Improving Spatial Feature Representation from Aerial Scenes by Using Convolutional Networks

Keiller Nogueira, Waner O. Miranda, Jefersson A. dos Santos

Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
Email: {keillernogueira, wanermiranda, jefersson}@ufmg.br

***Abstract*—The performance of image classification is highly dependent on the quality of extracted features. Concerning high resolution remote image images, encoding the spatial features in an efficient and robust fashion is the key to generating discriminatory models to classify them. Even though many visual descriptors have been proposed or successfully used to encode spatial features of remote sensing images, some applications, using this sort of images, demand more specific description techniques. Deep Learning, an emergent machine learning approach based on neural networks, is capable of learning specific features and classifiers at the same time and adjust at each step, in real time, to better fit the need of each problem. For several task, such image classification, it has achieved very good results, mainly boosted by the feature learning performed which allows the method to extract specific and adaptable visual features depending on the data. In this paper, we propose a novel network capable of learning specific spatial features from remote sensing images, with any pre-processing step or descriptor evaluation, and classify them. Specifically, automatic feature learning task aims at discovering hierarchical structures from the raw data, leading to a more representative information. This task not only poses interesting challenges for existing vision and recognition algorithms, but also brings huge opportunities for urban planning, crop and forest management and climate modelling. The propose convolutional neural network has six layers: three convolutional, two fully-connected and one classifier layer. So, the five first layers are responsible to extract visual features while the last one is responsible to classify the images. We conducted a systematic evaluation of the proposed method using two datasets: (i) the popular aerial image dataset UCMerced Land-use and, (ii) a multispectral high-resolution scenes of the Brazilian Coffee Scenes. The experiments show that the proposed method outperforms state-of-the-art algorithms in terms of overall accuracy.**

***Keywords*-Deep Learning; Remote Sensing; Feature Learning; Image Classification; Machine Learning; High-resolution Images;**

I. Introduction

A lot of information may be extracted from the earth's surface through images acquired by airborne sensors, such as spatial features and structural patterns. A wide range of fields have taken advantages of this information, including urban planning [1], crop and forest management [2], disaster relief [3] and climate modelling. However, extract information from these remote sensing images (RSIs), by manual efforts (e.g., using edition tools), is both slow and costly, so automatic methods appears as an appealing alternative for the community. Although the literature presents many advances, the spatial information coding in RSIs is still considered an open and challenging task [4]. Traditional automatic methods [5], [6] extract information from RSIs in two separated basic step: (i) spatial feature extraction and, (ii) learning step, that uses machine learning methods. In a typical scenario, since different descriptors may produce different results depending on the data, it is imperative to design and evaluate many descriptor algorithms in order to find the most suitable ones for each application [7]. This process is also expensive and, likewise, does not guarantee a good descriptive representation. Another automatic approach, called deep learning, overcome this limitation, since it can learn specific and adaptable spatial features and classifiers for the images, all at once. In this paper, we propose a method to automatic learn the spatial feature representation and classify each remote sensing image focusing on the deep learning strategy.

Deep learning [8], a branch of machine learning that favours multi-layered neural networks, is commonly composed with a lot of layers (each layer composed of processing units) that can learn the features and the classifiers at the same time, i.e, just one network is capable of learning features (in this case, spatial ones) and classifiers (in different layers) and adjust this learning, in processing time, based on the accuracy of the network, giving more importance to one layer than another depending on the problem. Since encoding the spatial features in an efficient and robust fashion is the key to generating discriminatory models for the remote sensing images, this feature learning step, which may be stated as a technique that learn a transformation of raw data input to a representation that can be effectively exploited [8], is a great advantage when compared to typical methods, such as typical aforementioned ones, since the multiple layers responsible for this, usually composed of nonlinear processing units, learn adaptable and specific feature representations in some form of hierarchy, depending on the data, with low-level features being learned in former layers and high-level in the latter ones. Thus, the network learns the features of different levels creating more robust classifiers that use all this extracted information.

In this paper, we propose a new approach to automatically classify aerial and remote sensing image scenes. We used a specific network called Convolutional Neural Network (CNN), or simply ConvNet. This kind of deep learning technique uses the natural property of a image being stationary, i.e., the statistics of one part of the image are the same as any other part. Thus, features learned at one part of the image can also be applied to other parts of the image, and the same features may be used at all locations. We proposed a network with six layers: three convolutional ones, two fully connected and one softmax at the end to classify the images. So, the five first layers are responsible to extract visual features while the last one is responsible to classify the images. Between some of these layers, we used some techniques, such as dropout regularization [9], Local Response Normalization (LRN) [10] and max-polling.

In practice, we claim the following benefits and contributions over existing solutions:

- Our main contribution is a novel ConvNet to improve feature learning of aerial and remote sensing images.
- A systematic set of experiments, using two datasets, reveals that our algorithm outperforms the state-of-the-art baselines [11], [12] in terms of overall accuracy measures.

The paper is structured as follows. Related work is presented in Section II. We introduce the proposed network in Section III. Experimental evaluation, as well as the effectiveness of the proposed algorithm, is discussed in Section IV. Finally, in Section V we conclude the paper and point promising directions for future work.

II. Related Work

The development of algorithms for spatial extraction information is a hot research topic in the remote sensing community [4]. It is mainly motivated by the recent accessibility of high spatial resolution data provided by new sensor technologies. Even though many visual descriptors have been proposed or successfully used for remote sensing image processing [13], [14], [15], some applications demand more specific description techniques. As an example, very successful low-level descriptors in computer vision applications do not yield suitable results for coffee crop classification, as shown in [7]. Thus, common image descriptors can achieve suitable results in most of applications. Furthermore, higher accuracy rates are yielded by the combination of complementary descriptors that exploits late fusion learning techniques. Following this trend, many approaches have been proposed for selection of spatial descriptors in order to find suitable algorithms for each application [16], [11], [17]. Cheriyadat [11] proposed a feature learning strategy based on Sparse Coding, which learned features from well-known datasets are used for building detection in larger image sets. Faria et al. [16] proposed a new method for selecting descriptors and pattern classifiers based on rank aggregation approaches. Tokarczyk et al. [17] proposed a boosting-based approach for the selection of low-level features for very-high resolution semantic classification.

Despite the fact the use of Neural Network-based approaches for remote sensing image classification is not recent [18], its massive use is recent motivated by the study on deep learning-based approaches that aims at the development of powerful application-oriented descriptors. Many works have been proposed to learn spatial feature descriptors [19], [20], [21], [12]. Firat et al. [19] proposed a method that combines Markov Random Fields with ConvNets for object detection and classification in high-resolution remote sensing images. Hung et al. [20] applied ConvNets to learn features and detect invasive weed. In [21], the authors presented an approach to learn features from Synthetic Aperture Radar (SAR) images. Zhang et al. [12] proposed a deep feature learning strategy that exploits a pre-processing salience filtering. Moreover, new effective hyperspectral and spatio-spectral feature descriptors [22], [23], [24], [25] have been developed mainly boosted by the deep learning growth in recently years.

Our work differs from others in the literature in many aspects. As introduced, classification accuracy is highly dependent on the quality of extracted features. A method that learns adaptable and specific spatial features based on the images could exploits better the feasible information available on the data. Moreover, to the best of our knowledge, there is no work in the literature that proposes a ConvNet-based approach to learn spatial features in both remote sensing and aerial domains. The ConvNet methods found in the literature are designed to be focused on very specific application scenarios, such as weed detection or urban objects. Thus, the proposed network is totally different (in the architecture, number of neurons and layers, etc) when compared to others in the literature. In this work, we experimentally demonstrate the robustness of our approach by achieving state-of-the-art results not only in a well-known aerial dataset but also in a remote sensing image dataset, which contains non-visible bands.

III. Convolutional Neural Networks for Remote Sensing Images

Neural Network (NN) is generally presented as systems of interconnected processing units (neurons) which can compute values from inputs leading to a output that may be used on further units. These neurons work in agreement to solve a specific problem, learning by example, i.e., a NN is created for a specific application, such as pattern recognition or data classification, through a learning process. ConvNets, a type of NN, were initially proposed to work over images, since it tries to take leverage from the natural property of an image, i.e., its stationary state. More specifically, the statistics of one part of the image are the same as any other part. Thus, features learned at one part can also be applied to another region of the image, and the same features can be used in several locations. When compared to other types of networks, convNets present several other advantages: (i) automatically learn local feature extractors, (ii) are invariant to small translations and distortions in the input pattern, and (iii) implement the principle of weight sharing which drastically reduces the number of free parameters and thus increases their generalization capacity.

The proposed ConvNet has six layers: three convolutional, two fully-connected and one classifier layer. So, the five first layers are responsible to extract visual features while the last one, a *softmax* layer, is responsible to classify the images. Next, we present some basic concepts followed by the proposed architecture.

A. Processing Units

As introduced, artificial neurons are basically processing units that compute some operation over several input variables and, usually, have one output calculated through the activation function. Typically, an artificial neuron has a weight vector $W = (w_1, w_2, \cdots, w_n)$, some input variables $X = (x_1, x_2, \cdots, x_n)$ and a threshold or bias b. Mathematically, vectors w and x have the same dimension, i.e., w and x are in $\Re^n$. The full process of a neuron may be stated as in Equation 1.

$$z = f\left(\sum_{i}^{N} X_i * W_i + b\right) \tag{1}$$

where z, x, w and b represent output, input, weights and bias, respectively. $f(\cdot) : \Re \rightarrow \Re$ denotes an activation function.

Conventionally, a nonlinear function is provided in $f(\cdot)$. There are a lot of alternatives for $f(\cdot)$, such as sigmoid, hyperbolic, and rectified linear function. In this paper, we are interested in the latter one because neurons with this configuration has several advantages when compared to others: (i) works better to avoid saturation during the learning process, (ii) induces the sparsity in the hidden units, and (iii) does not face gradient vanishing problem[1] as with sigmoid and tanh function. The processing unit that uses the rectifier as activation function is called Rectified Linear Unit (ReLU) [26]. The first step of the activation function of a ReLU is presented in Equation 1 while the second one is introduced in Equation 2.

$$a = \begin{cases} z, if z > 0 \\ 0, otherwise \end{cases} \qquad \Leftrightarrow \qquad a = f(z) = max(0, z) \tag{2}$$

The processing units are grouped into layers, which are stacked forming multilayer NNs. These layers give the foundation to others, such as convolutional and fully-connected.

B. Network Components

Amongst the different layers, the convolutional one is the responsible to capture the features from the images, where the first layer obtains the low-level features (like edges, lines and corners) while the others get high-level features (like structures, objects and shapes). The process made in this layer can be decomposed into two phases: (i) the convolution step, where a fixed-size window runs over the image defining a region of interest, and (ii) the processing step, that uses the pixels inside each window as input for the neurons that, finally, perform the feature extraction from the region. Formally, in the latter step, each pixel is multiplied by its respective weight generating the output of the neuron, just like Equation 1. Thus, only one output is generated concerning each region defined by the window. This iterative process results in a new image (or feature map), generally smaller than the original one, with the visual features extracted. Many of these features are very similar, since each window may have common pixels, generating redundant information. Typically, after each convolutional layer, there are pooling layers that were created in order to reduce the variance of features by computing some operation of a particular feature over a region of the image. Specifically, a fixed-size window runs over the features extracted by the convolutional layer and, at each step, a operation is realized to minimize the amount and optimize the gain of the features. Two operations may be realized on the pooling layers: the max or mean operation, which selects the maximum or mean value over the feature region, respectively. This process ensures that the same result can be obtained, even when image features have small translations or rotations, being very important for object classification and detection. Thus, the pooling layer is responsible for sampling the output of the convolutional one preserving the spatial location of the image, as well as selecting the most useful features for the next layers.

After several convolutional and pooling layers, there are the fully-connected ones. It takes all neurons in the previous layer and connects it to every single neuron it has. The previous layers can be convolutional, pooling or fully-connected, however the next ones must be fully-connected until the classifier layer, because the spatial notion of the image is lost in this layer. Since a fully-connected layer occupies most of the parameters, overfitting can easily happen. To prevent this, the dropout method [27] was employed. This method randomly drops several neuron outputs, which does not contribute to the forward pass and backpropagation anymore. This neuron drops are equivalent to decreasing the number of neurons of the network, improving the speed of training and making model combination practical, even for deep neural networks. Although this method creates neural networks with different architectures, those networks share the same weights, permitting model combination and allowing that only one network is needed at test time.

Finally, after all convolution, pooling and fully-connected layers, a classifier layer may be used to calculate the class probability of each instance. The most common classifier layer is the softmax one [8], based on the namesake function. The softmax function, or normalized exponential, is a generalization of the multinomial logistic function that generates a K-dimensional vector of real values in the range $(0, 1)$ which represents a categorical probability distribution. Equation 3 shows how softmax function predicts the probability for the jth class given a sample vector X.

$$h_{W,b}(X) = P(y = j|X; W, b) = \frac{\exp^{X^T W_j}}{\sum_{k=1}^{K} \exp^{X^T W_k}} \tag{3}$$

where j is the current class being evaluated, X is the input vector, and W represent the weights.

[1] The gradient vanishing problem occurs when the propagated errors become too small and the gradient calculated for the backpropagation step vanishes, making impossible to update the weights of the layers and achieve a good solution.

In addition to all these processing layers, there are also normalization ones, such as Local Response Normalization (LRN) [28] layer. This is the most useful when using processing units with unbounded activations (such as ReLU), because it permits the local detection of high-frequency features with a big neuron response, while damping responses that are uniformly large in a local neighborhood.

C. Training

After modelling a network, in order to allow the evaluation and improvement of its results, a loss function needs to be defined, even because the goal of the training is to minimize the error of this function, based on the weights and bias, as presented in Equation 4. Amongst several functions, the log loss one has become more pervasive because of exciting results achieved in some problems [28]. Equation 5 presents a general log loss function, without any regularization term.

$$\arg\min_{W,b}[\mathcal{J}(W,b)] \tag{4}$$

$$\begin{aligned}\mathcal{J}(W,b) = -\frac{1}{N}\sum_{i=1}^{N}(y^{(i)} \times \log h_{W,b}(x^{(i)}) + \\ (1-y^{(i)}) \times \log(1-h_{W,b}(x^{(i)})))\end{aligned} \tag{5}$$

where y represents a possible class, x is the data of an instance, W the weights, i is an specific instance, and N represents the total number of instances.

With the cost function defined, the neural network can be trained in order to minimize the loss by using some optimization algorithm, such as Stochastic Gradient Descent (SGD), to gradually update the weights and bias in search of the optimal solution:

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha\frac{\partial\mathcal{J}(W,b)}{\partial W_{ij}^{(l)}}$$
$$b_{i}^{(l)} = b_{i}^{(l)} - \alpha\frac{\partial\mathcal{J}(W,b)}{\partial b_{i}^{(l)}}$$

where α denotes the learning rate.

However, as presented, the partial derivatives of the cost function, for the weights and bias, are needed. To obtain these derivatives, the backpropagation algorithm is used. Specifically, it must calculate how the error changes as each weight is increased or decreased slightly. The algorithm computes each error derivative by first computing the rate at which the error δ changes as the activity level of a unit is changed. For classifier layers, this error is calculated considering the predicted and desired output. For other layers, this error is propagated by considering the weights between each pair of layers and the error generated in the most advanced layer.

The training step of our Neural Network occurs in two steps: (i) the feed-forward one, that passes the information through all the network layers, from the first until the classifier one, and (ii) the backpropagation one, which calculates the error δ generated by the Neural Network and propagates this error through all the layers, from the classifier until the first one. As presented, this step also uses the errors to calculate the partial derivatives of each layers for the weights and bias.

D. Final Architecture

Figure 1 presents the final architecture of our CNN. The proposed network maximizes the multinomial logistic regression objective. Specifically, Equation 6 presents the loss function of the proposed network that is, actually, a simplified form of the function presented in Equation 5 with a new regularization term, called weight decay, to help prevent overfitting.

$$\begin{aligned}\mathcal{J}(W,b) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{1}1\{y^{(i)}=k\}\times \\ \times P(y^{(i)}=j|x^{(i)};W,b) + \frac{\lambda}{2}\sum W^2\end{aligned} \tag{6}$$

where y represents a possible class, x is the data of an instance, W the weights, i is an specific instance and N represents the total number of instances. The $1\{\cdot\}$ is the "indicator function" so that $1\{a\,true\,statement\} = 1$, and $1\{a\,false\,statement\} = 0$.

The kernels of all convolutional layers are connected to all kernel maps in the subsequent layer. The neurons in the fully-connected layers are connected to all neurons in the previous layer. Local Response Normalization (LRN) layers follow the first and second convolutional layers. Max-pooling layers follow both response-normalization layers as well as the third convolutional layer. The ReLU non-linearity is applied to the output of every convolutional and fully-connected layer. The first convolutional layer filters the input image, which may have varied size depending on the application, with 96 kernels of size $5 \times 5 \times 3$ with a stride[2] of 3 pixels. The second convolutional layer uses the (response-normalized and pooled) output of the first convolutional layer as input and filters it with 256 kernels. The third convolutional layer has 256 kernels connected to the (normalized, pooled) outputs of the second convolutional layer. Finally, the fully-connected layers have 1024 neurons each and the classifier one has the probability distribution over the possible classes.

IV. Experimental Evaluation

In this section, we present the experimental setup as well as the results obtained.

A. Dataset

Datasets with different properties were chosen in order to better evaluate the robustness and effectiveness of the proposed network and the features learned with it. The first one is a multi-class land-use dataset that contains aerial high resolution scenes in the visible spectrum. The second dataset has multispectral high-resolution scenes of coffee crops and non-coffee areas.

[2] This is the distance between the centers of each window step.

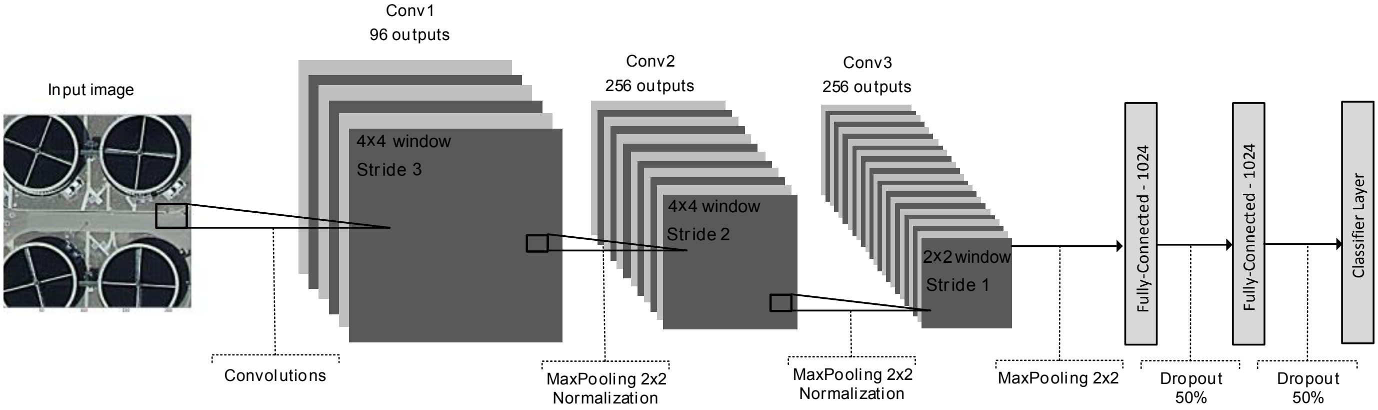

Fig. 1. The proposed Convolution Neural Network architecture. It contains six layers: the first three are convolutional, two others are fully-connected. The output of the last fully-connected layer is fed into a classifier layer which produces the probability distribution over the possible class labels.

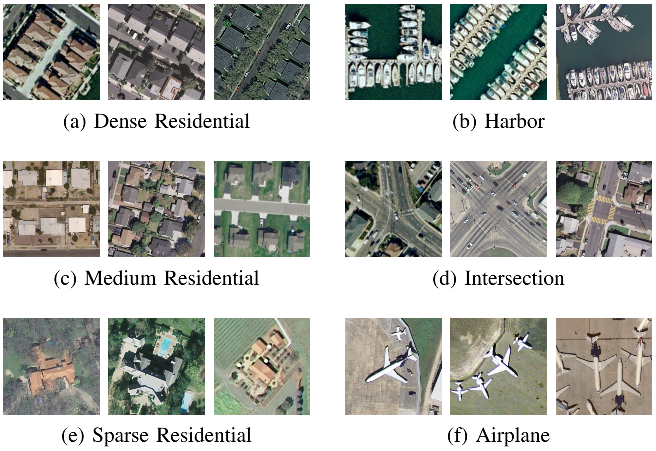
(a) Dense Residential
(b) Harbor
(c) Medium Residential
(d) Intersection
(e) Sparse Residential
(f) Airplane

Fig. 2. Some samples from the UCMerced Land Use Dataset.

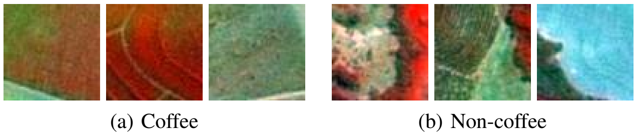
(a) Coffee
(b) Non-coffee

Fig. 3. Example of coffee and non-coffee samples in the Brazilian Coffee Scenes dataset. The similarity among samples of opposite classes is notorious as well as the intraclass variance.

1) UCMerced Land-use Dataset: This manually labelled and publicly available dataset [29] is composed of 2,100 aerial scene images with 256 × 256 pixels equally divided into 21 land-use classes selected from the United States Geological Survey (USGS) National Map: agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium density residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis courts.

The data set represents highly overlapping classes such as the dense residential, medium residential, and sparse residential which mainly differs in the density of structures. Samples of some class are shown in Figure 2. For providing diversity to the dataset, these images, that have pixel resolution of one foot, were obtained from different US locations.

2) Brazilian Coffee Scenes: This dataset [30] is composed of scenes taken by the SPOT sensor in 2005 over four counties in the State of Minas Gerais, Brazil: Arceburgo, Guaranésia, Guaxupé and Monte Santo. This dataset is very challenging for several different reasons: (i) high intraclass variance, caused by different crop management techniques, (ii) scenes with different plant ages, since coffee is an evergreen culture and, (iii) images with spectral distortions caused by shadows, since the South of Minas Gerais is a mountainous region.

This dataset has 2,876 multispectral high-resolution scenes, with 64 × 64 pixels, equally divided into two classes: *coffee* and *non-coffee*. Figure 3 shows some samples of these classes.

B. Baselines

We used several recently proposed methods as baseline [12], [11]. For the UCMerced Land-use dataset, only the best method of each work [12], [11] were considered as baseline. Cheriyadat [11] proposed an unsupervised method that uses features extracted by dense low-level descriptors to learn a set of basis functions. Thus, the low-level feature descriptors are encoded in terms of the basis functions to generate new sparse representation for the feature descriptors. A linear SVM is used over this representation, classifying the images. For this scenario, dense sift with feature encoding (using the basis functions) yielded the best result for the UCMerced dataset, and was used as baseline. In [12], salient regions are exploited by an unsupervised feature learning method to learn a set of feature extractors which are robust and efficient and do not need elaborately designed descriptors such as the scale-invariant-feature-transform-based algorithm. Then, a machine learning technique is used over the features extracted by the proposed unsupervised method, classifying the images. In this case, for the UCMerced dataset, linear SVM with the proposed saliency algorithm yielded the best result.

For the Brazilian Coffee Scenes dataset, we have used BIC [31] and ACC [32] descriptors with Linear SVMs as baselines. We choose the aforementioned descriptors based on several works, such as [14], [33], [30], which demonstrate that these are the most suitable descriptors to describe coffee crops.

C. Experimental Protocol

We conducted a five-fold cross validation in order to assess the accuracy of the proposed algorithm for both dataset. Therefore, the dataset was arranged into five folds with almost same size, i.e., the images are almost equally divided into five sets, where each one is balanced in relation to the number of images per class, so one fold may not have images from only or a few classes, giving diversity to each set. At each run, three folds are used as training-set, one as validation (used to tune the parameters of the network) and the remaining one is used as test-set. The results reported are the average of the five runs followed by the standard deviation.

The proposed ConvNets was built by using a framework called Convolutional Architecture for Fast Feature Embedding [34], or simply Caffe. This framework is more suitable due to its simplicity and support to parallel programming using CUDA®, a NVidia® parallel programming based on graphics processing units. Thus, in this paper, Caffe was used along with libraries as CUDA and CuDNN [3]. All computational presented experiments were performed on a 64 bits Intel® i5® 760 machine with 2.8GHz of clock and 20GB of RAM memory. A GeForce® GTX760 with 4GB of internal memory was used as graphics processing units, under a 6.5 CUDA version. Fedora 20 (kernel 3.11) was used as operating system.

The ConvNet and its parameters were adjusted by considering a full set of experiments guided by [35]. We started the setup experiments with a small network and, after each step, new layers, with different number of processing units, were being attached until a plateau was reached, i.e., until there is no change in the loss and accuracy of the network. At the end, a initial architecture was obtained. After defining this architecture, the best set of parameters was selected based on convergence velocity versus the numbers of iterations needed. During this step, a myriad of parameters combinations, for each dataset, were experimented and, for the best ones, new architectures, close to the initial one, were also experimented. For each dataset, we basically used the same network architecture proposed in Section III-D with several peculiarities related to the input image and the classifier layer. For the UCMerced Land-use Dataset, the input image has 256×256 pixels and the classifier layers has 21 neurons, since each image can be classified into 21 classes. For the Brazilian Coffee Scenes Dataset, the input image has 64×64 pixels and the classifier layers has 2 neurons, since the dataset has only 2 classes (coffee and non-coffee).

D. Results and Discussion

The results for the UCMerced Land-use dataset are presented in Table I. One can see that the proposed ConvNet outperforms all the baselines [11], [12] in, at least, 10% in terms of overall accuracy. It is worth to point out that all baselines are more hand-working, since features need to be extracted first to be, then, used with some machine learning technique (in this case SVM). Meanwhile, the proposed method does not need to extract the features in advance, since it can learn the features by itself.

TABLE I
RESULTS, IN TERMS OF ACCURACY, OF THE PROPOSED METHOD AND THE BASELINES FOR THE UCMERCED LAND-USE DATASET.

Method	Accuracy(%)
Our ConvNet	**89.39 ± 1.10**
With-Sal [12]	82.72 ± 1.18
Dense Sift [11]	81.67 ± 1.23

TABLE II
RESULTS, IN TERMS OF ACCURACY, OF THE PROPOSED METHOD AND THE BASELINES FOR THE BRAZILIAN COFFEE SCENES DATASET.

Method	Accuracy(%)
Our ConvNet	**89.79 ± 1.73**
BIC [31]+SVM	87.03 ± 1.17
ACC [32]+SVM	84.95 ± 1.98

The results for the Brazilian Coffee Scenes dataset are presented in Table II. Our ConvNet performs slight better than BIC and outperforms ACC. Once again, all baselines are more hand-working, since features need to be extracted first to be, then, used with some machine learning technique . In the opposite direction, as introduced, the proposed network learns all at once. Furthermore, it is worth to mention that agricultural scenes is very hard to classify since the method must to differentiate among different vegetation. BIC is showed to be the a suitable descriptor for coffee crop classification after several comparisons with other descriptors [14].

Figure 6 shows some features extracted by the network at each convolutional layer for Figure 6a. Moreover, Figure 7 shows some filters used by the network at each convolutional layer to extract the features of the aforementioned image. In this case, the convolutional layers are a collection of block filters capable of considering, not only the color channels (first convolutional layers, per example), but the gradients and contours considered useful for the classification.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new approach based on Convolutional Neural Networks to learn spatial feature arrangements from remote sensing domains. Experimental results show that our method is effective and robust. We have achieved state-of-the-art accuracy results for the well-known UCMerced dataset by outperforming all the baselines. Our method also presented suitable results for coffee crop classification, which is considered a challenging dataset.

As future work, we intend to fine-tune an existing network, such as ImageNet [28], and compare the results with the proposed method. We are also considering perform some modifications in our net in order to improve even more the obtained results, test new datasets and new applications.

[3] It is a GPU-accelerated library of primitives for deep neural networks

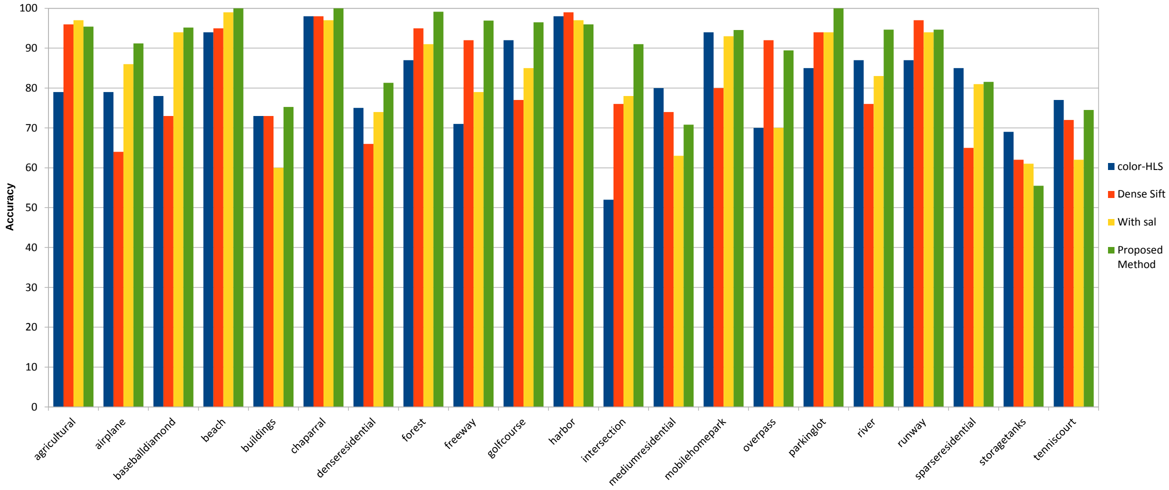

Fig. 4. Per-class classification rates of the proposed net and baselines for the UCMerced Land-use dataset.

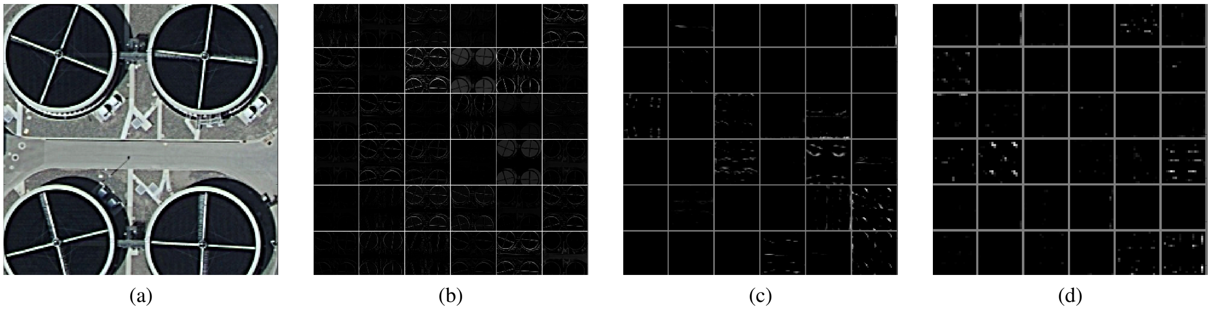
(a) (b) (c) (d)

Fig. 6. An image from the UCMerced Land-use dataset followed by the features extracted in the three convolutional layers of the network. (a) the original image, (b)-(d) features extracted from the first, second and third convolutional layer, respectively.

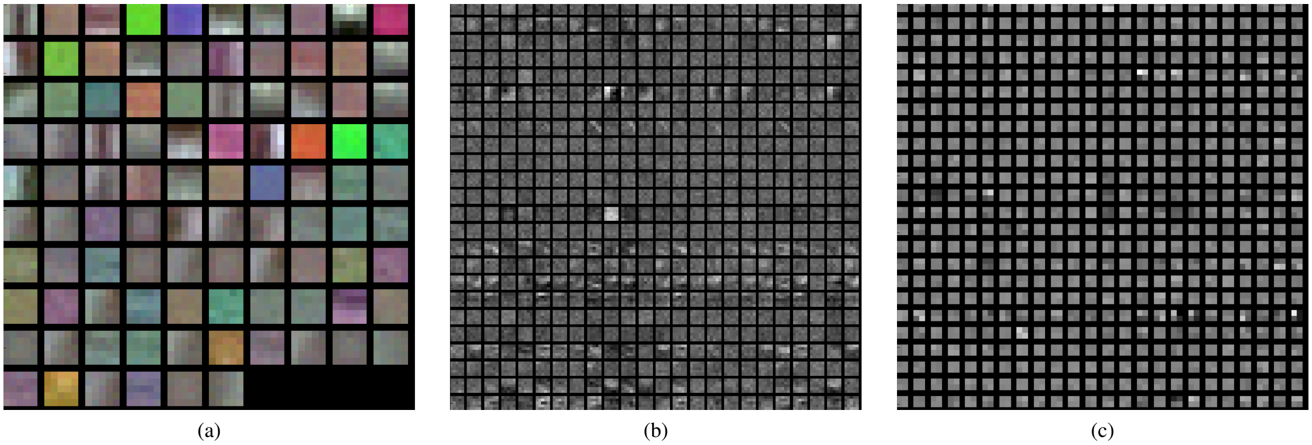
(a) (b) (c)

Fig. 7. Filters from each convolutional layers of the network for Figure 6a.

ACKNOWLEDGMENT

This work was partially financed by CNPq (grant 449638/2014-6), CAPES, and Fapemig (APQ-00768-14).

REFERENCES

[1] J. R. Taylor and S. T. Lovell, "Mapping public and private spaces of urban agriculture in chicago through the analysis of high-resolution aerial images in google earth," *Landscape and Urban Planning*, vol. 108, no. 1, pp. 57–70, 2012.

[2] U. Bradter, T. J. Thom, J. D. Altringham, W. E. Kunin, and T. G. Benton, "Prediction of national vegetation classification communities in the british uplands using environmental data at multiple spatial scales, aerial images and the classifier random forest," *Journal of Applied Ecology*, vol. 48, no. 4, pp. 1057–1065, 2011.

[3] S. Li, W. Li, J. Kan, and Y. Wang, "An image segmentation approach of forest fire area based on aerial image," *Journal of Theoretical and*

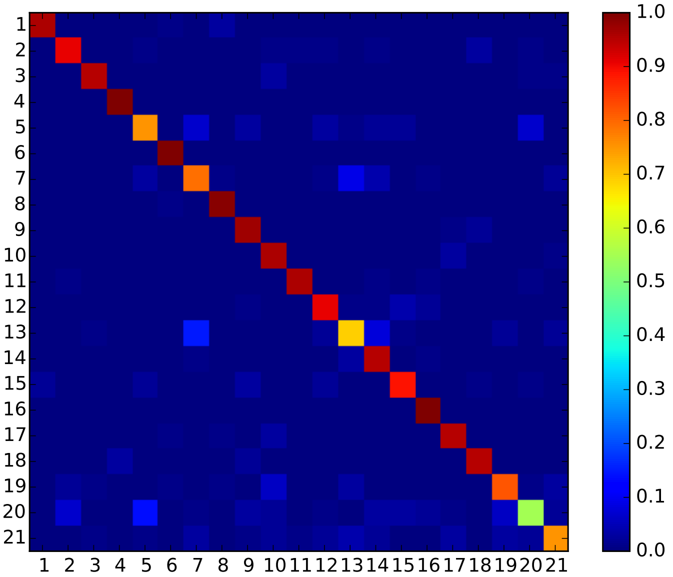

Fig. 5. Confusion matrix showing the classification performance with the UCMerced Land-use dataset. The rows and columns of the matrix denote the actual and predicted classes, respectively. The class labels are assigned as follows: 1 = Agricultural, 2 = airplane, 3 = baseballdiamond, 4 = beach, 5 = buildings, 6 = chaparral, 7 = denseresidential, 8 = forest, 9 = freeway, 10 = golfcourse, 11 = harbor, 12 = intersection, 13 = mediumresidential, 14 = mobilehomepark, 15 = overpass, 16 = parkinglot, 17 = river, 18 = runway, 19 = sparseresidential, 20 = storagetanks and 21 = tenniscourt.

Applied Information Technology, vol. 46, no. 1, 2012.

[4] J. Benediktsson, J. Chanussot, and W. Moon, "Advances in very-high-resolution remote sensing [scanning the issue]," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 566–569, March 2013.

[5] X. Huang, L. Zhang, and W. Gong, "Information fusion of aerial images and lidar data in urban areas: vector-stacking, re-classification and post-processing approaches," *International Journal of Remote Sensing*, vol. 32, no. 1, pp. 69–84, 2011.

[6] A. Avramović and V. Risojević, "Block-based semantic classification of high-resolution multispectral aerial images," *Signal, Image and Video Processing*, pp. 1–10, 2014.

[7] J. dos Santos, O. Penatti, P. Gosselin, A. Falcao, S. Philipp-Foliguet, and R. Torres, "Efficient and effective hierarchical feature propagation," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. PP, no. 99, pp. 1–12, 2014.

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

[9] S. Wager, S. Wang, and P. Liang, "Dropout training as adaptive regularization," in *Advances in Neural Information Processing Systems*, 2013, pp. 351–359.

[10] A. E. Robinson, P. S. Hammon, and V. R. de Sa, "Explaining brightness illusions using spatial filtering and local response normalization," *Vision research*, vol. 47, no. 12, pp. 1631–1644, 2007.

[11] A. M. Cheriyadat, "Unsupervised feature learning for aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 439–451, 2014.

[12] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2175–2184, April 2015.

[13] 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.

[14] 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*, Angers, France, May 2010, pp. 203–208.

[15] R. Bouchiha and K. Besbes, "Comparison of local descriptors for automatic remote sensing image registration," *Signal, Image and Video Processing*, vol. 9, no. 2, pp. 463–469, 2013.

[16] F. Faria, D. Pedronette, J. dos Santos, A. Rocha, and R. Torres, "Rank aggregation for pattern classifier selection in remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1103–1115, April 2014.

[17] P. Tokarczyk, J. Wegner, S. Walk, and K. Schindler, "Features, color spaces, and boosting: New insights on semantic classification of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 1, pp. 280–295, Jan 2015.

[18] A. Barsi and C. Heipke, "Artificial neural networks for the detection of road junctions in aerial images," *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, no. 3/W8, pp. 113–118, 2003.

[19] O. Firat, G. Can, and F. Yarman Vural, "Representation learning for contextual object and region detection in remote sensing," in *International Conference on Pattern Recognition*, Aug 2014, pp. 3708–3713.

[20] C. Hung, Z. Xu, and S. Sukkarieh, "Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a uav," *Remote Sensing*, vol. 6, no. 12, pp. 12 037–12 054, 2014.

[21] H. Xie, S. Wang, K. Liu, S. Lin, and B. Hou, "Multilayer feature learning for polarimetric synthetic radar data classification," in *IEEE International Geoscience & Remote Sensing Symposium*, July 2014, pp. 2818–2821.

[22] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised feature extraction of hyperspectral images," in *International Conference on Pattern Recognition*, 2014.

[23] M. E. Midhun, S. R. Nair, V. T. N. Prabhakar, and S. S. Kumar, "Deep model for classification of hyperspectral image using restricted boltzmann machine," in *International Conference on Interdisciplinary Advances in Applied Computing*, 2014, pp. 35:1–35:7.

[24] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2014.

[25] D. Tuia, R. Flamary, and N. Courty, "Multiclass feature learning for hyperspectral image classification: Sparse and hierarchical solutions," *{ISPRS} Journal of Photogrammetry and Remote Sensing*, no. 0, pp. –, 2015.

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[27] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[29] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS)*, 2010.

[30] 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 Workshops (CVPRW), 2015 IEEE Conference on*, 2015, pp. 44–51.

[31] 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.

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

[33] J. A. dos Santos, F. A. Faria, R. da S Torres, A. Rocha, P.-H. Gosselin, S. Philipp-Foliguet, and A. Falcao, "Descriptor correlation analysis for remote sensing image multi-scale classification," in *International Conference on Pattern Recognition*, Nov 2012, pp. 3078–3081.

[34] 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.

[35] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 437–478.