies," *Pattern Recognition Letters*, pp. –, 2016.

[11] J. A. dos Santos, O. A. B. Penatti, P.-H. Gosselin, A. X. Falcao, S. Philipp-Foliguet, and R. da S Torres, "Efficient and effective hierarchical feature propagation," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 7, no. 12, pp. 4632–4643, 2014.

[12] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep learning," 2016, book in preparation for MIT Press. [Online]. Available: http://goodfeli.github.io/dlbook/

[14] N. Sunderhauf, C. McCool, B. Upcroft, and P. Tristan, "Fine-grained plant classification using convolutional neural networks for feature extraction," in *Working notes of CLEF 2014 conference*, 2014.

[15] K. Hara, V. Jagadeesh, and R. Piramuthu, "Fashion apparel detection: The role of deep convolutional neural network and pose-dependent priors," *arXiv preprint arXiv:1411.5319*, 2014.

[16] Z. Ge, C. McCool, C. Sanderson, A. Bewley, Z. Chen, and P. Corke, "Fine-grained bird species recognition via hierarchical subset learning," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 561–565.

[17] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 584–599.

[18] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Computer Vision and Pattern Recognition Workshop*, 2015, pp. 44–51.

[19] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14 680–14 707, 2015.

[20] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 580–587.

[22] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshop*, 2014, pp. 512–519.

[23] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *CoRR*, vol. abs/1602.01517, 2016.

[24] Y. Yang and S. Newsam, "Comparing sift descriptors and gabor texture features for classification of remote sensed imagery," in *International Conference on Image Processing*, 2008, pp. 1852–1855.

[25] J. A. dos Santos, O. A. B. Penatti, and R. da S. Torres, "Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification," in *International Conference on Computer Vision Theory and Applications*, 2010, pp. 203–208.

[26] O. A. B. Penatti, E. Valle, and R. da S. Torres, "Comparative study of global color and texture descriptors for web image retrieval," *Journal of Visual Communication and Image Representation*, vol. 23, no. 2, pp. 359–380, 2012.

[27] Y. Yang and S. Newsam, "Geographic image retrieval using local invariant features," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 51, no. 2, pp. 818–832, 2013.

[28] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Computer Vision and Pattern Recognition*, 1997, pp. 762–768.

[29] R. de O. Stehling, M. A. Nascimento, and A. X. Falcao, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *International Conference on Information and Knowledge Management*, 2002, pp. 102–109.

[30] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the fourth ACM international conference on Multimedia*. ACM, 1997, pp. 65–73.

[31] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[32] B. Tao and B. W. Dickinson, "Texture recognition and image retrieval using gradient indexing," *Journal of Visual Communication and Image Representation*, vol. 11, no. 3, pp. 327–342, 2000.

[33] J. A. Montoya-Zegarra, N. J. Leite, and R. da Silva Torres, "Rotation-invariant and scale-invariant steerable pyramid decomposition for texture image retrieval," in *SIBGRAPI*, 2007, pp. 121–128.

[34] M. Unser, "Sum and difference histograms for texture classification," *Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 118–125, 1986.

[35] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *International Conference on Computer Vision*, vol. 2, 2003, pp. 1470–1477.

[36] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek, "Visual word ambiguity," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1271–1283, 2010.

[37] O. A. B. Penatti, F. B. Silva, E. Valle, V. Gouet-Brunet, and R. da S. Torres, "Visual word spatial arrangement for image retrieval and classification," *Pattern Recognition*, vol. 47, no. 2, pp. 705–720, 2014.

[38] S. Avila, N. Thome, M. Cord, E. Valle, and A. de A. Araújo, "Pooling in image representation: the visual codeword point of view," *Computer Vision and Image Understanding*, vol. 117, no. 5, pp. 453–465, 2013.

[39] D. G. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, 2012, pp. 1106–1114.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[42] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.