DEEP LEARNING FOR SUPERPIXEL-BASED CLASSIFICATION OF REMOTE SENSING IMAGES

C. Gonzalo-Martín[a,b]*, A. Garcia-Pedrero[a,b], M. Lillo-Saavedra[c,d], E. Menasalvas[a,b]

[a] Center for Biomedical Technology, Universidad Politécnica de Madrid, Campus de Montegancedo, Pozuelo de Alarcón 28233, Spain - {consuelo.gonzalo}@upm.es
[b] Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, Campus de Montegancedo, Boadilla del Monte 28660, Spain
[c] Faculty of Agricultural Engineering, University of Concepción, Chile
[d] Water Research Center for Agriculture and Mining, CRHIAM, University of Concepción, Chile

KEY WORDS: Convolutional Neural Networks, Remote Sensing Image Classification, Superpixels

ABSTRACT:

Recently deep learning-based methods have demonstrated excellent performance on different artificial-intelligence tasks. Even though, in the last years, several related works are found in the literature in the remote sensing field, a small percentage of them address the classification problem. These works propose schemes based on image patches to perform pixel-based image classification. Due to the typical remote sensing image size, the main drawback of these schemes is the time required by the window-sliding process implied in them. In this work, we propose a strategy to reduce the time spent on the classification of a new image through the use of superpixel segmentation. Several experiments using CNNs trained with different sizes of patches and superpixels have been performed on the ISPRS semantic labeling benchmark. Obtained results show that while the accuracy of the classification carried out by using superpixels is similar to the results generated by pixel-based approach, the expended time is dramatically decreased by means of reducing the number of elements to label.

1. INTRODUCTION

Recently deep learning-based methods have demonstrated excellent performance on different artificial-intelligence tasks, including speech recognition (Hinton et al., 2012a), natural language processing (Dahl et al., 2012), and computer vision (Krizhevsky et al., 2012). In the later area, Convolutional Neural Networks (CNNs) play a major role for processing visual-related problems such as image classification (Lee et al., 2009; Sermanet et al., 2013), object detection (Girshick et al., 2014), and face recognition (Taigman et al., 2014).

CNNs are biological-inspired variants of feed-forward neural networks, where each layer is a non-linear feature detector performing local processing of contiguous features within each layer (Biem, 2014). In this regard, the CNN architecture is able to exploit the strong spatially local correlation present in natural images by enforcing a local connectivity pattern between neurons of adjacent layers. This leads to higher conceptual representation as information moves up to the output layer. In this context, a CNN is able to generate patterns from an image in a hierarchical manner similar to that of the mammalian visual cortex. Empirical studies have demonstrated that these methods often provided better results than traditional machine learning methods (Larochelle et al., 2009; Salakhutdinov and Hinton, 2009).

Two different approaches to exploit the hierarchical analysis capacity of CNN for image analysis can be found in literature: feature extraction and classification. In feature extraction approach, pre-trained CNN models are used to automatically extract image features that later are analyzed by traditional machine learning methods (Sharif Razavian et al., 2014). In classification approach, a CNN is trained from scratch using a large set of images (Krizhevsky et al., 2012), which requires high performance equipment for processing (e.g., graphics processing units).

*Corresponding author

As far as our knowledge the use of CNNs for processing remotely sensed imagery is relatively recent. Particularly, CNNs have been used in remote sensing area for generating thematic maps following a pixel-based approach (Paisitkriangkrai et al., 2015; Zou et al., 2015). In a pixel-based approach, during training phase, training images are broken down into overlapping patches, where each patch is centered on a pixel which provide the class for the whole patch. Once CNN is trained with these patches, it is applied to test images using a window-sliding approach in order to provide a label to each pixel in the images (pixel-based classification). As it is known, window-sliding approach is a highly time-consuming process, especially when images have billions of pixels as very-high resolution remote sensing images.

In this work, we propose a strategy to reduce the time spent to classify a new image through the use superpixel segmentation. Superpixels (SPs) are a form of image segmentation, but the focus lies more on a controlled oversegmentation, not on segmenting meaningful objects. By controlling the size and compactness of the SPs, the image can be divided into several homogeneous regions with determined number of pixels. Thus it is possible to create SPs that can be completely contained by a patch of determined size. In this way, SPs are expected to maintain all the characteristics of a reduced environment inside of the area learned by a CNN. This allows to generate, during testing phase, only patches which centers correspond to the centroids of the SPs. This reduces the number of elements that must be considered in the window-sliding approach during labeling process of a new image, which is a bottleneck of this type of approaches.

Several experiments using CNNs trained with different sizes of windows and SPs have been performed on the ISPRS semantic labeling benchmark. Obtained results show that while the accuracy of the classification carried out by using SPs is similar to the results generated by the conventional window-sliding approach, the expended time is dramatically decreased by means of reducing the number of elements to label.

2. DATA AND METHODS

2.1 Data description

In this work, the ISPRS labeling contest dataset[1] has been used. This dataset consists of very-high resolution true ortho-photo (TOP) tiles and their corresponding digital surface models (DSMs). The images includes different urban scenes from a relatively small village with many detached buildings and small multi story buildings (Vaihingen, Germany). In addition, sixteen labeled scenes serving as ground-truth data are part of this dataset. These images have been classified manually into six land cover classes: impervious surfaces, building, low vegetation, tree, car, and clutter/background. This last class includes different types of land-covers that have a small presence in the analyzed scenes.

2.2 Superpixels

Superpixel processing is carried out by the SLIC algorithm (Achanta et al., 2010), which is based on the well-known *k-means* method to group pixels in a conventional color space. SLIC superpixels are generated according to two criteria: spectral similarity (limited to three channels) and spatial proximity. In the SLIC procedure, the generation of SPs is based on the assumption that limiting the search space to a region proportional to the desired SP size reduces considerably the calculation time. In fact, its computational complexity is linear in the number of pixels in the image (Achanta et al., 2012). Moreover, a weighted distance that combines spectral and spatial proximity allows controlling the size and compactness of the SP (Achanta et al., 2012). Therefore, it has two parameters: k, the desired number of superpixels, and c, the compactness factor. A larger value of c emphasizes the importance of the spatial proximity resulting in more compact SPs.

In this work, a modified version of SLIC (Gonzalo-Martín et al., 2016; Garcia-Pedrero et al., 2015) is used to generate SPs. This version extends the definition of spectral proximity provided by the original SLIC to work with multispectral images of B bands. The first step of the segmentation framework begins with the sampling of k initial cluster centers on a regularly spaced grid of g pixels. The initial centers are defined as:

$$C_i = [p^1, p^2, \ldots, p^B, x, y]^T, \qquad i = 1, 2, \ldots, k \tag{1}$$

where p^b represents the spectral value in band $b-th$ of pixel p at position x and y, and B denotes the number of spectral bands. To produce similar sized superpixels, the grid interval is defined as $g = N/k$, where N is the total number of pixels in the image. g determines the size of the superpixels, the greater value of g, the larger the SPs. This allows to adapt superpixels to specific requirements for real-world applications as a determined scale of analysis.

In the next step, each pixel p is associated to the nearest cluster center whose search space overlaps its location. The search region is enclosed to an area of $2g \times 2g$ pixels around each superpixel center. Then the cluster centers are updated to be the mean vector of all pixels belonging to the cluster. Both steps are repeated iteratively until a maximum of 10 iterations, since no further significant changes in superpixel quality could be observed (Achanta et al., 2010).

The clustering distance is a weighted relationship between spectral and spatial measures. The first measure ensures superpixel homogeneity, and the second one enforces compactness and regularity in superpixels shape. In order to work with multispectral images, the spectral square distance between pixels i and j is defined as follows:

$$d_c^2 = \sum_{b=1}^{B} (p_i^b - p_j^b)^2 \tag{2}$$

The spatial square distance is calculated as:

$$d_s^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \tag{3}$$

where x and y denote the position of the pixel. Finally, the clustering distance is calculated as:

$$D = d_c + \left(\frac{c}{g}\right) d_s \tag{4}$$

where c controls the compactness of the superpixels. According to Garcia-Pedrero et al. (2015), a 3.9% of the maximum pixel value in image as c is optimal.

2.3 Convolutional Neural Network

The concept of CNN is not new, in fact, it was first proposed in 1980 by (Fukushima, 1980) with the name of NeoCognitron, and later refined by (LeCun et al., 1989). CNNs have some characteristics that distinguish them from traditional feed-forward neural networks. Unlike traditional feed-forward layers, convolutional layers have neurons with limited *receptive fields*, i.e., they only process a local image region that affects a particular element in the output. Moreover, as their name reflects, the output of this layer is computed as a spatial convolution using a learned filter over its input. The result of this convolution is a set of features of the image. Nowadays, both technological and algorithmic advances have allow the implementation and use of CNNs. Regarding technological advances, it should be mentioned the advent on the market of affordable graphics processing units (GPUs), together the availability of large databases of annotated images. From an algorithmic point of view, two main contributions have been the proposal of the rectified linear unit (ReLu) (Jarrett et al., 2009), which allows a faster training, and the dropout strategy (Hinton et al., 2012b) to reduce overfitting. All these advances were used by Krizhevsky et al. (2012) achieving outstanding results in image processing tasks. This work may be considered as the starting point for the widespread use of CNNs.

CNN architecture typically comprises several layers of different types (Castelluccio et al., 2015):

Convolutional layers. As their name suggests, they compute the convolution of the input image with the weights of the network. These layers are characterized by few parameters: the size of filters, the filter spatial support, the step between different windows and an optional zero-padding which controls the size of the layer output. The analysis of the image is done at different scale in the different layers. As the layers are deeper, the features extracted from the image are higher-level.

Pooling layers. The mission of these layers is to reduce the size of the input layer through some local non-linear operations. Their most important parameters are the support of the pooling window and the step between different windows.

Normalization layers. Their objective is to improve generalization of the CNN. For that, they use inhibition schemes inspired in the real neurons of the brain. Neurons typically used in these layers are sigmoid type.

[1] `http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html`. Last access: June 2016.

Fully-connected layers. These layers are typically used in the last levels of the network. Since the size of the image is reduced in the previous layers, their full connectivity is not a limitation for using them. These layers have the capacity of abstracting the low-level information generated in previous layers for a final decision.

2.4 Architecture

The CNN architecture used in this work is inspired on the approach presented in Paisitkriangkrai et al. (2015). A scheme of this architecture is shown in Figure 1. It is defined by the alternation of three convolutional layers, which compute the convolution between the input of each layer and a set of learned filters; and other additional layers (contrast normalization and max-pooling layers), which apply a non-linear transformation (rectified linear unit - ReLU) and sub-samples the output of the corresponding convolutional layer, respectively. The role of these additional layers is to improve the robustness of the network to distortions and small translations. Moreover, two fully-connected layers are included at the end of the architecture. The output of these layers feeds a k-way soft-max layer which produces a probability distribution over 6 class labels. To reduce overfitting in the fully-connected layers, the dropout method has been used (Hinton et al., 2012a). In Figure 1 the expression i x j x k under each convolutional layer represents the size of the kernels associated to this layer, where the number of kernels for the three first layers are 32, 64, and 128, respectively. Each of these kernels generates a feature map for feeding the next layer.

2.5 Superpixel-based labeling approach

As mentioned above, CNNs have demonstrated a good performance in computer vision tasks such as classification where a single label is assigned to the entire image. However, to perform a pixel-based classification it is necessary to break the image down into overlapping patches. Each patch is centered on a pixel which provide the class for the entire patch. CNN is trained using a large set of patches randomly selected trying to maintain a good distribution of the classes of interest. During the testing phase a window-sliding approach is commonly followed to provide a label to each pixel of a test image. This approach, shown in Figure 2(a), consists in applying a trained CNN to a patch defined by a window, the output label is then assigned to the center pixel of the window. To produce a thematic map, several windows are generated by centering a window in each pixel of the image. A mirror padding strategy is used to label the pixels around the boundaries of the image. Since all pixels must be processed, the sliding window approach is a time consuming task, especially if we consider that a very-high resolution image usually has billions of pixels.

To alleviate this drawback, an alternative approach based on the use of superpixels during labeling process was explored in this work. In the proposed approach, images are segmented into superpixels using the method described in Section 2.2, the resulting superpixels are then used as minimum processing units during labeling process. Since superpixels are the product of a controlled over-segmentation of the image; they tend to be similar in size and color, as well as belonging to only one object. These properties allowed to generate superpixels completely contained by a window of determined size, maintaining all the characteristics of a reduced environment inside of the area to analyze with a CNN. The proposed method reduces the number of windows to those centered on the pixels corresponding to the centroids of the superpixels, as shown in Figure 2(b), then superpixels are labeled according to the output of the CNN obtaining a thematic map. Thus the time required for labeling a new image is drastically reduced.

3. EXPERIMENTS AND RESULTS

Since the objective of this work is to prove the effectiveness of the superpixel-based labeling approach (Section 2.5) regarding a traditional window-sliding approach, a set of experiments using different superpixels and windows sizes (W) have been carried out. Twelve images of the labeled dataset have been used for training, while the remaining four (15, 28, 34, 37) were used for testing purposes. From training images, a total of 10000 patches for each class have been randomly selected, then using a proportion of 70-30 to create training and validation sets, respectively. A total of twelve different CNNs were generated, one for each combination between three window sizes (32x32, 48x48, 64x64) and four SP sizes (20, 30, 40, 50). Each CNN was trained using a different training and validation sets, obtaining an accuracy of about 80% in the validation set.

Once CNNs were trained, each of them was used for labeling the four testing images. The labeling process was carried out using both the conventional pixel-based (window-sliding) approach and the superpixel-based approach. F1-score values were calculated in each case to measure the error in the classification, and how these results are affected by the labeling approaches. Table 1 summarizes the obtained results.

Figures 3(a), 3(b), and 3(c) represent the differences in percentage between F1-score after applying the different approaches for labeling testing images, using a window size of 32, 48, and 64, respectively. As can be observed, the differences increase as the SP size does. On the other hand, as can be observed in Figure 3(d) differences between both labeling approaches decrease as the window size increases. However the use of larger windows also increase the calculation time due to the increase in the amount of data to be processed (larger patches).

The results show that the pixel-based approach is always slightly better, however in general terms, the difference between the two labeling approaches is negligible. If we estimate the time required for the labeling process in term of processing data, in our experiments, depending on the SP size, the reduction has been around a 96% of data, when the labeling process is performed based on superpixels against the same process based on pixels (sliding-window).

Figure 4 shows the comparison between the classification results obtained with the pixel-based approach (Figure 4(a)) and the superpixel-based approach (Figure 4(b)) for a W=64 and SP size of 50. In agreement with numerical results, the differences between both images are contemptible. However, in order to show the possible discrepancies between both images, their difference is displayed in Figure 4(c). As can be appreciated, these differences are mainly related to the edges between regions of different classes.

All experiments have been done using an NVIDIA GTX970 GPU with 1664 CUDA cores and 4 GB of memory. Codes were developed in python using Caffe framework (Jia et al., 2014). The size of the trained CNN has been determined by the time and memory limitations imposed by the GPUs used. In our experiments, the average time reduction has been from 5.5 hours for the pixel-based labeling against 20 min. for the SP-based one.

4. CONCLUSIONS

From the results obtained in the experiments carried out in this work, it is concluded that the proposed methodology based on superpixels to automatically label satellite imagery using CNN,

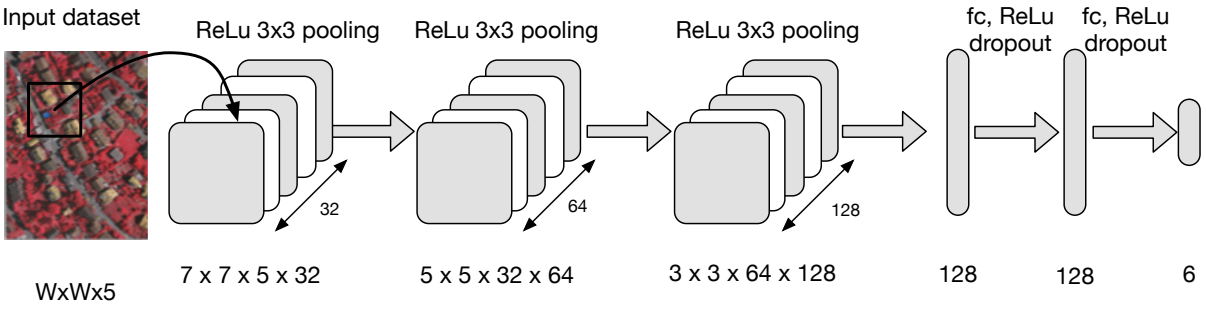
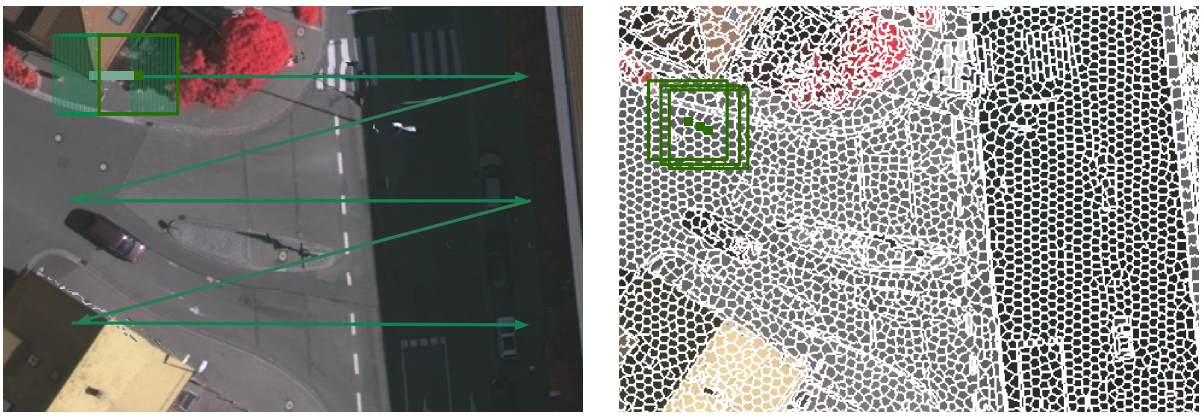


Figure 1. CNN architecture used during the experiments.



(a) Window-sliding approach (b) Our approach

Figure 2. Green boxes indicate the windows that serve to feed a CNN during the labeling process in order to generate the final thematic map of the scene under analysis.

Table 1. Classification results (F1-values) obtained by applying the different labeling approaches. The difference between both approaches is shown in percentage.

W size	**Image**	**Superpixel size**											
		20			30			40			50		
		Pixel	SP	Diff. (%)	Pixel	SP	Diff. (%)	Pixel	SP	Diff. (%)	Pixel	SP	Diff. (%)
32	15	0.7202	0.7187	0.1435	0.7256	0.7233	0.2294	0.7248	0.7219	0.2854	0.7204	0.7166	0.3830
	28	0.7312	0.7297	0.1541	0.7346	0.7326	0.1985	0.7306	0.7269	0.3684	0.7384	0.7335	0.4908
	34	0.7468	0.7449	0.1858	0.7531	0.7511	0.2041	0.7530	0.7496	0.3386	0.7574	0.7530	0.4344
	37	0.7748	0.7732	0.1635	0.7703	0.7678	0.2475	0.7765	0.7732	0.3328	0.7629	0.7585	0.4417
48	15	0.7555	0.7539	0.1643	0.7584	0.7567	0.1770	0.7515	0.7486	0.2865	0.7515	0.7474	0.4093
	28	0.7510	0.7493	0.1624	0.7463	0.7435	0.2798	0.7532	0.7497	0.3478	0.7631	0.7587	0.4450
	34	0.7591	0.7575	0.1634	0.7618	0.7597	0.2129	0.7692	0.7666	0.2614	0.7703	0.7657	0.4561
	37	0.7936	0.7914	0.2107	0.7911	0.7887	0.2335	0.7932	0.7899	0.3301	0.7976	0.7940	0.3638
64	15	0.7401	0.7388	0.1238	0.7617	0.7598	0.1923	0.7540	0.7510	0.2999	0.7548	0.7511	0.3699
	28	0.7659	0.7648	0.1135	0.7670	0.7647	0.2264	0.7534	0.7502	0.3148	0.7626	0.7579	0.4706
	34	0.7645	0.7626	0.1861	0.7581	0.7561	0.1993	0.7699	0.7666	0.3308	0.7715	0.7673	0.4200
	37	0.7986	0.7971	0.1540	0.7971	0.7958	0.1272	0.7978	0.7954	0.2384	0.7979	0.7944	0.3478

provides a dramatically computer time reduction, estimated on the base of number of data to be processed, with a negligible decrease of the label accuracy. It is hoped that the results can be improved by increasing the deep of the CNN, as well as the size of the windows, which will be feasible with faster GPUs.

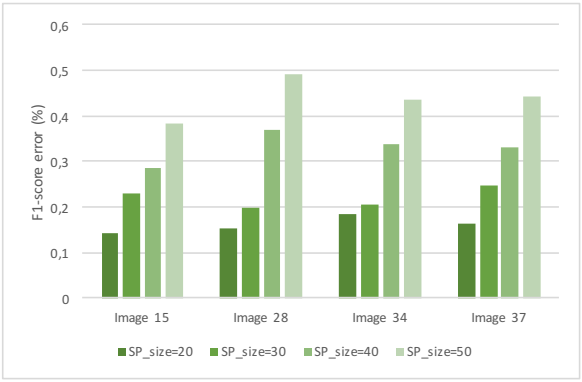
ACKNOWLEDGEMENTS

This work has been funded by the Universidad Politécnica de Madrid (AL16-PID-17). A. Garcia-Pedrero (grant 216146) acknowledges the support for the realization of his doctoral thesis to the Mexican National Council of Science and Technology (CONACyT).

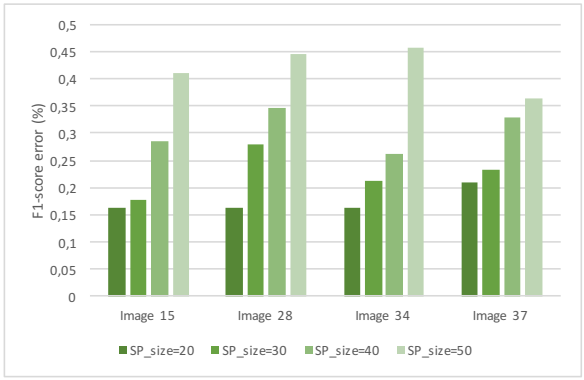
References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Süsstrunk, S., 2010. Slic superpixels. *École Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep*.

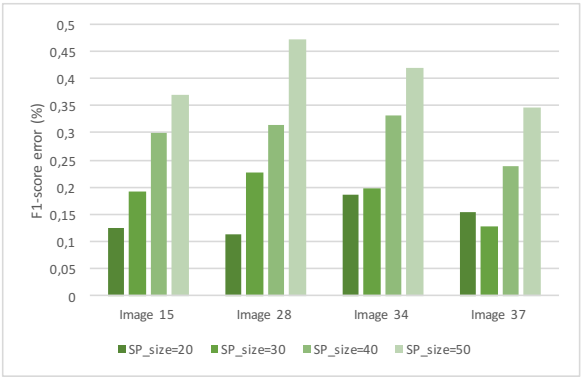
Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Susstrunk, S., 2012. SLIC Superpixels Compared to State-



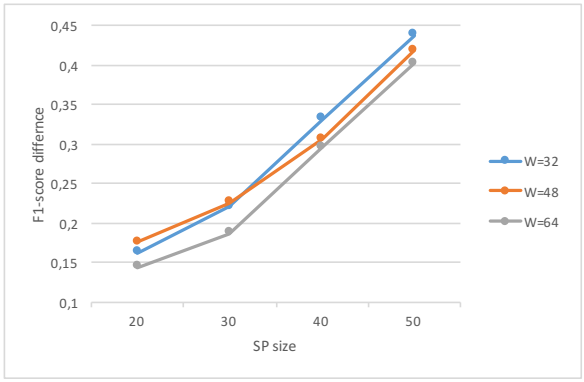
(a) W=32



(b) W=48

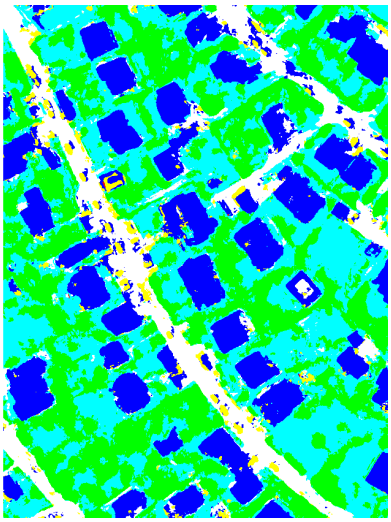


(c) W=64

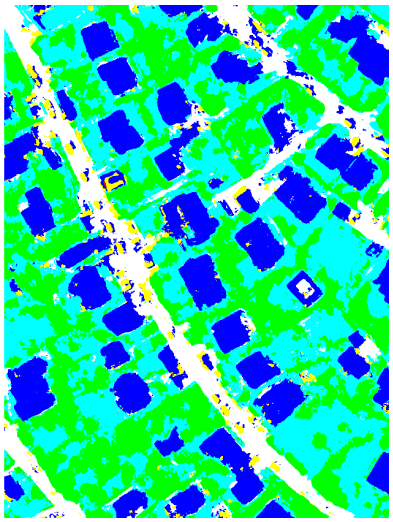


(d) % Average F1-score error

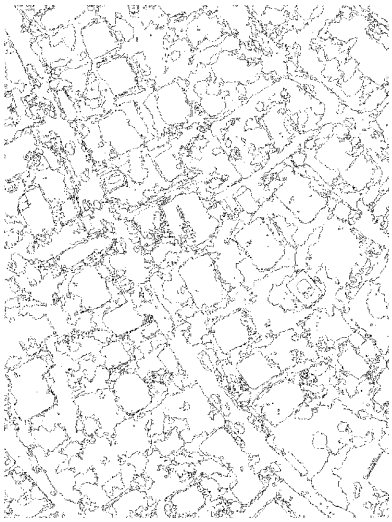
Figure 3. % F1-score errors.



(a) Window-sliding approach



(b) Superpixel approach



(c) Differences between both approaches

Figure 4. Labeling results.

of-the-Art Superpixel Methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34(11), pp. 2274–2282.

Biem, A., 2014. Neural networks: A review. In: C. C. Aggarwal (ed.), *Data Classification: Algorithms and Applications*, Chapman and Hall/CRC, chapter 8, pp. 206–236.

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

Dahl, G. E., Yu, D., Deng, L. and Acero, A., 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on* 20(1), pp. 30–42.

Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics* 36(4), pp. 193–202.

Garcia-Pedrero, A., Gonzalo-Martin, C., Fonseca-Luengo, D. and Lillo-Saavedra, M., 2015. A geobia methodology for fragmented agricultural landscapes. *Remote Sensing* 7(1), pp. 767–787.

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gonzalo-Martín, C., Lillo-Saavedra, M., Menasalvas, E., Fonseca-Luengo, D., García-Pedrero, A. and Costumero, R., 2016. Local optimal scale in a hierarchical segmentation method for satellite images. *Journal of Intelligent Information Systems* 46(3), pp. 517–529.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., r. Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. and Kingsbury, B., 2012a. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29(6), pp. 82–97.

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

Jarrett, K., Kavukcuoglu, K., Lecun, Y. et al., 2009. What is the best multi-stage architecture for object recognition? In: *2009 IEEE 12th International Conference on Computer Vision*, IEEE, pp. 2146–2153.

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

Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp. 1097–1105.

Larochelle, H., Bengio, Y., Louradour, J. and Lamblin, P., 2009. Exploring strategies for training deep neural networks. *The Journal of Machine Learning Research* 10, pp. 1–40.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D., 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation* 1(4), pp. 541–551.

Lee, H., Grosse, R., Ranganath, R. and Ng, A. Y., 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, pp. 609–616.

Paisitkriangkrai, S., Sherrah, J., Janney, P. and Hengel, A., 2015. Effective semantic pixel labelling with convolutional networks and conditional random fields. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 36–43.

Salakhutdinov, R. and Hinton, G. E., 2009. Deep boltzmann machines. In: *International conference on artificial intelligence and statistics*, pp. 448–455.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y., 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*.

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

Taigman, Y., Yang, M., Ranzato, M. and Wolf, L., 2014. Deepface: Closing the gap to human-level performance in face verification. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zou, Q., Ni, L., Zhang, T. and Wang, Q., 2015. Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters* 12(11), pp. 2321–2325.