Deep Learning for Urban Remote Sensing

(Invited Paper)

Nicolas Audebert*†, Alexandre Boulch*, Hicham Randrianarivo*‡, Bertrand Le Saux*,
Marin Ferecatu‡, Sébastien Lefèvre† and Renaud Marlet§
*ONERA The French Aerospace Lab, DTIM, F-91761 Palaiseau, France
†Univ. Bretagne-Sud, UMR 6074, IRISA, F-56000 Vannes, France
‡CNAM ParisTech, CEDRIC Lab., F-75141, France
§LIGM, UMR 8049, Ecole des Ponts, UPE, Champs-sur-Marne, France

***Abstract*—This work shows how deep learning techniques can benefit to remote sensing. We focus on tasks which are recurrent in Earth Observation data analysis. For classification and semantic mapping of aerial images, we present various deep network architectures and show that context information and dense labeling allow to reach better performances. For estimation of normals in point clouds, combining Hough transform with convolutional networks also improves the accuracy of previous frameworks by detecting hard configurations like corners. It shows that deep learning allows to revisit remote sensing and offers promising paths for urban modeling and monitoring.**

I. Introduction

Deep learning is a new way to solve old problems in remote sensing. Various changes in the technical ecosystem made it possible: abundant data (from more and more automated sensing and processing), a better understanding of the theory of machine learning that led to complex algorithms and computational capacities which allow training in tractable times.

It comes out that we can now use such powerful statistical models for various remote sensing tasks: detection, classification or data fusion. Since the early applications to road detection back in 2010 [18], convolutional networks have been successfully used for classification and dense labeling of aerial imagery. They have defined new state-of-the-art performances and showed the re-use of cross-domain databases is possible to gain and transfer knowledge [20], [9]. New challenges will soon be addressed, such as image registration or 3D data analysis. Serendipity plays a role here: while meta-data for standard decision-making are not always available, the co-existence of various correlated, continuous data allows the training of regression models which give the same output as analytic processes, but faster and with more robustness.

In the following, we propose several deep learning approaches for urban monitoring and assessment: classification (Section II), contextual classification (Section III), dense semantic mapping (Section IV) of aerial images and normal estimation in point clouds (Section V). Two European towns are chosen to evaluate the results: Vaihingen (ISPRS dataset [22]) and Zeebruge (IEEE-GRSS dataset [12]). Datasets contain several Infrared-Red-Green (ISPRS) or Red-Green-Blue (IEEE-GRSS) tiles, and a LiDAR-captured point cloud.

II. Multiscale semantic classification

Semantic labeling (also known as semantic segmentation in computer vision) consists in automatically building maps of geo-localized semantic classes (e.g., land use: buildings, roads, vegetation; or objects: vehicles) upon Earth Observation data [9]. In the following, we present our approach for multiscale classification using pre-trained convolutional neural networks (CNNs) based on AlexNet [13]. While easy to implement, it yields state-of-the-art performances on various datasets [9], [2] and thus works as an efficient baseline.

A. Approach

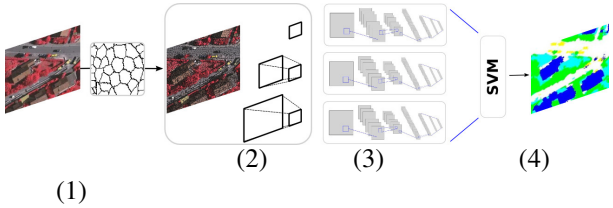


Fig. 1. Semantic labeling workflow: (1) superpixel segmentation; (2) multi-scale patch extraction; (3) classification with parallel CNNs; (4) fusion with multi-class SVM.

a) Superpixel segmentation: We first segment orthophotos using the SLIC (Simple Linear Iterative Clustering [1]) method. This allows to generate coherent regions at sub-object level. Patches used to feed the CNN will then be extracted around the superpixel centroïd, and the class estimated by the algorithm will be assigned to the whole superpixel.

b) Multiscale data extraction: For each superpixel, we extract patches at various sizes: 32×32 (which is roughly the size of a car), 64×64 and 128×128. Then we resize all patches to 228×228 to fit the AlexNet input layer. This allows to extract a multiscale pyramid of appearances for each superpixel location, more representative than monoscale 32×32 or 64×64 (cf. Table I).

c) Convolutional Neural Networks: We use a pretrained AlexNet neural network [13] as a feature extractor. It is made of 5 convolutional layers, some of which followed by max-pooling, and two fully connected layers with a final softmax. The model weights remain those obtained by training for the ImageNet classification task. Patches extracted from the image

978-1-5090-5808-2/17/$31.00

at different scales are passed through the network and the last layer outputs before the softmax are used as feature vectors.

d) Data fusion and classification: We concatenate the resulting vectors to produce one feature vector (sample). At training time, we process the images with ground truth and for each superpixel we associate the newly computed sample with the label obtained by majority vote. We then use this training set to train a linear Support Vector Machine (SVM), whose parameters are optimized by stochastic gradient descent (SGD). At testing time, we use the SVM to predict the label of each unknown superpixel, and then associate to all pixels in this region the predicted output label. Thus, the SVM performs both the data fusion of various networks (i.e., various scales) and the classification.

III. Contextual classification

A. Contextual information and graph model

The classifier from Section II makes decisions using information from a single superpixel location. The basic principle behind contextual classification is to also get benefit from information of the neighborhood to regularize the classification map. Such information can consist in the appearance of superpixel neighbors or the relationships with and between neighbors.

Locally, around each superpixel, we extract a subgraph by picking neighbor superpixels that fall in a circle of radius r (cf. Fig. 2). Nodes correspond to the superpixels with values defined as the features extracted by the AlexNet CNN at scale 32×32. Edges are defined between all superpixels with values defined by the pairwise context features: distance between neighbors, normalized distance w.r.t. the neighborhood, relative orientation, appearance similarity, and neighbor importance (inverse of log-distance).

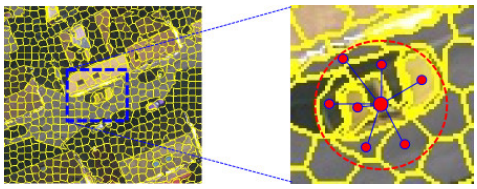


Fig. 2. Subgraph modeling of relationships between superpixel neighbors.

B. Structural model learning and prediction

The training set consists in the local subgraphs x^n associated with the set y^n of labels y_i^n of the superpixel nodes. We denote it by $\{X = \{x^n\}_{n=1}^N, Y = \{y^n\}_{n=1}^N\}$. The Structural SVM [19] generalizes the SVM for structured output labels. It introduces an auxiliary evaluation function $g(x, y, w)$ over subgraphs (linear combination over nodes and edges): this includes unary and pairwise costs. It is denoted as a scalar product by introducing the reshaping function ϕ:

$$\begin{aligned} g(x, y, w) &= \langle w, \phi(x, y) \rangle \\ &= \sum_{i=1}^{N} \phi(x_i, y_i, w) + \sum_{i,j=1}^{N} \psi(x_i, x_j, y_i, y_j, w) \end{aligned} \quad (1)$$

Then SSVM minimizes the same objective function as the standard SVM:

$$w^* = \operatorname{argmin} \frac{1}{2}||w||^2 + \frac{C}{N} \sum_{n=1}^{N} l(x^n, y^n, w) \quad (2)$$

but with a loss function (Eq. 3) which judges whether the prediction made for training the subgraph is *good* or *similar enough* to the training output (vector of labels). This is a multi-label problem which is minimized with quadratic pseudo-boolean optimization.

$$l(x^n, y^n, w) = \max_{y \in Y} \Delta(y, y^n) - g(x^n, y^n, w) + g(x^n, y, w) \quad (3)$$

Predicting a label for each superpixel of an unknown image is achieved by considering local subgraphs for which we predict vectors of labels following Eq. 4. Table I shows that context helps refining the classification rates.

$$f(x^n) = \operatorname*{argmax}_{y \in Y} g(x^n, y, w) \quad (4)$$

IV. Dense prediction for semantic labeling

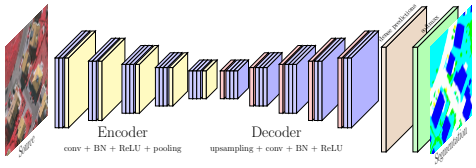


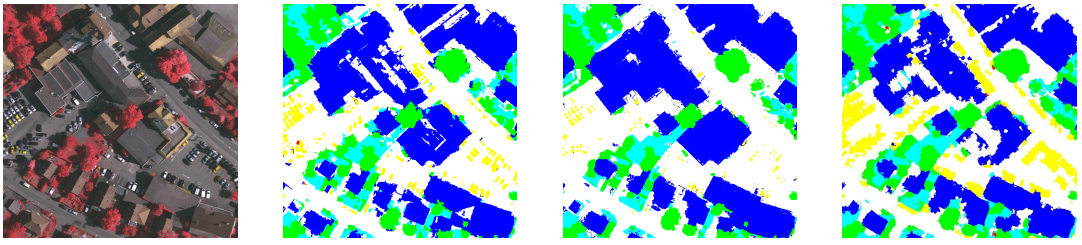
Fig. 3. Fully convolutional architecture for semantic labeling (SegNet [3]) of remote sensing data extracted from the ISPRS Vaihingen dataset.

Introduced in [16], fully convolutional networks (FCN) are designed for dense prediction instead of flat classification. Fully connected layers are replaced by convolutions that keep the 2-dimensionality of the data (cf. Fig. 3). Therefore, we can train such a network to classify all pixels of the image. We show that this approach is state-of-the-art for semantic labeling of remote sensing data.

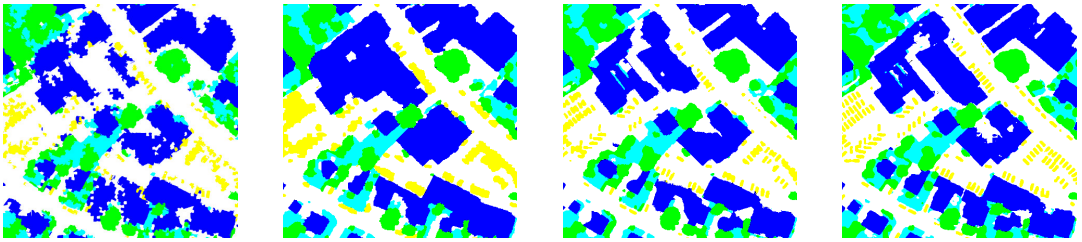
A. Deep network architecture

We use the SegNet architecture from [3]. SegNet uses an encoder-decoder architecture (cf. Fig. 3). The encoder is based on VGG-16 [6], in which convolutions are followed by a batch normalization and a ReLU ($max(0, x)$). Blocks of convolutions end with a max-pooling layer. The decoder is mirrored from the encoder with pooling replaced by unpooling. The unpooling layer unpacks the previous layer's activations at the indices corresponding to the maximum activations computed in the associated encoder pooling layer, and upsamples by padding with zeroes everywhere else. This relocates abstracted activations at the saliency points detected by the low level filters, thus increasing the spatial accuracy of the semantic labeling.

SegNet weights are initialized using VGG-16 trained on ImageNet and the decoder weights are randomly initialized. We train the network using SGD with a learning rate of 0.1 and a momentum of 0.9, and we divide the learning rate by 10 every 5 epochs.



(a) IRRG data (b) "SVL" [10] (c) RF + CRF [21] (d) Multiscale (ours, section II)



(e) Context (ours, section III) (f) CNN + RF + CRF [20] (g) "DLR" (FCN + CRF)[17] (h) SegNet (ours, section IV)

Fig. 4. Semantic mappings from several methods on an extract of the ISPRS testing set of Vaihingen

B. Results

Tiles from ISPRS dataset ($\approx 2500 \times 2500$) are processed using a 128×128 sliding window with a stride of 32px. Besides memory management reasons, we average the overlapping predictions to produce a smoother final map.

On the validation set, the overall accuracy reaches 89.11% and a F1 score of 80.57% on cars (to compare with the results from Tab. I). Compared to our previous works using superpixel and deep features (cf. Sec. II), SegNet predictions are more detailed, especially on cars where each instance is clearly segmented. SegNet is also more accurate on buildings, confusing less often roads and buildings than previous CNN and FCN. Moreover, our method even outperforms competitors using hand-crafted features and structured models such as Conditional Random Fields (CRF). A qualitative comparison of several methods is provided in Fig. 4.

TABLE I
F1 MEASURES PER CLASS AND OVERALL ACCURACY OF VARIOUS WORKFLOWS FOR SEMANTIC LABELING.

Approach	Imperv. surface	Building	Low veget.	Tree	Car	Overall accur.
Monoscale 32×32 (II)	81.26	81.58	62.71	77.88	40.10	76.33
Monoscale 64×64 (II)	81.13	82.36	62.46	76.13	41.03	75.98
Context w/ 32×32 (III)	82.00	82.40	58.18	78.38	32.46	78.36
Multiscale (section II)	85.04	89.28	72.50	81.66	61.93	82.41
SegNet (section IV)	92.96	94.57	83.93	81.64	80.57	89.11

V. Normal Estimation

Estimating normals in a point cloud, i.e., the local orientation of the unknown underlying surface, is a crucial first step for numerous algorithms, such as surface reconstruction and scene understanding. Many methods have been proposed for that, e.g., based on regression [11], Voronoï diagrams [8], sample consensus [15] or Hough transform [4]. These methods have different sensitivities to the presence of edges on the surface (not to oversmooth it) as well as to point outliers, to sampling noise and to variations of point density, which are common issues due to the way point clouds are captured.

We propose a novel method [5] for normal estimation in unorganized point clouds, based on a deep neural network. It is robust to noise, outliers, density variation and sharp edges, and it scales well to millions of points. It outperforms most of the time the state-of-the-art of normal estimation.

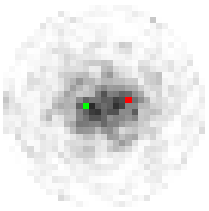
We first generate normal hypotheses as in [4], randomly picking triplets of points in a given neighborhood, which defines tentative tangent planes and thus possible normal directions. In a usual Hough-based setting as [4], each direction hypothesis votes in a problem-specific accumulator (a spherical map) and the estimated normal is computed from the most voted bin of the accumulator. This approach has good robustness properties but is sensitive to bin discretization. In our work, rather than blindly go for the most voted bin, which can be wrong, especially when close to edges or in presence of density variation, we let a trained CNN make the decision.

For this, we build an image-like accumulator representing possible directions by projecting the sphere on a plane and normalizing its orientation. It is a 33×33 regular grid that is much less discretized than the sphere in [4]. Fig. 5a shows an accumulator filled from a noisy point cloud: the green dot indicates the actual normal coordinates, which differs from the maximum of the distribution, marked with the red dot. Besides, to deal with density variation, we do not pick triplets uniformly but according to local density. Last, to reduce the sensitivity to scale, we consider a multiscale neighborhood analysis, actually creating a multicanal tensor input, like RGB channels for processing color images in CNNs.

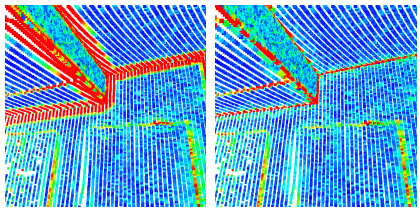
We train our network using synthetic ground-truth data. The

training set consists of point clouds randomly sampled on artificial sharp corners created with random angles. The network is based on LeNet [14]. It is composed of 4 convolutional layers and 2 max poolings followed by 4 fully-connected layers. The last regression layer learns and predicts the 2 angles which represent the 3D direction of the normal.

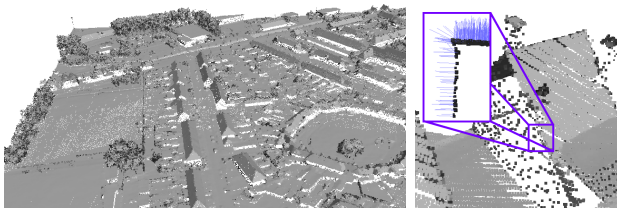
Fig. 5b illustrates normal estimation errors on an indoor LiDAR scene. Colors represent the intensity of the deviation from the ground truth (red indicates an error greater than 10°). An experiment on aerial LiDAR data is shown on Fig. 5c. The left image represents a tile with 2.3M points from the Data Fusion Contest 2015 [12]. The gray shade at each point depends on the illumination, and thus on the normal orientation. The right image details a case of very high density variation, as roofs are much more densely sampled than facade walls. Details on the method and more quantitative results are presented in [5].



(a) Accumulator



(b) Angular error: regression (left), ours (right)



(c) Aerial LiDAR: large tile (left), robustness to density variation (right)

Fig. 5. Normal estimation.

VI. Conclusion

We presented new approaches for producing better Earth Observation products. First, starting with aerial or satellite images, we use deep convolutional neural networks for semantic labeling and for the production of accurate thematic maps (using multiscale or context classification and dense segmentation). Second, with LiDAR point clouds as input, we show that neural networks can be used as a beneficial alternative to purely geometric procedures for estimating normals, a preliminary to shape extraction. Our results show that deep learning can diffuse to various topics of remote sensing and give alternate ways to deal with old problems.

VII. Acknowledgments

The authors would like to thank the Belgian Royal Military Academy for acquiring and providing the Zeebrugge dataset used in this study, and the IEEE GRSS Image Analysis and Data Fusion Technical Committee. The Vaihingen dataset was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) [7] : http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html. N. Audebert's work is funded by ONERA-TOTAL research project Naomi. The research of A. Boulch and B. Le Saux leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. FP7-SEC-607522 (Inachus Project)

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE T. Pattern Anal. and Mach. Intelligence*, 34(11):2274–2282, 2012.

[2] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. How Useful is Region-based Classification of Remote Sensing Images in a Deep Learning Framework? In *Proc. of IGARSS*, Beijing, China, 2016.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[4] Alexandre Boulch and Renaud Marlet. Fast and robust normal estimation for point clouds with sharp features. *Computer Graphics Forum*, 31(5):1765–1774, 2012.

[5] Alexandre Boulch and Renaud Marlet. Deep learning for robust normal estimation in unstructured point clouds. *Computer Graphics Forum*, 35(5):281–290, 2016.

[6] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *Proc. of BMVC*, 2014.

[7] M. Cramer. The dgpf test on digital aerial camera evaluation overview and test design. *Photogramm. Fernerkundung Geoinf.*, 2:73–82, 2010.

[8] Tamal K Dey and Samrat Goswami. Provable surface reconstruction from noisy samples. In *Symposium on Computational Geometry (SoCG)*, pages 330–339, 2004.

[9] Lagrange *et al.* Benchmarking classification of earth-observation data: from learning explicit features to convolutional networks. In *Proc. of IGARSS*, Milano, Italy, 2015.

[10] Markus Gerke. Use of the Stair Vision Library within the ISPRS 2d Semantic Labeling Benchmark (Vaihingen). Technical report, International Institute for Geo-Information Science and Earth Observation, 2015.

[11] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *ACM SIGGRAPH Computer Graphics*, 26(2):71–78, 1992.

[12] IEEE GRSS DFTC. 2015 IEEE GRSS data fusion contest. http://www.grss-ieee.org/community/technical-committees/data-fusion, 2015.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Adv. in Neural Info. Proc. Sys. 25*, pages 1097–1105, 2012.

[14] Y. LeCun, L. Bottou, Y. Bengio, , and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Bao Li, Ruwen Schnabel, Reinhard Klein, Zhiquan Cheng, Gang Dang, and Shiyao Jin. Robust normal estimation for point clouds with sharp features. *Computer & Graphics*, 34(2):94–106, 2010.

[16] J. Long, E. Shelhamer, and T. Darell. Fully convolutional networks for semantic segmentation. In *Proc. of CVPR*, 2015.

[17] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla. Semantic Segmentation of Aerial Images with an Ensemble of CNNs. *ISPRS Annals of Photogram., Rem. Sens. and Spatial Info. Sc.*, 3:473–480, 2016.

[18] Volodymyr Mnih and Geoffrey Hinton. Learning to detect roads in high-resolution aerial images. In *Proc. of ECCV*, Crete, Greece, 2010.

[19] Sebastian Nowozin and Christoph Lampert. Structured learning and prediction in computer vision. *Found. Trends. Comput. Graph. Vis.*, 6(3–4):185–365, March 2011.

[20] S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. Van-Den Hengel. Effective semantic pixel labelling with convolutional networks and conditional random fields. In *Proc. of CVPRw/Earth-Vision*, 2015.

[21] Nguyen Tien Quang, Nguyen Thi Thuy, Dinh Viet Sang, and Huynh Thi Thanh Binh. An Efficient Framework for Pixel-wise Building Segmentation from Aerial Images. In *Proceedings of 6th ISoICT*, page 43, 2015.

[22] Franz Rottensteiner, Gunho Sohn, Markus Gerke, and Jan Dirk Wegner. J. of Photogramm. and Rem. Sens.*: Special issue on Urban object detection and 3D building reconstruction*, volume 93. July 2014.