An Efficient Deep Representation Based Framework for Large-Scale Terrain Classification

Yupeng Yan, *Anand Rangarajan* and *Sanjay Ranka*
University of Florida, Gainesville, Florida, USA
yanyp@ufl.edu, {anand, ranka}@cise.ufl.edu

***Abstract*—In this paper, we present a novel terrain classification framework for large-scale remote sensing images. A well-performing multi-scale superpixel tessellation based segmentation approach is employed to generate homogeneous and irregularly shaped regions, and a transfer learning technique is sequentially deployed to derive representative deep features by utilizing successful pre-trained convolutional neural network (CNN) models. This design is aimed to overcome the big problem of lacking available ground-truth data and to increase the generalization power of the multi-pixel descriptor. In the subsequent classification step, we train a fast and robust support vector machine (SVM) to assign the pixel-level labels. Its maximum-margin property can be easily combined with a graph Laplacian propagation approach. Moreover, we analyze the advantages of applying a feature selection technique to the deep CNN features which are extracted by transfer learning. In the experiments, we evaluate the whole framework based on different geographical types. Compared with other region-based classification methods, the results show that our framework can obtain state-of-the-art performance w.r.t. both classification accuracy and computational efficiency.**

***Keywords*—remote sensing, superpixel segmentation, convolutional neural network, transfer learning, feature selection, semi-supervised learning**

I. Introduction

Efficient and accurate labeling of very high resolution (VHR) imagery is an important application in machine learning and data science. There are several factors making this problem interesting and challenging. First, the samples to be classified are attached to a grid. This requires the spatial aspect of the data to be directly addressed. Second, there is no straightforward demarcation of training and testing samples. In this application, the expert labels a few locations in the image as belonging to one of the classes but it is not straightforward to determine whether the expert intended to label a single pixel or a homogeneous region. In addition, the entire image is available during training allowing us to operate with a semi-supervised learning mindset. Third, the very high resolution nature of the images forces us to consider more efficient labeling schemes: for example, we contend that the labeling of superpixels (*i.e.* homogeneous and irregularly shaped regions) is more efficient. Finally, we need to take into account previous efforts at training on similar imagery. Consequently, transfer learning must be integrated with superpixellization.

Previous efforts at high resolution image labeling have in the main not comprehensively addressed all of these issues at the same time. We are not aware of any previous effort that (i) uses a state of the art method for superpixel segmentation in order to improve efficiency in labeling, (ii) deploys deep learning filters (in transfer learning mode) to obtain better discriminative features, (iii) performs semi-supervised learning using support vector machines thereby leveraging a few labeled exemplars to efficiently label a very high resolution image.

In summary, we present a very efficient semi-supervised image labeling algorithm which integrates state of the art segmentation and deep learning (in transfer learning mode). In contrast to other methods, we label superpixels instead of pixels and demonstrate that this approach is superior in terms of accuracy than competing methods. Further, we clearly show the importance of having a state of the art superpixel segmentation method by facilitating comparisons with lesser superpixel estimation approaches. Transfer learning is deployed using non-overlapping patches to obtain better and more discriminative features for classification. Previous approaches do not reach the size and scale of the problems addressed here: very high resolution imagery requires a rethink of many fundamental issues so that scalability is obtained. We hope to convince the reader that the integrated superpixel segmentation and transfer learning driven classifier presented here is the way forward in very high resolution imagery applications.

II. Related Work

Many existing remote sensing classification frameworks are built upon effective visual descriptors. Traditional low-level features like the color histogram , histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT) play an important role in detecting and distinguishing the objects. Middle-level descriptors such as bag of visual words (BoVW) and semantic-spatial matching (SSM) [1] show their promising performance in scene categorization. Taking advantage of different properties, a reasonable combination of these features with appropriate weights can achieve very good results in many real-world applications [2], [3], [4], [5], [6].

Nevertheless, we should note that the inter-pixel relationships are not fully exploited in these earlier studies. Rather than manually or heuristically extracting the effective features, researchers have designed a variety of deep learning architectures to automatically learn the representative features from the images. Maggiori *et al.* [7] proposed an end-to-end framework for densely labeling large-scale satellite data. A two-step training approach is accomplished by initializing the CNN with a large amount of raw data and then refining the resulting networks with a small portion of the labeled image. Chen *et al.* [8] presented a regularized deep feature extraction framework for hyperspectral images. Their architectures based on a 3-D CNN can extract the spectral, spatial and spectral-spatial features and thus effectively improve the classification accuracies with the help of virtual training samples. Indeed, they have attempted different methods to alleviate the problem of lacking rich label information; however, these methods still cannot avoid the potential problem of overfitting when training the deep networks without more available labeled data. To this end, [9] was the first to use the pre-trained CNN models (from the everyday object recognition database) for aerial and remote

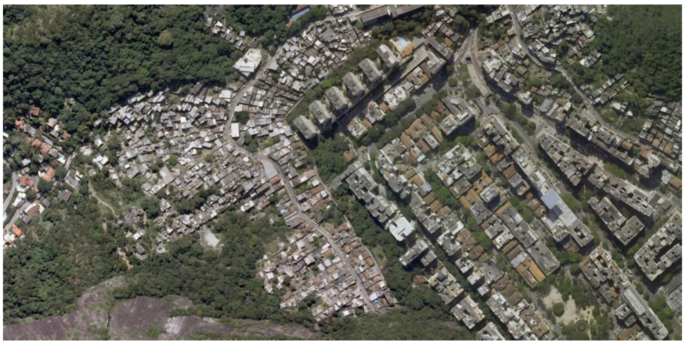

Fig. 1. An example of the VHR satellite image captured from Rio, Brazil.

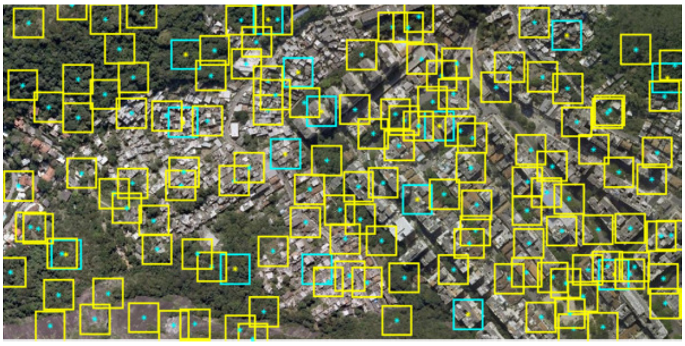

Fig. 2. Illustration of the rectangular patches created for deep superpixel feature extraction. The cyan windows around yellow crosses (*i.e.* expert labeled points) represent the patches that are used to train the classifier. The yellow windows around cyan stars (*i.e.* centers of the superpixels) represent the patches that are used to receive the superpixel-level label.

sensing classification. The strong generalization power of the deep features allow them to obtain the best results on their datasets, and the results are even better when fusing the deep feature with other descriptors.

On the other hand, all of these above-mentioned methods adopt the pixel-based strategy. This means that they either treat the whole image or the patches centered at each pixel as the input to their frameworks, so the actual computation cost can be very expensive especially if the test image size in the latter case becomes significantly large. Accordingly, the development of a region-based classification approach is necessary. A Gaussian Multiple Instance Learning (GMIL) approach [10] was proposed to capture complex spatial patterns, where groups of contiguous pixels were modeled by a Gaussian distribution. But, the appropriate block size has to be determined by multiple experiments and this can be a bottleneck for global-scale application. Instead of arbitrarily partitioning the image into blocks, Zhang *et al.* [11] created a superpixel-based graphical model to obtain the contextual information and spatial dependence. A watershed algorithm was employed to generate the superpixel. However, this can cause oversegmentation due to the noise and even lead to the loss of salient object contours. The final classification result may be badly affected by the lack of a state of the art superpixel segmentation approach.

To overcome these problems, we integrate the approaches of superpixel-based segmentation and the generation of deep features by transfer learning into our framework. It fully considers the computational efficiency and meanwhile achieves state-of-the-art classification performance. We provide the details in the following sections.

III. Large-Scale Labeling Framework

In this paper, we aim to label all the pixels in large-scale VHR images with a very small fraction of ground-truth data. Typically, the candidate categories include various human settlement and natural objects like buildings, wasteland, trees, *etc.* (and please see an example in Fig. 1). To fulfill such requirements, we propose an efficient CNN-based classification framework and describe the the main steps in the following subsections.

A. Superpixel Tessellation

The Ultrametric Contour Map (UCM) [12] is a popular method to produce a hierarchy of tessellations at different scales. In our framework, we only use the fine-scale tessellation to generate the superpixels of a suitable size. It is also feasible to move the tessellation to a coarser level if we seek to derive deep features from multiple scales.

Basically, there are six major steps in UCM superpixel estimation: (i) extracting scale-space features from brightness, texture and wavelength channels, (ii) constructing a weighted graph where the edge links pixels within a certain distance, (iii) computing the top K eigenvectors from the weighted graph, (iv) obtaining the complementary spectral information from the eigenvectors, (v) combining the local and spectral information linearly, and (vi) applying an oriented watershed transform (OWT) [12] to extract global closed contours.

Note that this UCM approach (denoted as OWT-UCM in Section IV) combines local and global contour information by obtaining cues from both the original image and the eigenvector images. This perceptual grouping strategy is mainly responsible for obtaining a good superpixel segmentation. We can see the resulting tessellation of Fig. 1 in Fig. 3, where the areas of significant (or less) variation are captured by smaller (or larger) superpixels.

B. Deep Superpixel Feature Extraction

After superpixel segmentation is in place, we describe each superpixel with multi-pixel features. In recent years, following the explosive development of deep learning architectures, we have seen state-of-the-art performance in computer vision tasks [13], [14]. Deep convolutional neural networks (CNN) can learn the most distinguishing features from pixel granularity and find a good way to classify these high-level features.

As demonstrated in [9], deep CNN features from everyday objects can be generalized to the remote sensing domain. Considering the fact that only a small number of the pixels can be labeled by the experts when taking into account the large-scale underlying image data, it is impractical to build our own remote-sensing model. But fortunately, it is still suitable for us to obtain the deep representative features by a transfer learning method, by removing the last fully-connected layers of a pre-trained CNN model and then treating the remaining network as a fixed feature extractor as in [9], [15].

Note that, before processing the pixel-level information with the CNN feature extractor, we need to select the most appropriate input patches for the superpixels. The patches should preserve the necessary side information such as color, texture, and shape of objects as much as possible, so they can fully represent the superpixels belonging to the same category. To this end, we obtain such patches by cropping the center of the superpixels as shown in Fig. 2. The patch size is chosen based on the average size of the obtained superpixels, so we can learn the common structure from the neighborhoods if the superpixel is too small (*e.g.* slum or urban), and discard the visually repeating areas if the superpixel is too large (*e.g.* forest and sea).

After obtaining the final output vectors from the extractor, we can use them as the deep CNN features of the superpixels for the subsequent classification step. Other global features can also be concatenated with the output vectors in order to increase the discriminating power.

C. Superpixel Label Prediction

In order to maximize the usage of limited ground-truth points (or pixels), we extract the features by cropping the patches centered at the training points (see bounding boxes around cyan stars in Fig. 2). This is quite different from the work of Sethi *et al.* [5] since they train the classifier with the features obtained from the superpixels which the training points fall in—several training points belonging to the same superpixel may be repeatedly used and thus some distinguishing information (not relying on the segmentation) is probably lost.

We assign the label to each superpixel based on the probability scores produced by our classifier. Then these superpixel-wise labels are percolated down through the segmentation hierarchy, which means all the pixels contained in a superpixel are assigned to the same label. In this way, we can avoid directly labeling the VHR images so as to improve the computational efficiency.

Additionally, as we mentioned earlier, in this machine learning set up, the test data is present with the training samples in the form of the entire image, so it is always available at the same time when we train the classifier. We can therefore use a semi-supervised learning approach by either adding a graph Laplacian smoothing regularizer to the objective function [2], or simply applying a Laplacian propagation method [16] at the back-end of the classifier.

IV. Experiments

A. Experimental Settings

We conduct extensive experiments on VHR satellite imagery to evaluate the classification accuracy and computational efficiency of our framework. They are collected from different resources, such as several big cities or the nearby suburbs in Brazil and USA, having large or larger scale with four kinds of geographic settings as shown in Table I. All of these images have the same resolution of 1 meter and they contain millions of pixels which can represent 1 to 4 square kilometers of actual area.

Additionally, apart from the major categories such as slum (or residential regions), urban (or downtown) and forest, these images also contain a set of diverse regions such as trees and forests, lawn and grass fields, water bodies, sandy areas along the shores, *etc*. Thus we create several new categories for each image so as to make them be classified comprehensively in our framework. Note that each pixel is labeled with one or more categories in order to eliminate the ambiguity of mixed regions or boundary areas.

In order to evaluate the OWT-UCM segmentation method in our framework, we compare it to two other superpixel algorithms which can also generate homogeneous and compact segments—Simple Linear Iterative Clustering (SLIC) [17] and Linear Spectral Clustering (LSC) [18]. These two algorithms have been proven to be very effective in the computer vision community as well as in the remote sensing area [19]. In Fig. 3, we show the segmentation results by drawing the contours of each superpixel segment on Fig. 1. As we can see, LSC and SLIC segmentation methods are trying to make the superpixels uniform with similar size and shape. However, although the global image information has been considered in LSC, the boundaries still seem to be unsmooth or locally incoherent when compared with the actual ground-truth. In sharp contrast, the superpixels generated from OWT-UCM form much more reasonable and accurate boundaries. This is highly beneficial to the subsequent step of superpixel classification.

TABLE I: The basic information of our VHR imagery, including the image names, image sizes in terms of height and width, and the number of expert-labeled pixels provided as the ground-truth.

Image	Number of Pixels	
	Total Size	Expert-Labels
Rio-1,2,3,4,5	741×1491	$30, 50, 40, 40, 40$
Madison	1807×2062	40
Milwaukee	2184×2600	30
Detroit	1648×2305	40

Then, we adopt MatConvNet [20] to extract the deep CNN feature for each superpixel as described in Section III-B. It allows us to conveniently borrow the powerful pre-trained CNN models of large benchmark datasets such as ImageNet Large Scale Visual Recognition Competition (ILSVRC) [21], where many everyday items have the similar visual patterns as our remote-sensing objects but with much more variation (as shown in Fig. 4). In our experiments, we use the well-known AlexNet [13] and VGG-16 [14] architectures in order to implement transfer learning and measure their generalization power on our VHR images. The output feature is a long vector with 4,096 dimensions.

At the final stage of our framework, we train a linear SVM with a very small fraction of ground-truth points. The works in [22], [23] have shown that a linear SVM can yield classification results that outperform the original CNN in many visual recognition tasks. We report the misclassification errors of pixels and superpixels in a similar way to [5].

B. Results and Discussion

1) Classification Results: We compare three segmentation methods: OWT-UCM, LSC and SLIC. Meanwhile, we add a control group where we divide the whole image into hundreds or thousands of square grids, each of which containing 7×7 pixel regions and which can be equivalently regarded as the elementary segment. For each segmentation method, we generate the features using both AlexNet and VGG-16, and we also add one additional group for OWT-UCM by extracting the visual descriptors as reported in [5]. In Table II, we report the misclassification errors corresponding to Fig. 1.

From Table II we see that the best performance is obtained from the combination of OWT-UCM and VGG-16. On the one hand, OWT-UCM provides more accurate segmentation of natural boundaries because the superpixel error is almost unaffected when we change the ratio thresholds; however, due to the existing "unpurified" superpixels (shown in Fig. 3) which

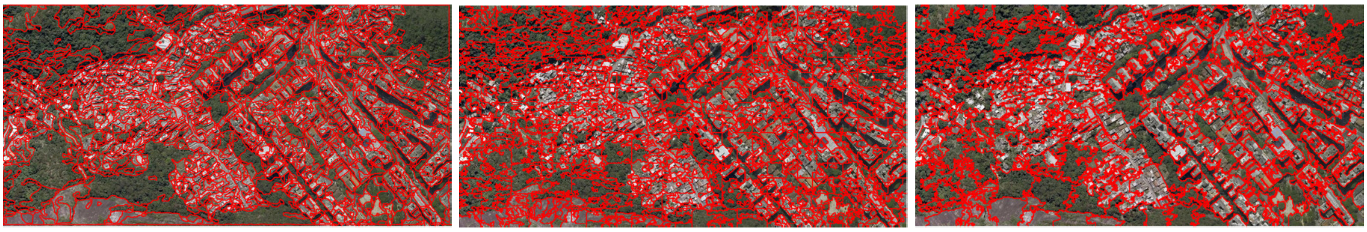

Fig. 3. Superpixel maps obtained from OWT-UCM (left), SLIC (middle) and LSC (right) segmentation methods.

Fig. 4. Example ILSVRC images which are highly similar to the objects in remote-sensing categories. The semantic labels of each column (from left to right) are business district, divided highway, riparian forest and seashore.

TABLE II: Misclassification errors of one image from Rio, Brazil (Rio-1): The three columns below "superpixel" correspond to different thresholds: 25%, 10%, and 0%. This implies that a superpixel is counted as correct only if the ratio of misclassified pixels inside is not more than the threshold.

Tessellation	Features	Pixel	Superpixel		
OWT-UCM	Heuristic	16.68	24.49	24.59	24.59
	AlexNet	12.01	**18.89**	18.99	18.99
	VGG-16	**11.62**	18.96	**18.96**	**18.96**
SLIC	AlexNet	15.55	21.12	29.44	49.44
	VGG-16	15.64	21.35	29.21	49.66
LSC	AlexNet	14.98	18.93	25.51	52.26
	VGG-16	15.24	19.75	25.51	51.85
Grids	AlexNet	15.67	-	-	-
	VGG-16	14.88	-	-	-

involve multiple categories but share the same label, the errors of LSC and SLIC are significantly increased when we require higher ratio of correctly-classified pixels contained in one superpixel. Note that no superpixel errors are reported for the grid-based method because it cannot generate any homogeneous superpixels.

On the other hand, the features extracted from the CNN pre-trained models are clearly superior to the traditional low-level or high-level features selected in a heuristic way. This can be easily seen when we use OWT-UCM as the segmentation method. Additionally, when we compare the classification accuracies of AlexNet and VGG-16, we see that the classification performance stemming from using VGG-16 as a feature extractor exceeds almost all of the others produced by AlexNet. As Yu *et al.* [24] have explained, VGG-16 can remove more unrelated background information than AlexNet. We can obtain useful information from it to make the superpixels more representative. This is very helpful for the final prediction, and more importantly, it also indicates that we are always able to improve our classification accuracies by employing a more powerful CNN model which is well-trained from a general large-scale labeled dataset. Our framework can be adapted to deep CNN architectures, so we do not have to spend huge efforts on collecting a large amount of training data.

TABLE III: Comparisons of pixel-based misclassification errors for 8 images from Rio, Madison, Milwaukee, and Detroit, using a VGG-16 architecture to extract the deep features.

Image	Tessellation			
	OWT-UCM	SLIC	LSC	Grids
Rio-1	**11.62**	15.64	15.24	14.88
Rio-2	**12.32**	15.49	15.26	14.06
Rio-3	**13.87**	17.41	16.88	15.22
Rio-4	**8.05**	13.02	10.85	12.11
Rio-5	**8.35**	13.69	11.03	11.20
Madison	**15.33**	19.03	19.65	18.08
Milwaukee	21.21	23.05	**21.09**	21.87
Detroit	**16.80**	20.52	19.65	18.22

Fig. 5 and 6 show the ground-truth maps and classification results of some representative VHR images. We use VGG-16 as the deep CNN feature extractor because it can yield better classification performance for different segmentation methods. The misclassification errors of all the VHR images are reported in Table III. Similar to the quantitative results in Table II, OWT-UCM with VGG-16 achieves the best (or second-best) performance in all of the cases. Even though the accuracies of the grid-based method are sometimes competitive to ours, the classification map produced by OWT-UCM still looks better, because it is visually closer to the natural boundaries while the grid-based one suffers from a jagged shape due to the finer square-cell partitioning structure.

Note that we run each image on a single machine with a NVIDIA GTX 1050 Ti GPU. For the segmentation-based methods like OWT-UCM, SLIC and LSC, it takes less than 7 minutes to extract the deep CNN features and generate the classification results for all the images. However, for the grid-based method, we have to spend around 40 minutes on the largest image of Madison. This expensive computational cost is a result of its higher density of segments (*i.e.* square grids). It demonstrates that our framework is competitive in terms of both classification accuracy and computational speed in large-scale applications.

2) Deep CNN Feature Analysis: We carry out an analysis of attributes for the deep CNN features extracted by the VGG-16 model. Our intention is to reduce the computational complexity of the SVM classifier and the effects of overfitting when the number of available training samples is very limited. Therefore, we adopt the generalized Fisher score [25] to rank the attributes based on the relevance in descending order and then retain the

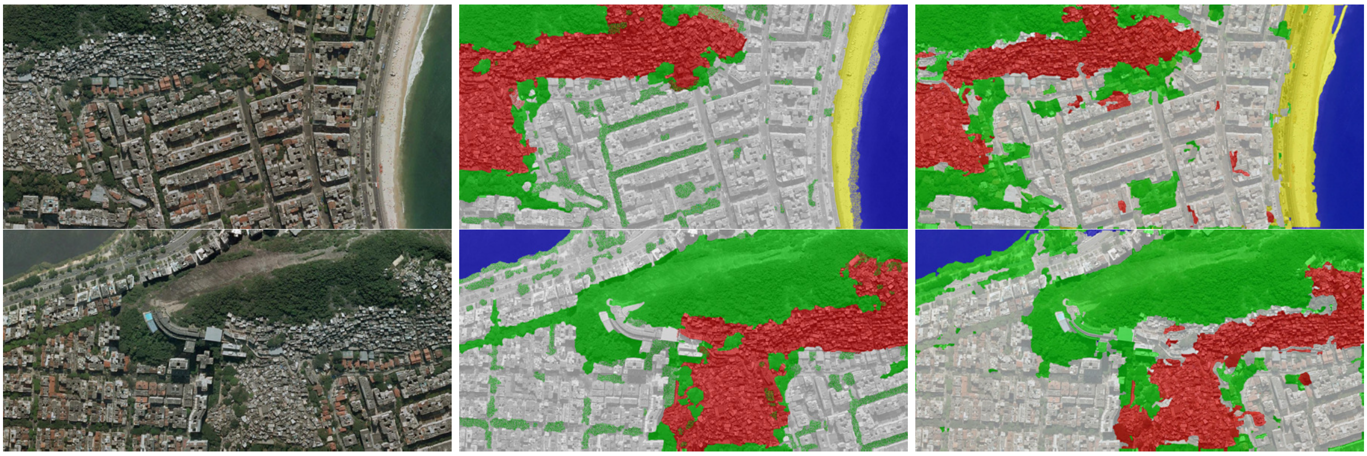

Fig. 5. Example results for the images from Brazil. The left, middle, and right column, respectively, correspond to the actual image, ground-truth maps, and SVM classification results. Different colors such as red, white, green, yellow, and blue are used to identify the slum, urban regions, trees, beach sand, and sea.

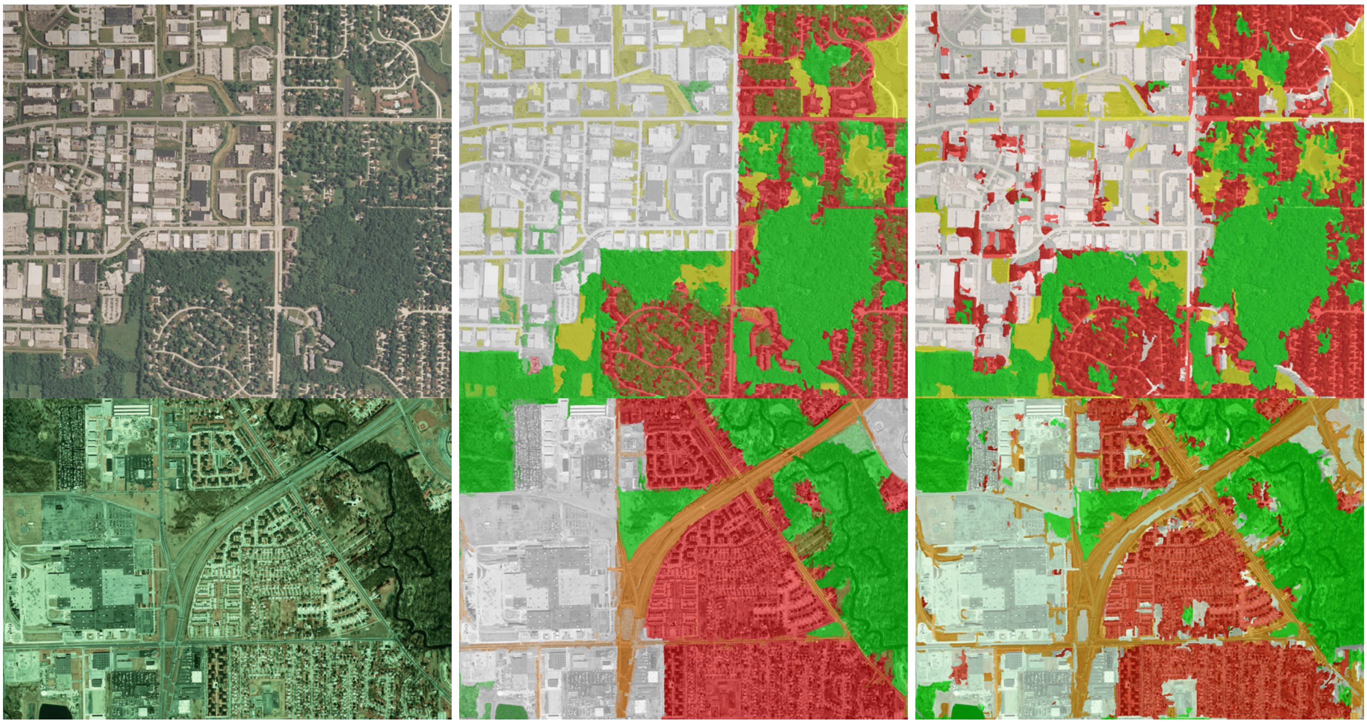

Fig. 6. Example results for the larger images from USA. The left, middle, and right column, respectively, correspond to the actual image, ground-truth maps, and SVM classification results. Different colors such as red, white, green, yellow, *etc.* are used to identify the residential areas, commercial areas (or downtown), forest (or large trees), grass (or lawn) and main roads (or highway).

most relevant ones to keep the discriminating properties of the original features.

We show the changes of pixel-based errors by increasing the number of top-ranking attributes in Fig. 7. The error dramatically decreases at the beginning and then slightly fluctuates near the baseline. This means that a small subset of the attributes is enough for a very good approximation of the original features. To further explore this finding, we evaluate the classification accuracies on the remaining images and report the quantitative results in Table IV. According to Table III, a comparable performance can be achieved by only using a few top attributes (*e.g.* top-16) from each category, which shows the great advantage of applying a simple feature selection technique after completing transfer learning. Therefore, we can significantly reduce the burden of our classifier by feeding it much shorter features. It does not have much impact on the classification accuracies and sometimes can even be helpful if the irrelevant overfitting noise features are removed.

V. Conclusion

We have developed an efficient machine learning algorithm to accurately label large-scale VHR imagery. It integrates a locally homogeneous state of the art superpixel segmentation algorithm (OWT-UCM) with transfer learning-based deep

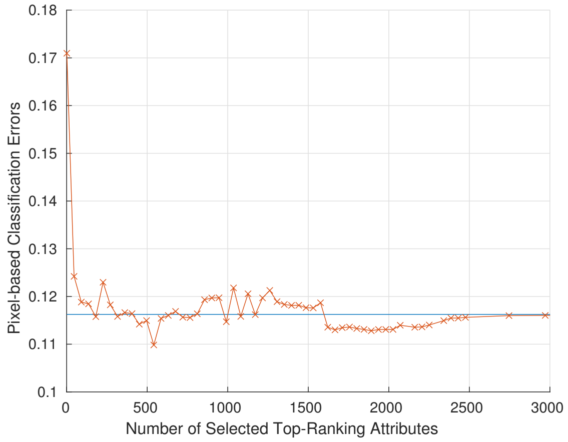

Fig. 7. Pixel-based errors computed by selecting different numbers of top-ranking attributes to construct the short CNN features. The blue line (denoted as *baseline*) shows the error of using the original deep CNN features.

TABLE IV: Comparison of the pixel-based misclassification errors produced by using different numbers of top-ranking attributes from each category.

Image	Number of Top-Ranking Attributes				
	2	4	8	16	32
Rio-1	14.29	14.50	14.73	**12.55**	12.76
Rio-2	22.36	20.15	16.09	16.08	**13.64**
Rio-3	24.15	19.91	15.33	15.25	**14.86**
Rio-4	15.77	19.20	10.79	8.83	**8.78**
Rio-5	14.60	10.58	**10.27**	10.68	10.79
Madison	24.96	17.88	17.12	**14.76**	14.91
Milwaukee	36.98	29.54	25.62	**23.44**	23.50
Detroit	28.12	24.11	22.77	19.88	**19.34**

learning techniques to achieve excellent overall classification performance. Only a very small amount of expert labeled data is provided to our framework for training the SVM model, and the hierarchical structure of the superpixel-to-pixel labeling makes it a very efficient and accurate approach to assign the pixel-level labels. We derive representative deep features by utilizing popular CNN models trained from complicated recognition tasks. A feature selection method is empirically shown to play an important role in significantly reducing the complexity of supervised or semi-supervised classification without sacrificing performance. We demonstrate its effectiveness by measuring and comparing the misclassification errors on remote sensing satellite images. Our future work will focus on extending our framework to more applications including spatio-temporal datasets.

Acknowledgments

This work is partially supported by NSF IIS 1743050.

References

[1] Y. Yan, X. Tian, L. Yang, Y. Lu, and H. Li, "Semantic-spatial matching for image classification," in *Multimedia and Expo (ICME), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[2] Y. Yan, M. Sethi, A. Rangarajan, R. R. Vatsavai, and S. Ranka, "Graph-based semi-supervised classification on very high resolution remote sensing images," *International Journal of Big Data Intelligence*, vol. 4, no. 2, pp. 108–122, 2017.

[3] Y. Yan, M. Sethi, A. Rangarajan, and S. Ranka, "Super-scalable computation framework for automated terrain identification," in *Dependable, Autonomic and Secure Computing, 2017 IEEE 15th Intl. Conf.* IEEE, 2017, pp. 1127–1134.

[4] 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, 2015.

[5] M. Sethi, Y. Yan, A. Rangarajan, R. R. Vatsavai, and S. Ranka, "Scalable machine learning approaches for neighborhood classification using very high resolution remote sensing imagery," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 2069–2078.

[6] ——, "An efficient computational framework for labeling large scale spatiotemporal remote sensing datasets," in *Contemporary Computing (IC3), 2014 Seventh International Conference on*. IEEE, 2014, pp. 635–640.

[7] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645–657, 2017.

[8] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[9] O. A. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 44–51.

[10] R. R. Vatsavai, "Gaussian multiple instance learning approach for mapping the slums of the world using very high resolution imagery," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2013, pp. 1419–1426.

[11] G. Zhang, X. Jia, and J. Hu, "Superpixel-based graphical model for remote sensing image mapping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 11, pp. 5861–5871, 2015.

[12] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *arXiv preprint arXiv:1508.00092*, 2015.

[16] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 811–818.

[17] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[18] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1356–1363.

[19] N. Audebert, B. Le Saux, and S. Lefevre, "How useful is region-based classification of remote sensing images in a deep learning framework?" in *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International*. IEEE, 2016, pp. 5091–5094.

[20] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 689–692.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[22] B. Athiwaratkun and K. Kang, "Feature representation in convolutional neural networks," *arXiv preprint arXiv:1507.02313*, 2015.

[23] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[24] W. Yu, K. Yang, Y. Bai, T. Xiao, H. Yao, and Y. Rui, "Visualizing and comparing AlexNet and VGG using deconvolutional layers," *International Conference of Machine Learning (ICML) Workshop on Visualization for Deep Learning*, 2016.

[25] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," *arXiv preprint arXiv:1202.3725*, 2012.