Land Use Classification in Remote Sensing Images by Convolutional Neural Networks

Marco Castelluccio, Giovanni Poggi, Carlo Sansone, Luisa Verdoliva

***Abstract*—We explore the use of convolutional neural networks for the semantic classification of remote sensing scenes. Two recently proposed architectures, CaffeNet and GoogLeNet, are adopted, with three different learning modalities. Besides conventional training from scratch, we resort to pre-trained networks that are only fine-tuned on the target data, so as to avoid overfitting problems and reduce design time. Experiments on two remote sensing datasets, with markedly different characteristics, testify on the effectiveness and wide applicability of the proposed solution, which guarantees a significant performance improvement over all state-of-the-art references.**

***Index Terms*—Convolutional neural networks, remote sensing, land use classification.**

I. Introduction

Thanks to the rapid progresses in remote sensing technology, and the reduction of acquisition costs, a large bulk of images of the Earth is readily available nowadays. They are taken from satellites or airplanes, with various imaging modalities, spatial and spectral resolutions, dynamic ranges.

With no shortage of data, the focus shifts on the ability to automatically extract valuable information from them. In recent years, there have been great advances in remote sensing image processing, for both low-level tasks, such as denoising or segmentation, and high level ones, such as classification. A plethora of land cover classification algorithms have been developed, with solid theoretical foundations, based on spectral and spatial properties of the pixels. However, the task becomes incrementally more difficult as the level of abstraction increases, going from pixels, to objects, and then scenes.

Labeling an image according to a set of semantic categories is the goal of scene classification. This is a very challenging problem, because land covers characterizing a given class may present a large variability and objects may appear at different scales and orientations. High intra-class variability then couples with low inter-class distance, a problem that grows ever more as finer classifications are sought. The same land covers and even the same objects can be found in images belonging to different classes. An example is shown in Fig.1 where the difference is made only by the density of buildings.

In this context, low-level features typical of pixel-based or object-based approaches [1], [2], [3], encoding spectral, textural, and geometrical properties, become mostly ineffective. More complex features and descriptors are necessary to capture the semantics of the scene. These have been the object of intense research efforts in the last few years, leading to good results in many fields. Nonetheless, even these sophisticated and successful descriptors are rapidly giving way to deep neural networks.

The Authors are with the DIETI, Università Federico II di Napoli, Naples, Italy. E-mail: mar.castelluccio@studenti.unina.it, {poggi, carlosan, verdoliv}@unina.it

(a) (b)

Fig. 1. Too close to call? Two similar images from the dense residential (a) and medium residential (b) classes of the UC-Merced dataset. Notice the different scales of observation.

Artificial neural networks take inspiration from models of the biological brain, and try to reproduce some of its functions by using simple but massively interconnected processing units, the neurons. A typical neural network architecture comprises several layers of neurons feeding one another, by which the "deep" attribute. Deep learning has provided impressive results in object recognition [4]. Recently, it has been also applied to remote sensing tasks [5] [6] [7], including land use image classification [8], showing always a great potential.

Considering the subtle differences among categories in scene classification, the superiority of deep learning with respect to "shallow" descriptors mentioned before can be easily claimed. While the latter aim at *reproducing the behavior* of the human interpreter, associating labels to images through a black box, deep neural networks try to *replicate cerebral mechanisms*, learning and combining internal descriptions of increasing levels of abstraction. That this be actually the case may be the object of endless controversies. Nonetheless, performance figures speak consistently in favor of deep learning for such tasks.

This work adds another piece of evidence in this sense. We use deep convolutional neural networks (CNNs or ConvNets) to tackle the remote sensing scene classification task. Two recently proposed promising architectures, CaffeNet [9] and GoogLeNet [10], are considered and tested. To cope with the scarcity of remote sensing training data, we explore various training modalities: not only the usual training from scratch but also the fine-tuning of pre-trained networks. Experiments on two publicly available remote sensing datasets, with widely different characteristics, prove CNNs to provide always an

excellent performance. On the well-known UC Merced Land Use dataset we obtain a gain of almost 3% with respect to the best reference, and almost 5% on the more recent Brazilian Coffee Scenes dataset. Note that CNNs had been already applied to remote sensing scene classification, in [8]. In that work, however, only the output of the penultimate layer of pre-trained networks is used, as a "shallow" image descriptor.

In the rest of the paper, after a thorough analysis of related work in the field (Section II), we provide the necessary background on ConvNets (Section III), describe how we applied them to the scene classification problem (Section IV), comment experimental results, also in comparison with reference techniques (Section V) and finally draw conclusions (Section VI).

II. Related work

In the last few years, there has been intense research on remote sensing scene classification, focusing both on the use of suitable image descriptors and of a proper classification task. Local descriptors, in fact, like local binary patterns (LBP) [11], scale-invariant feature transform (SIFT) [12], or histograms of oriented gradients (HOG) [13], with their invariance to geometric and photometric transformations, have proven effective in a variety of computer vision applications, especially object recognition. They can be extracted both in sparse (keypoint-based) and dense way. In any case, given the high dimensionality of the feature space, they need a subsequent coding phase in order to obtain an expressive but compact representation of the image.

The bag of visual words (BOVW) is a common and successful tool to reach this goal. In its basic version, k-means clustering is used to create off-line a dictionary of visual words. This is then used on-line to quantize the extracted features and associate with each one the label of the closest cluster centroid. Eventually, the histogram of such labels is fed to a classifier, typically a support vector machine (SVM). The SIFT-BOVW approach has been successfully applied in [14] to land use image classification for remote sensing applications.

The basic version of BOVW, however, neglects information on the spatial distribution of visual words. Hence, there have been several efforts in the literature to make up for this deficiency. One popular approach is the spatial pyramid match kernel (SPMK) proposed in [15] for object and scene categorization. It consists in partitioning the image at different levels of resolution and computing weighted histograms of the number of matches of local features at each level. Another alternative, considered in [16], is to perform a randomized spatial partition (RSP), aiming at a better characterization of the spatial layout of the images. These partition patterns are then weighted according to their discriminative abilities, and boosted into a robust classifier.

Note that SPMK considers only the absolute spatial arrangement of visual words. In order to capture both their absolute and relative spatial arrangements the spatial co-occurrence kernel (SCK) [14] and its pyramidal version (SPCK) [17] were proposed. Improved versions can be obtained by simply combining SCK and SPCK with the BOVW model (SCK+BOVW and SPCK+, respectively) or SPCK with SPMK (SPCK++) [14], [17]. The same goal of capturing both absolute and relative spatial relationships of local features is pursued in [18], where a pyramid-of-spatial-relatons (PSR) is proposed, which achieves higher robustness to rotations and translations. A major difference with respect to previous approaches is that SIFT features are evaluated densely and not only on interest points.

Dense SIFT features are used also in [19], and encoded in terms of a learnt basis functions to generate a new sparse representation for the feature descriptors. In [20], instead, histograms of dense SIFT features are matched through a fast approximation of the Earth movers distance. This helps exploring the relations among visual codes, which can be used as a key discriminative feature for image classification. Another way to improve the histogram-based feature extraction is proposed in [21], where histograms are regarded as Dirichlet-distributed probability mass functions, and then transformed through a Fisher kernel to enhance their discriminative power. Strong improvements also come from using more recent encoding methods as done in [22], where Fisher vectors (FV) [23], vectors of locally aggregated descriptors (VLAD) [24], and vectors of locally aggregated tensors (VLAT) [25] are used in combination with HOG and color features.

Another effective way to improve the classification performance is to augment the available dataset by adding rotated or flipped versions of the training images. For this reason [26] proposes Max-SIFT, a flipping invariant descriptor which is obtained from the maximum of a SIFT descriptor and its flipped copy.

Most of the features proposed in the current literature are extracted only from the gray-scale image. However, Yang and Newsam [14] showed that color histogram descriptors, evaluated on hue, lightness and saturation (Color-HLS), may provide a very good performance. The interactions among RGB color bands is exploited also in mCENTRIST [27], an extension of the CENTRIST algorithm [28], based in turn on LBP histograms and principal component analysis. LBP features are used also in [29], where the maximal conditional mutual information (MCMI) scheme is proposed to select an optimal subset of these features. In [30], instead, feature selection is performed separately for each class, with both one-versus-all and one-versus-one strategies.

Another trend is towards the combination of different features, pursued for example in [31], [32], [33]. In [32], in particular, a hierarchical scheme for multiple feature fusion (HMFF) is proposed. In order to capture structural, shape, textural and color characteristics, four different features are considered, and a two-step classification procedure is performed.

In [34] the use of high-level features is advocated for complex real-world scenes recognition. A visual parts-based method is proposed, inspired by [35], where several part detectors are used for objects or patterns at various orientations. Improvements of this method are presented in [36] and in [37]. High-level features are used also in [38] in the context of semisupervised feature learning. [19] resorts instead to fully unsupervised feature learning (UFL), and the same does [39], together with spectral clustering (UFL-SC).

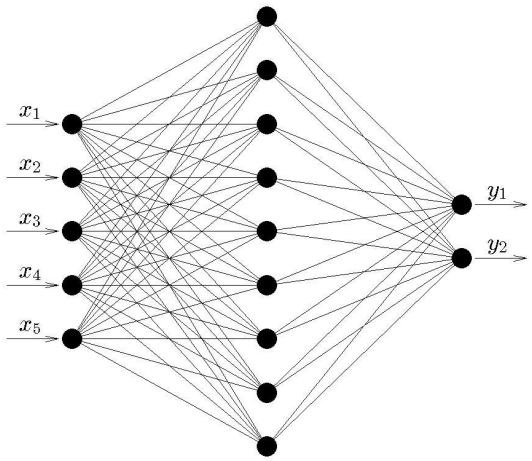

Fig. 2. An example three-layer Perceptron. This architecture may be used to make a binary decision on a 5-component input vector, using 9 features computed in the hidden layer. The same basic architecture, with 784, 30 and 10 neurons in the input, hidden and output layers, respectively, can be used to classify handwritten digits with accuracy exceeding 97%.

We conclude this review by mentioning the very recent work of [8], the only one, to the best of our knowledge, considering ConvNets for land use classification. As already said, however, CNNs are only used to produce shallow feature vectors for SVM classification, and no training on remote sensing data is carried out. Nonetheless, the performance is competitive with the previous state of the art.

III. Convolutional Neural Networks

The interest for convolutional neural networks (CNN) has been growing very fast, in the last few years, because of their impressive results [4] in a series of challenging problems involving image classification and retrieval [40]. CNN's evolve from the multilayer perceptron (MLP), proposed back in 1974 [41], with a number of technical solution that help solve its bottlenecks. Indeed, the MLP provides already excellent results in some classification problems. For example, a simple three-layer architecture similar to that of Fig.2, provides an accuracy beyond 97% in the recognition of the handwritten digits of the MNIST dataset [42].

The basic intuition behind these systems is that a processing architecture based on a large number of layered and massively interconnected simple units, may be more fit than sophisticated algorithms to tackle complex pattern recognition problems. The basic processing unit, the neuron, is indeed very simple, as shown in Fig.3. It computes the output activation by comparing the weighted sum of its input with a threshold and applying a suitable nonlinearity

$$o_j = \phi(\sum_i w_{ij} x_i - \theta_j) \quad (1)$$

Such configurable weights, w_{ij} are the core of the net, and they are learnt, typically through backpropagation [43], based on an adequately large set of labeled training examples.

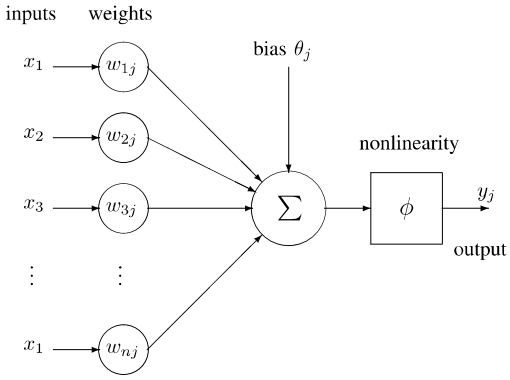

Fig. 3. The neuron, computes a nonlinear function of the weighted sum of the inputs plus a bias term.

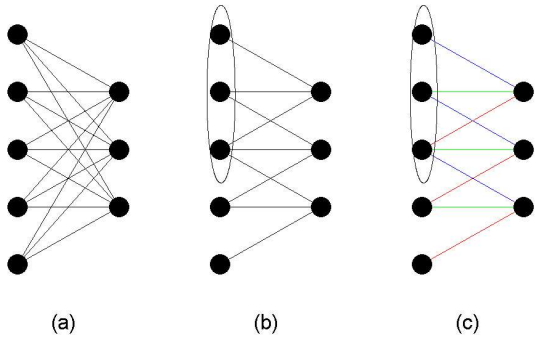

Fig. 4. From MLP to CNN. In MLP (a) all neurons of the second layer are fully connected with those of the first layer; with CNNs, neurons have a limited receptive field, see the oval in (b); moreover, all neurons of a layer share the same weights, see the color coding in (c). In this toy example, the number of free parameter to learn drops from 15 to 3.

The main problem with MLP is the very large number of free parameters (the weights) to be set. To solve challenging problems, such as image classification and retrieval, several layers of densely interconnected neurons must be considered. Learning all these weights would require a huge training set, and exceedingly large computational power. These problems are mostly solved by CNNs.

Convolutional neural networks were first proposed in 1980 by Fukushima [44] (called NeoCognitron) and then refined by LeCun [45]. Fig.4 provides some intuition into their major structural differences with respect to MLP. While in MLP (a) all neurons of layer $(n+1)$ are connected to all neurons of layer n, in CNN (b) neurons have limited "receptive fields". Moreover, all neurons of a layer are identical to one another, except for their receptive fields, sharing the same weights, color coded in (c). These constraints reduce sharply the number of free parameters to learn. As the name suggests, the $(n+1)$-th layer computes (before the nonlinearity) a spatial convolution of the outputs of the n-th layer. Therefore, it extracts some basic features of the image which are passed on to the next layer for further processing.

Besides being simpler than MLP, this architecture is much

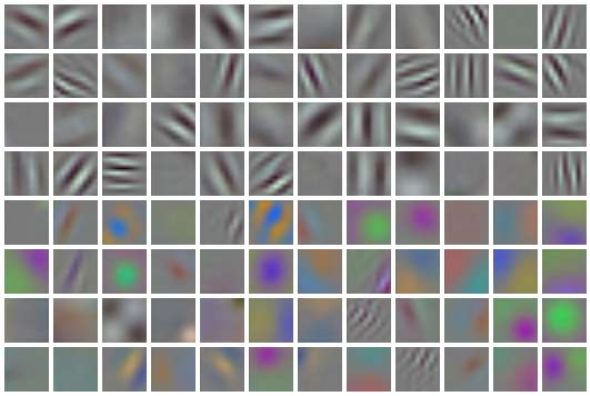

Fig. 5. The first-layer filters learned by CaffeNet over general purpose optical images. Filters appear to be sensitive to low-pass blobs and band-pass patterns of various frequencies and orientations, using both intensity and color information.

more similar to its biological template. In fact, according to the studies initiated by Hubel and Wiesel [46], the visual cortex is organized in layers composed by similar cells, with different receptive fields over the lower layer. In the lowest layer, elementary features are extracted, corresponding to bright spots, lines, corners, etc., which are then combined in higher layers so as to match more and more complex templates.

In actual CNNs, each layer comprises various sublayers of neurons operating in parallel on the previous layer, so as to extract a number of features at once, like a bank of filter does. An example is given in Fig.5, which shows the filters learned in the first layer of a CNN designed for image classification. Each filter elaborates the three input color bands, producing by convolution a corresponding feature map. In this case, therefore, 96 feature maps are produced as output of the second layer, which become the input of the next one.

Although CNNs have been introduced many years ago, only in the last few years implementing and training large CNNs has become possible due to both technology progresses and research findings. Major enabling factors have been the fast growth of affordable computing power, especially graphical processing units (GPUs), and the diffusion of large datasets of labeled images for training. On the other hand, research has contributed many new solution for faster and more reliable learning, like the rectified linear unit (ReLU) [47], [48], a neuron with a simplified non-linearity which allows much faster training, and techniques to reduce over-fitting, such as dropout [49], or data augmentation. The first CNN exploiting all these solutions, proposed in the seminal work of Krizhevsky *et al.* [4], improved image classification results by more than 10% w.r.t. the previous state of the art.

Today's CNN architectures comprise typically several layers, of various types.

1) Convolutional layers: described before, are the most important ones. They compute the convolution of the input image with the weights of the network. Neurons in the first hidden layer view only a small image window, and learn low-level features. Those in deeper layers view (indirectly) larger portions of the image, and are able to learn more expressive features by combining low-level ones. Each layer is characterized by a few hyper-parameters: the number of filters to learn, their spatial support, the stride between different windows and an optional zero-padding which controls the size of the layer output.

2) Pooling layers: reduce the size of the input layer through some local non-linear operations, for example $\max()$, so as to reduce the number of parameters to learn and provide some translation invariance. The most relevant hyper-parameters are the support of the pooling window and the stride between different windows.

3) Normalization layers: inspired by inhibition schemes present in the real neurons of the brain, aim at improving generalization. They are typically used with sigmoid neurons (not ReLU).

4) Fully-connected layers: are typically used as the last few layers of the network. By removing constraints, they can better summarize the information conveyed by lower-level layers in view of the final decision. Despite full connectivity, their complexity is still affordable thanks to the previous size-reducing layers.

IV. Using CNNs for remote sensing scene classification

In this work, we use Convolutional Neural Networks to carry out remote sensing scene classification. Several architectures have been already proposed and tested in the literature [50], especially for computer vision tasks, and most of them have been implemented and made available online. Here we focus on two very promising architectures, CaffeNet and GoogLeNet.

Caffe is one of the most popular libraries for deep learning (convolutional neural networks in particular). It is developed by the Berkeley Vision and Learning Center (BVLC) and community contributors. Caffe is easily customizable through configuration files, easily extendible with new layer types, and provides a very fast ConvNet implementation (leveraging GPUs, if present). It provides C++, Python and MATLAB APIs. Fig. 6 shows the specific architecture used in this work, the reference implementation from [9], which is in turn a modification of the network from [4] (the reader is referred to the original papers for more details). It comprises 5 convolutional layers, each followed by a pooling layer, and 3 fully-connected layers.

GoogLeNet, presented in [10], is the CNN architecture that won the ILSVRC14 competition. Its main peculiarity is the use of "inception modules", based on the "network in network" idea of [51], inspired also by theoretical results from [52]. In very concise terms, inception modules reduce the complexity of the expensive 3d filters of conventional architectures by means of a prior depth reduction phase. Thanks to the reduced complexity, multiple filters can be use in parallel at different resolutions, as shown in Fig. 7. To improve the effectiveness of the gradient backpropagation, given the depth of the network, GoogLeNet employs also auxiliary classifiers connected to intermediate layers. The Inception architecture has two main advantages: *i)* by employing filters of different sizes at each layer, it retains more accurate spatial information; moreover

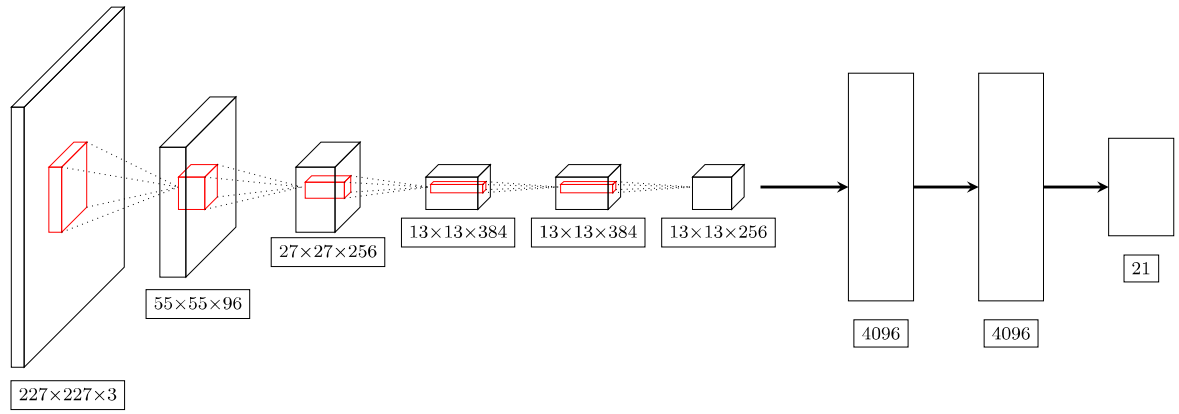

Fig. 6. The CaffeNet architecture used in this work. The boxes show the size of each feature layer and, for fully connected layers, the size of the output. Most receptive fields are 3×3, maxpool layers are not shown.

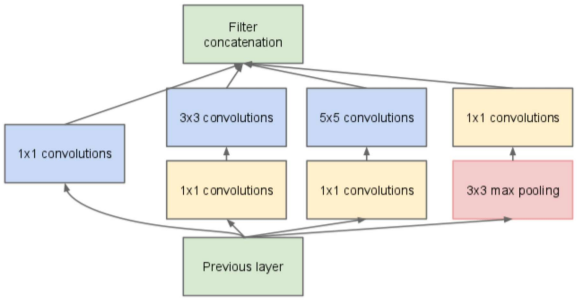

Fig. 7. The inception module; convolutions of different sizes allow the network to process features at different spatial scales. They are then aggregated and fed to the next layer. 1x1 convolutions are used for dimension reduction before the more expensive 3x3 and 5x5 convolutions.

ii) it significantly reduces the number of free parameters of the network, making it less prone to overfitting and allowing it to be deeper. The latter is a critical feature for performance, according to the findings from [53]. We refer the reader to the original paper by Szegedy *et al.* [10] for a detailed graphical illustration of the 22-layer GoogLeNet architecture with all relevant parameters.

These architectures have been developed to process natural images for computer vision applications. In that context, huge datasets of labeled images have been created and made available online, in particular Imagenet[1] [54], associated with a score of object classification challenges. When trying to use these CNNs on the available remote-sensing datasets, we run into the major problem of limited training data. In fact, even the largest datasets available in this field are, to date, far too small for correctly training a large CNN. The typical effect is overtraining, that is, the network works perfectly on the training data but does not generalize well to test data, providing eventually poor results.

This is a problem common to many other tasks where training data are hard to obtain. A possible solution, already explored in the literature, e.g. [55], [56], [57], is to use a pre-trained CNN and repurpose it to the task of interest.

[1]http://www.image-net.org/challenges/LSVRC/

Features learned in the lower layers of a CNN, in fact, like edges or color blobs, may be general enough to be useful for other classification tasks as well. Clearly, the success of this approach depends on several factors, the most important being the "distance" between the original task on which the CNN was originally trained and the target task. In particular, using CNNs trained on the Imagenet dataset makes full sense for UC-Merced data, since optical remote-sensing images have strong low-level similarities with general-purpose optical images. The same would not apply to SAR images, for example, due to their peculiar pixel-level statistics. In the experimental section we will also consider a borderline case, with a dataset of remote sensing images including an infrared band. As a further non-negligible advantage, the fine-tuning approach is usually faster than training the CNN from scratch, which may take days or weeks of computation time.

There are two major ways to adapt a pre-trained CNN

1) submitting the training images to the CNN, and regarding the output of the penultimate layer as feature vectors, used to train an off-line classifier;
2) use the training images to "fine-tune" the CNN for the task of interest.

The first solution is straightforward and does not require any further effort, except for the design of a classifier. It has been already considered for the classification of aerial images [8], with good results. Nonetheless, the second solution is certainly more promising, as it allows a deeper adaptation to the data of interest, exploiting the full potential of CNNs. With fine-tuning, one needs to decide which layers of the original network must be freezed, and which ones are instead allowed to keep learning, and at which rate. These choices impact on both accuracy and design time and, as before, depend very much on the similarity between the original and target problem, and on the amount of training data available. Typically, the first few layers are freezed, because low-level features can better fit different problems.

In summary, in this work, we consider and compare three options:

- *training from scratch:* the whole convolutional net is trained on the available target data;

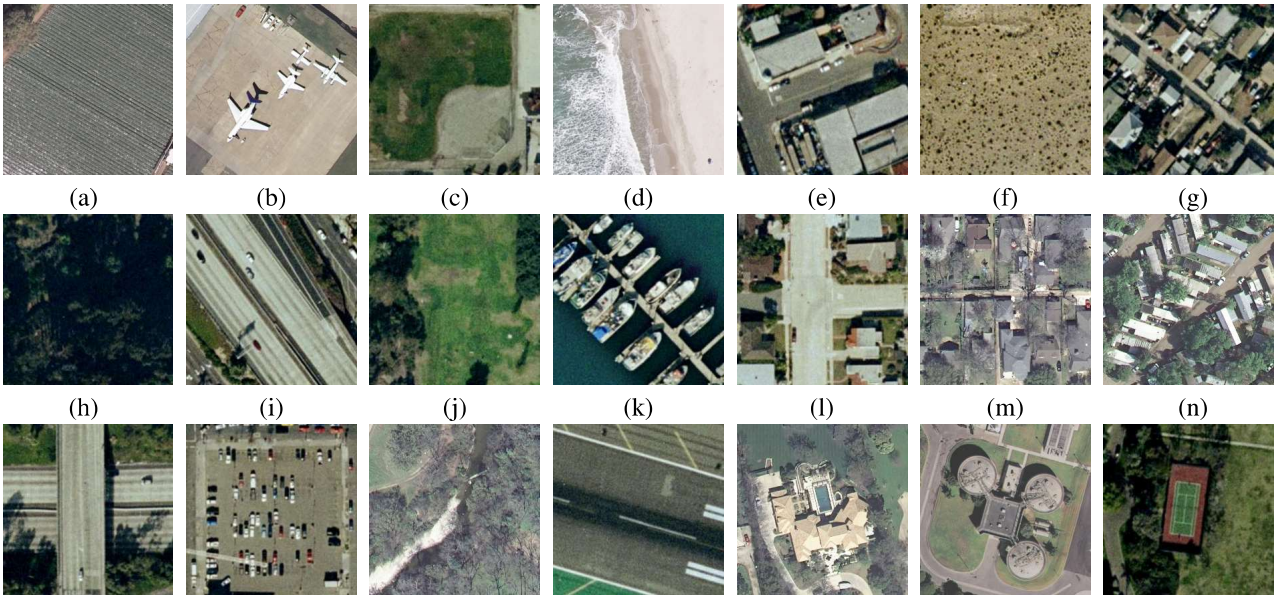

Fig. 8. Class representatives of the UC-Merced dataset. (a) agricultural; (b) airplane; (c) baseball diamond; (d) beach; (e) buildings; (f) chaparral; (g) dense residential; (h) forest; (i) freeway; (j) golf course; (k) harbor; (l) intersection; (m) medium residential; (n) mobile home park; (o) overpass; (p) parking lot; (q) river; (r) runway; (s) sparse residential; (t) storage tanks; (u) tennis court;

- *fine tuning:* a pre-trained net is used, adapting only a certain number of high-level layers;
- *feature vector:* the output of the penultimate layer of a pre-trained net is used as a feature vector for classification.

Note, however, that, differently from [8], we implement the last option by replacing the last layer of the network with a different fully-connected layer, with 21 outputs instead of 1000, followed by a Softmax classifier instead of an SVM. By so doing, we use only CNNs for all our experiments.

V. Experimental results

We carried out a number of experiments to assess the performance of the proposed approach, also in comparison with state-of-the-art results. We use two remote-sensing datasets. The well-known UC Merced Land Use dataset [14] (UC-Merced for short), includes aerial optical images, with low-level characteristics similar to those of the Imagenet. In recent years, many researchers have used this dataset, allowing for an extensive comparison of results with the literature. The Brazilian Coffee Scenes dataset [8], instead, includes satellite images with an infra-red band, hence less similar to general purpose images. Since it has been published very recently, more limited results are available, including however results with CNNs. In the next two subsections we discuss results separately for the two datasets

Experiments have been all carried out on a notebook equipped with an NVIDIA GeForce GT 750M 2048 MB GPU. For each training modality and dataset, the number of training iterations has been established by preliminary experiments. In fine-tuning modality, all learning rates are equal except for the first layer's, set to one tenth of the others. In feature vector modality, only the last fully connected layer is trained.

CNN	Design	Iterations	Accuracy
CaffeNet	from scratch	100,000	85.71
	fine-tuning	20,000	95.48
	feature vector	5,000	94.28
GoogLeNet	from scratch	100,000	92.86
	fine-tuning	20,000	**97.10**
	feature vector	5,000	94.38

TABLE I
Classification accuracy (%) of proposed CNN-based solutions on the UC-Merced dataset. Best result in bold.

In all cases we used a moderate data augmentation, through mirroring and random cropping, to increase the effective training set size.

A. UC-Merced

This dataset[2], released in 2010 [14], is extracted from large optical images (RGB color space) of the US Geological Survey, taken over various regions of the United States. 2100 256×256-pixel images are selected and manually labeled as belonging to 21 land use classes, 100 for each class. Fig.8 shows one example image for each class. More examples and more information are available in the original paper [14]. Due to their nature and relatively high resolution (30 cm) these images share many low-level features with general-purpose optical images making them good candidates for fine-tuning a pre-trained CNN.

[2]http://vision.ucmerced.edu/datasets/landuse.html

Method	Ref.	Year	Accuracy	Approach
BOVW			76.81	Keypoint SIFT and Bag-of-Visual-Words
SPMK	[15]	2006	75.29	Keypoint SIFT and Spatial Pyramid Match Kernel
Color-HLS	[14]	2010	81.19	Hue, Lightness, Saturation and Bag-of-Visual-Words
SCK	[14]	2010	72.52	Keypoint SIFT and Spatial Co-occurrence Kernel
BOVW+SCK	[14]	2010	77.71	Keypoint SIFT, Spatial Co-occurrence Kernel, and Bag-of-Visual-Words
SPCK	[17]	2011	73.14	Keypoint SIFT and Spatial Pyramid Co-occurrence Kernel
SPCK+	[17]	2011	76.05	Keypoint SIFT, Spatial Pyramid Co-occurrence Kernel, and Bag-of-Visual-Words
SPCK++	[17]	2011	77.38	Keypoint SIFT, Spatial Pyramid Match Kernel, and Spatial Pyramid Co-occurrence Kernel
BRSP	[16]	2012	77.80	Randomized Spatial Partition via Boosting
HMFF	[32]	2013	92.38	Multi-Feature Fusion and Hierarchical classifier
Dirichlet	[21]	2014	92.80	SIFT and Dirichlet-based Histogram Feature Transform
UFL	[19]	2014	81.67	dense SIFT and Unsupervised Feature Learning
mCENTRIST	[27]	2014	89.90	LBP on RGB + PCA
FV	[22]	2014	93.80	HOG + RGB and Fisher Vectors
VLAD	[22]	2014	92.50	HOG + RGB and Vectors of Locally Aggregated Descriptors
VLAT	[22]	2014	94.30	HOG + RGB and Vectors of Locally Aggregated Tensors
COPD	[34]	2014	91.33	Collection Of Part Detectors
Partlets	[36]	2015	91.33	Partlets
Sparselets	[37]	2015	91.46	Sparselets
MCMI-based	[29]	2015	88.20	LBP and feature selection based on incremental Maximal-Conditional-Mutual-Information
PSR	[18]	2015	89.10	dense SIFT, Pyramid-of-Spatial-Relatons, and Bag-of-Visual-Words
UFL-SC	[39]	2015	90.26	Unsupervised Feature Learning with Spectral Clustering and Bag-of-Visual-Words
CNN	[8]	2015	93.42	pre-trained ConvNet (CaffeNet) with SVM classifier
proposed		2015	**97.10**	pre-trained ConvNet (GoogLeNet) with fine-tuning on target data

TABLE II
CLASSIFICATION ACCURACY (%) OF REFERENCE AND PROPOSED METHODS ON THE UC-MERCED DATASET. BEST RESULT IN BOLD.

In Table I we report synthetic results for the two CNN architectures and the three design approaches considered. Results are always computed through five-fold validation, by averaging over the five folds. The first observation is that the fine-tuning approach, as expected, provides the best results with both CaffeNet and GoogLeNet, reaching an overall accuracy of 95.48% and 97.10%, respectively. This is about 10% and 5% better, respectively, than the design from scratch, confirming the limited value of this latter option when training data are limited. As for the feature vector approach, pretty good results are obtained with both architectures, but clearly inferior to those of the fine-tuning approach, with a gap of 1-3%.

In terms of computation time, the training from scratch is also much more demanding. Through preliminary experiments, we decided the number of training iterations to use for each modality, 100,000 for the training from scratch, much less for the fine tuning (20,000), and still less for the feature vector case (5,000). These are not marginal differences, for such computation-intensive experiments. Notice, however, that the fine tuning approach provides a very good performance already at 5,000 iterations, 95.12% and 96.48% respectively, hence it is preferable anyway to the feature vector approach.

Turning to the comparison between the two CNN architectures, GoogLeNet appears to provide consistently the best performance, as suggested by the literature, while being slightly less demanding in terms of computation. In the following, when talking of the proposed approach, we will hence refer to GoogLeNet with fine tuning over 20,000 iterations.

Several approaches have been proposed recently for remote sensing scene classification, and most of them have been tested on the UC-Merced dataset, following the same experimental protocol, with 5-fold cross validation. Therefore there is plenty of data available for a solid comparison with the state of the art. In Table II we report the overall accuracies for all these comparable methods, as they appear in the original papers, together with the accuracy of our best CNN solution. The proposed method guarantees a large performance gain w.r.t. to all references, with a minimum gap of almost 3%. This applies also to [8] which uses feature vectors extracted from CaffeNet, with pre-training on Imagenet. Notice that, in the very same conditions, we obtain somewhat better results (see Table I) probably because of the different classifier.

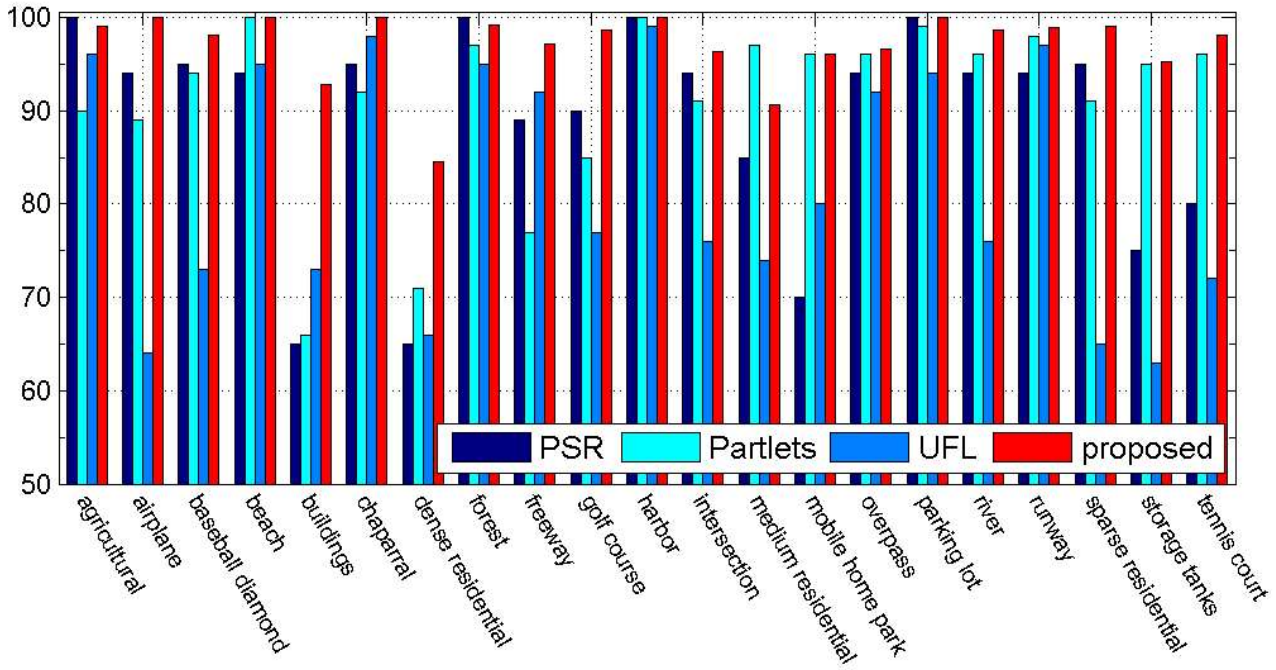

Fig. 9. Per-class accuracies of the proposed method and some state-of-the-art references on the UC-Merced dataset.

In Fig.9 we show the per-class accuracies provided by the proposed method and some selected references, PSR [18], Partlets [36], and UFL [19], using again the results reported in the original papers. The proposed method provides the best performance almost uniformly over all classes, often with perfect or near-perfect accuracy. The worst result (84.5% accuracy) is observed for the dense residential class which, as already observed, suffers the presence of other very close classes, like medium residential and mobile home park. In any case, for the same class, reference methods perform more than 10% worse.

Finally, in Fig.10, we show all the images of fold #1 (only one fold, to save space) that have been wrongly classified with the proposed method. Only 8 images out of a total of 420 have been misclassified, in this fold, all belonging to classes that have some very "close" neighbors, as noted commenting the bar graph of Fig.9. A correct classification of some of these images may be difficult also for a human photointerpreter. It is worth underlining, however, that a suitable fusion of the outputs of CaffeNet and GoogLeNet would remove virtually all these errors, as also observed in [8].

Fig. 10. Classification errors in fold #1 of UC-Merced by the proposed method. (a) buildings classified as (→)storage tanks; (b) dense residential → mobile home park; (c) freeway → runway; (d) medium residential → dense residential; (e) medium residential → sparse residential; (f) medium residential → dense residential; (g) mobile home park → intersection; (h) river → golf course;

B. Brazilian Coffee Scenes

This dataset[3], released in 2015 [8], includes scenes taken by the SPOT sensor in the green, red, and near-infrared bands, over four counties in the State of Minas Gerais, Brazil. The scenes are partitioned in over 50,000 64×64-pixel tiles, labeled as coffee (1,438) non-coffee (36,577) or mixed (12,989). To provide a balanced dataset, 1,438 tiles of both coffee and

[3]www.patreo.dcc.ufmg.br/downloads/brazilian-coffee-dataset/

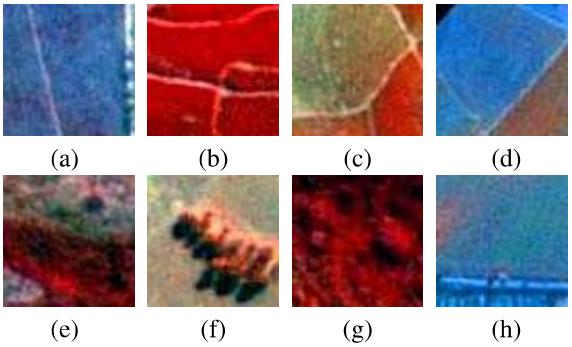

Fig. 11. Class representatives of the Brazilian Coffee Scenes dataset. (a)-(d) coffee tiles; (e)-(h) non-coffee tiles.

CNN	Design	Iterations	Accuracy
CaffeNet	from scratch	20,000	90.17
	fine-tuning	10,000	90.94
	feature vector	5,000	85.02
GoogLeNet	from scratch	20,000	**91.83**
	fine-tuning	10,000	90.75
	feature vector	5,000	84.02

TABLE III
CLASSIFICATION ACCURACY (%) OF PROPOSED CNN-BASED SOLUTIONS ON THE BRAZILIAN COFFEE SCENES DATASET. BEST RESULT IN BOLD.

non-coffee classes are kept and organized in 5 almost equally sized folds, while mixed tiles are all discarded. Fig.11 shows four tiles each of the coffee and non-coffee classes, obviously in false colors. Overall, this dataset is very different from UC-Merced. The images are not optical (green-red-infrared instead of red-green-blue), hence intrinsically different from the Imagenet samples used for pre-training. Moreover, there is a much larger number of samples for each class, which could make the design from scratch a more interesting option for CNN. In this case, however, there is no reference result for a comparative assessment of performance, except for those reported in [8].

Table III shows the results obtained with the proposed techniques. The most notable difference w.r.t. the UC-Merced case is the relatively poor performance of the feature vector approach. This can be attributed to the more marked differences w.r.t. Imagenet dataset used to pre-train the network. In fact, training from scratch provides a much better performance, in this case, probably due also to the larger number of samples available for each class. The optimal number of training iterations is much smaller than before, 20,000, and results almost equally good are obtained also at 10,000 iterations. Similar considerations apply to the fine-tuning case, where the best results are obtained at 10,000 iterations. The overall best, almost 92%, is provided by GoogLeNet with training from scratch. In general, results are significantly worse than with UC-Merced, despite the 2-class vs. 21-class problem. Indeed, as the Authors of [8] note, this is a rather challenging dataset, due to a large intra-class variability *"caused by different crop management techniques, different plant ages and/or spectral distortions and shadows"*. In any case, this result is almost 5% better than the top result reported in [8], see Tab.IV, obtained with BIC (Border-Interior Pixel Classification) a simple color descriptor, and 7% better than CNNs with the feature vector approach, for the mentioned reasons.

Method	Accuracy	Approach
BIC	87.0	color descriptors
BOVW	80.5	BOVW with dense SIFT features
CNN-1	84.8	pre-trained CaffeNet + SVM
CNN-2	81.2	pre-trained OverFeat + SVM
proposed	**91.8**	GoogLeNet trained from scratch

TABLE IV
CLASSIFICATION ACCURACY (%) OF REFERENCE AND PROPOSED METHODS ON THE BRAZILIAN COFFEE SCENE DATASET. ALL REFERENCE DATA FROM [8]. BEST RESULT IN BOLD.

VI. CONCLUSIONS

We have addressed the remote sensing scene classification task by resorting to convolutional neural networks. Two promising architectures have been considered with three design modalities. Experiments on two datasets with quite different properties have provided insightful information.

As expected, training a deep CNN from scratch is not always advisable with the limited-size datasets currently available in this field. A valid alternative consists in using pre-trained CNN and adapting it to the target task. However, a shallow adaptation, with the CNN used only as a feature vector generator for subsequent classification, gives up much of the potential of this approach. On the contrary, a careful fine-tuning, involving several layers of the architecture, provides very good results, in general. Overall, the experimental evidence is definitely encouraging. On the widespread UC-Merced dataset, the proposed method outperforms the best references technique by almost 3%. Moreover, it provides the best performance, by a wide margin, also on the more recent Brazilian Coffee Scenes dataset. This latter dataset allowed us to study also the behavior of this approach when pre-training and target data differ significantly.

The near-perfect performance on aerial images makes clear that the next big challenge is related to the classification of data acquired with other imaging modalities. The Synthetic Aperture Radar, therefore, is certainly a field of great interest for future research, drawing already the attention of several research groups [58], [59]. A recent work in this direction is [60], addressing automatic target recognition in SAR images.

REFERENCES

[1] M. Pesaresi and A. Gerhardinger, "Improved textural built-up presence index for automatic recognition of human settlements in arid regions with scattered vegetation," *IEEE Journal of Selected Topics on Earth Observation and Remote Sensing*, vol. 4, no. 1, pp. 16–26, march 2011.

[2] I.A. Rizvi and B.K. Mohan, "Object-based image analysis of high-resolution satellite images using modified cloud basis function neural network and probabilistic relaxation labeling process," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 12, pp. 4815–4820, december 2011.

[3] R. Gaetano, G. Masi, G. Poggi, L. Verdoliva, and G. Scarpa, "Marker-controlled watershed-based segmentation of multiresolution remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 2987–3004, june 2015.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Conference on Neural Information Processing Systems*, 2012, p. 10971105.

[5] 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 (ICONIAAC)*, 2014, pp. 35:1–35:7.

[6] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, in press 2015.

[7] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *International Conference on Image Processing*, 2014, pp. 5132–5136.

[8] 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 *IEEE Computer Vision and Pattern Recognition Workshops*, 2015, pp. 44–51.

[9] 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.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[11] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, july 2002.

[12] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005, p. 886893.

[14] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *International Conference on Advances in Geographic Information Systems*, 2010, pp. 270–279.

[15] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169–2178.

[16] Y. Jiang, J. Yuan, and G. Yu, "Randomized spatial partition for scene recognition," in *European Conference on Computer Vision*, 2012, pp. 730–743.

[17] Y. Yang and S. Newsam, "Spatial pyramid co-occurrence for image classification," in *IEEE International Conference on Computer Vision*, 2011, pp. 1465–1472.

[18] S. Chen and Y. Tian, "Pyramid of spatial relatons for scene-level land use classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 1947–1957, april 2015.

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

[20] Y. Zhang, X. Sun, H. Wang, and K. Fu, "High-resolution remote-sensing image classification via an approximate earth movers distance-based bag-of-features model," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 1055–1059, september 2013.

[21] T. Kobayashi, "Dirichlet-based histogram feature transform for image classification," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4321–4328.

[22] R. Negrel, D. Picard, and P.-H. Gosselin, "Evaluation of Second-order Visual Features for Land-Use Classification," in *International Workshop on Content-Based Multimedia Indexing*, 2014, pp. 1–5.

[23] F. Perronnin, J. Sanchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European Conference on Computer Vision (ECCV)*, 2010, pp. 143–156.

[24] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, january 2011.

[25] D. Picard and P. Gosselin, "Efficient image signatures and similarities using tensor products of local descriptors," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 680–687, june 2011.

[26] L. Xie, Q. Tian, J. Wang, and B. Zhang, "Image classification with Max-SIFT descriptors," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.

[27] Y. Xiao, J. Wu, and J. Yuan, "mCENTRIST: A multi-channel feature generation mechanism for scene categorization," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 823–836, 2014.

[28] J. Wu and J. M. Rehg, "CENTRIST: A visual descriptor for scene categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1489–1501, august 2011.

[29] J. Ren, X. Jiang, and J. Yuan, "Learning LBP structure by maximizing the conditional mutual information," *Pattern Recognition*, vol. 48, pp. 3180–3190, 2015.

[30] X. Chen, T. Fang, H. Huo, and D. Li, "Measuring the effectiveness of various features for thematic information extraction from very high resolution remote sensing imagery," *IEEE Transactions on Geoscience and Remote*, vol. 53, no. 9, pp. 4837–4851, september 2015.

[31] Vladimir Risojević and Zdenka Babić, "Fusion of global and local descriptors for remote sensing image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 4, pp. 836–840, july 2013.

[32] W. Shao, W. Yang, G.-S. Xia, and G. Liu, "A hierarchical scheme of multiple feature fusion for high-resolution satellite scene categorization," in *Computer Vision Systems, LNCS*, 2013, vol. 7963, pp. 324–333.

[33] Y. Zhong, Q. Zhu, and L. Zhang, "Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, in press 2015.

[34] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 98, pp. 119–132, 2014.

[35] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, september 2010.

[36] G. Cheng, J. Han, L. Guo, Z. Liu, S. Bu, and J. Ren, "Effective and efficient midlevel visual elements-oriented land-use classification using VHR remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4238–4249, august 2015.

[37] G. Cheng, J. Han, L. Guo, and T. Liu, "Learning coarse-to-fine sparselets for efficient object detection and scene classification," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1173–1181.

[38] W. Yang, X. Yin, and G.-S. Xia, "Learning High-level Features for Satellite Image Classification With Limited Labeled Samples," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4472–4482, august 2015.

[39] F. Hu, G.-S. Xia, Z. Wang, X. Huang, L. Zhang, and H. Sun, "Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 5, pp. 2015–2030, may 2015.

[40] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, may 2015.

[41] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D. thesis, Harvard University, 1974.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[43] P.J. Werbos, "Backpropagation through time: what it does and how to do it," *Proc. of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[44] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mecha- nism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[45] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[46] D.H.Hubel and T.N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[47] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *International Conference on Computer Vision*, 2009, pp. 2146–2153.

[48] V. Nair and G.E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning*, 2010.

[49] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, july 2012.

[50] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.

[51] M. Lin, Q. Chen, and S. Yan, "Network in network," *http://arxiv.org/pdf/1312.4400v3.pdf*, 2014.

[52] S. Arora, A. Bhaskara, R. Ge, and T. Ma, "Provable bounds for learning some deep representations," in *International Conference on Machine Learning*, 2014, pp. 1–9.

[53] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1573–1585, august 2014.

[54] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, april 2015.

[55] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[56] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.

[57] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems 27 (NIPS '14)*, Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, Eds. 2014, pp. 3320–3328, Curran Associates, Inc.

[58] A.A. Popescu, I. Gavat, and M. Datcu, "Contextual Descriptors for Scene Classes in Very High Resolution SAR Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 1, pp. 80–84, january 2012.

[59] R. Bahmanyar, S. Cui, and M. Datcu, "A comparative study of Bag-of-Words and Bag-of-Topics Models of EO image patches," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 6, pp. 1357–1361, june 2015.

[60] H. Wang, S. Chen, F. Xu, and Y.-Q. Jin, "Application of Deep Learning algorithms to MSTAR data," in *IEEE International Geoscience and Remote Sensing Symposium*, 2015, pp. 3743–3745.