HOW USEFUL IS REGION-BASED CLASSIFICATION OF REMOTE SENSING IMAGES IN A DEEP LEARNING FRAMEWORK ?

Nicolas Audebert[1, 2], Bertrand Le Saux [1], Sébastien Lefèvre [2]

[1] ONERA, *The French Aerospace Lab*, F-91761 Palaiseau, France
[2] Univ. Bretagne-Sud, UMR 6074, IRISA, F-56000 Vannes, France
Emails: {nicolas.audebert, bertrand.le_saux}@onera.fr, sebastien.lefevre@irisa.fr

ABSTRACT

In this paper, we investigate the impact of segmentation algorithms as a preprocessing step for classification of remote sensing images in a deep learning framework. Especially, we address the issue of segmenting the image into regions to be classified using pre-trained deep neural networks as feature extractors for an SVM-based classifier. An efficient segmentation as a preprocessing step helps learning by adding a spatially-coherent structure to the data. Therefore, we compare algorithms producing superpixels with more traditional remote sensing segmentation algorithms and measure the variation in terms of classification accuracy. We establish that superpixel algorithms allow for a better classification accuracy as a homogenous and compact segmentation favors better generalization of the training samples.

Index Terms— Remote sensing, Segmentation algorithms, Image classification, Deep learning, Superpixels

1. INTRODUCTION

Last years have seen the rise of deep learning approaches for computer vision and remote sensing is not an exception. However, deep networks are not designed to directly process high resolution images such as the ones used in remote sensing. These models are therefore used by focusing on the local appearance around a given location. For performance reasons, a preprocessing step dividing the image into coherent small regions is needed, and therefore arise the need for segmentation. Two main approaches are used in the litterature : sliding windows and image segmentation. Superpixel segmentations gained lots of interest in the context of remote sensing when it was used to establish state-of-the-art performances (both in classification accuracy and processing time) [1].

In this paper, we investigate what makes a good segmentation algorithm for classification purposes. Two aspects are evaluated: the pure segmentation quality (well-defined boundaries, coherence of the pixels inside a region) and the impact of the segmentation on classification through the size and shape of the regions (i.e the classification samples).

2. CLASSIFICATION FRAMEWORK

Deep learning has been the state-of-the-art in computer vision for a few years. Neural networks operate at the pixel level by simultaneously learning which features to extract and how to classify them. Convolutional neural networks are composed of layers of neurons computing convolutions on the previous layer outputs. These layers are stacked and combined with max-pooling (i.e. sampling maximum activations) and elementwise non-linear transfer functions (e.g. $tanh$) to extract high order features from the input. The network then produces a probability vector, on which a softmax is applied to predict the output label. This has been proven to be an effective baseline for computer vision tasks in [2].

Our framework uses the well-known AlexNet [3] architecture as a feature extractor, as [4] showed that the deep features extracted from AlexNet could be effectively transfered for remote sensing tasks. Patches extracted from the image are passed through the network and the last layer outputs before the softmax are used as feature vectors. More precisely, our framework (Fig. 1) achieves semantic segmentation with the following pipeline :

1. Divide the image into small regions using a segmentation algorithm.
2. For each region, extract 32×32, 64×64 and 128×128 patches centered on the region.
3. Resize all patches to 228×228 and process them through AlexNet.
4. Concatenate the resulting vectors to produce one feature vector (sample).

At training time, we process the images of the training set, for which we have the associated ground truth. We define the label of a region with a majority vote according to the associated ground truth. We then use the training set of newly acquired features to train a linear Support Vector Machine (SVM), whose parameters are optimized by stochastic gradient descent. At testing time, we use the SVM to predict the label of each region of the image to be classified, and then associate to all pixels in this region the predicted output label. In the end, we obtain a semantic map that we can compare to the ground truth.

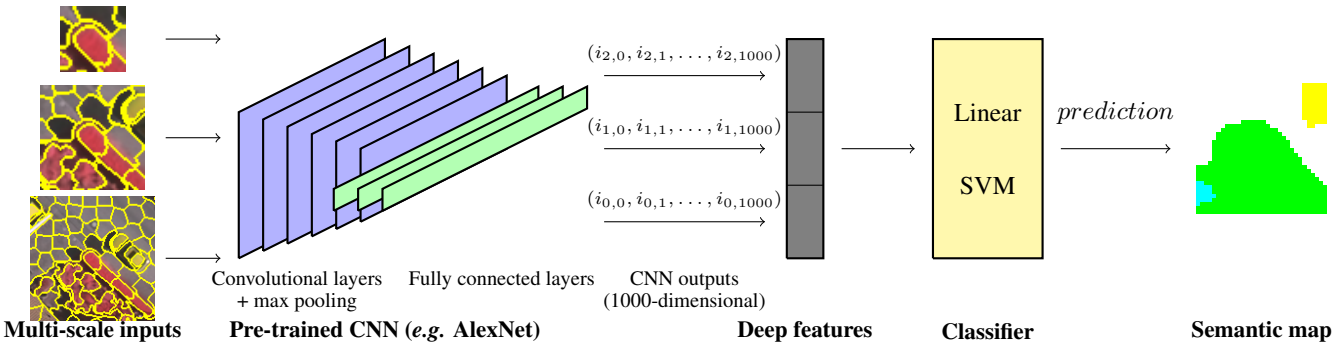


Fig. 1: Classification framework using deep multi-scale features

3. SEGMENTATION ALGORITHMS

To avoid discrepancies in the training samples, the segmented regions should be similar in shape and size. This motivates the use of superpixel algorithms rather than traditional segmentation ones. Indeed, the latter, both from the remote sensing and computer vision communities, create very inhomogenous regions in shape and size. This does not bode well with our multiscale framework, that expects similarly shaped training samples. Moreover, superpixel algorithms have been used successfully in the remote sensing literature [5]. Therefore, we choose to evaluate the following superpixel algorithms:

- **SLIC** (Simple Linear Iterative Clustering) [6]: starts from a grid and creates a segmentation by iteratively growing the regions by applying a k-means algorithm.
- **LSC** (Linear Spectral Clustering) [7]: embeds the image in a 14-dimension space and increase each region using weighted k-means starting from a grid.
- Quickshift [8]: clusters points belonging to the same dominant mode in a non-Euclidean color-(x,y) space, using the Lab color space.

We also test two popular segmentation algorithms from the remote sensing community:

- **MRS** (Multiresolution Segmentation) [9]: implemented in the eCognition software, MRS clusters points using a well-defined homogeneity criterion based on spatial and spectral information.
- **HSEG** (Hierarchical image Segmentation) [10]: based on Hierarchical Step-Wise Optimization (HSWO) with spectral clustering, HSEG builds a hierarchical segmentation using a dissimilarity criterion. We extract the most detailed segmentation using the RHSEG implementation.

As a baseline, we compare these segmentations to a sliding window (SW) approach. The window parameters are chosen to obtain as many windows as there are regions using the previously described algorithms to achieve the same processing time.

4. EXPERIMENTS

4.1. Experimental setup

The algorithms are tested on the ISPRS 2D Semantic Labeling Dataset [11]. We use part of the Vaihingen data, consisting of 16 IR-R-G orthoimages with pixel-level ground truth. We compare the segmented images to the ideal segmentation represented by the ground truth.

Segmentation algorithms are evaluted by several standard metrics proposed by [12]:

- The Undersegmentation Error (**UE**): defined as the ratio of pixels belonging to a region overlapping other regions. Formally, if respectively S, P and N denote the regions in the ground truth, the segmented regions and the number of pixels in the image:

$$UE = \frac{1}{N} \sum_{S \in GT} \sum_{P : P \cap S \neq \emptyset} min(|P \cap S|, |P \backslash P \cap S|)$$

- The Boundary Recall (**BR**): the recall of boundary pixels in the 3-pixel neighborhood of the ground truth boundaries :

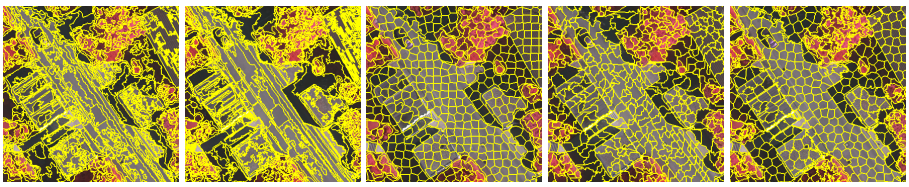
$$BR = \frac{true\ pos.}{true\ pos. + false\ neg.}$$

- The Average Purity (**AP**): average percentage of pixels of a region belonging to the region dominant class. Let avg and maj denote respectively the average function and the majority class:

$$AP = \underset{P \in seg}{avg} \left(\frac{|P \cap maj(P)|}{|P|} \right)$$

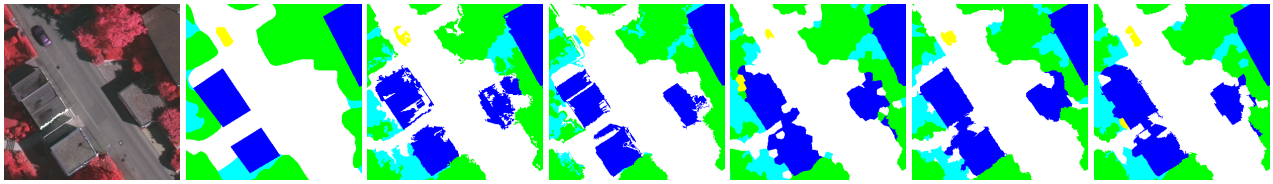
- The oracle: the pixel-wise classification accuracy that would be achieved by a perfect classifier, assigning the majority class label to each segment. This is the best case scenario and therefore is the maximum accuracy that can be achieved with this segmentation.

We split this dataset as follow : tiles 1, 5, 7, 11, 17, 23, 26, 28, 34 and 37 form the training set, while tiles 13, 21



(a) MRS (b) HSEG (c) SLIC (d) Quickshift (e) LSC

Fig. 2: Regions segmented by the different segmentation algorithms (zoom on a specific location)



(a) Orthoimage (b) Ground truth (c) MRS (d) HSEG (e) SLIC (f) Quickshift (g) LSC

Fig. 3: Semantic maps after classification using different segmentation algorithms (zoom on a specific location)

and 30 form the validation set and tiles 3, 15 and 32 form the testing set. Note that the "clutter" class is not represented in the testing set. This is justified by the fact that the ISPRS evaluation procedure does not take this class into account.

In order to compare the classification results, we use the following metrics:

- The overall pixel-wise accuracy on the testing set.
- The κ coefficient (inter-rater agreement).
- The F1 score for the "car" class, as an additionnal performance indicator. This allows us to consider specifically the problem of object detection.

Note that the segmentation parameters for each algorithm were chosen to achieve the best overall accuracy as we wish to optimize the classification performance, using roughly the same number of regions.

4.2. Results

The segmentation metrics of the evaluated algorithms are presented in Tab. 1. Using only this information, MRS and HSEG seem to be competitive with superpixel algorithms on most metrics.

The classification results for each algorithm are presented in Tab. 2. The superpixel algorithms obtain very similar results as the classification accuracy shows little variation w.r.t. the segmentation. However, these results establish an advantage of superpixels over traditional segmentations. Indeed, MRS and HSEG are lagging behind the superpixel methods on the classification accuracy and do not bring any additional gain compared to the baseline sliding window approach. This can be explained by the segmentation's geometrical properties. Superpixels tend be strongly convex and compact, while traditional segmentations usually produce very heterogenous regions in shape and size. However, better learning is achieved when the training samples are similar, as the classifier does not need to infer which pixels are meaningful in the example patch. The parameters achieving best classification accuracy for MRS back this result. To reach the accuracy presented in Tab. 2, the compacity parameter of MRS has to be significantly increased and the resulting segmentation is more homogenous and "superpixel-looking". However, MRS needs significantly more segments than the superpixel algorithms (especially efficient ones such as SLIC) – which comes at the cost of a higher processing time – while the accuracy is still lower than with superpixel algorithms.

This is partially illustrated in Fig. 2 and Fig. 3. MRS and HSEG are more erratic and the semantic maps suffer from irregular shapes and borders. The interior of objects such as cars and buildings is often attributed to the wrong class since the inside pixels do not belong to the same regions as the sucessfully classified ones. Superpixel algorithms tend to preserve more truthfully the shape and convexity of objects.

Furthermore, there is no direct link between the theoretical best-case (the oracle) and the actual accuracy. LSC has a lower oracle than SLIC on the dataset, but beats it in the at testing time. This means that the choice of the algorithm not only impacts the segmentation but also the information learned by the classifier. Indeed, the shape and size of the superpixels directly alter the samples provided to the SVM. This supports the idea that the homogeneity of the superpixels is crucial in this classification framework.

Finally, according to the resulting F1 scores on the "car" pixels, object detection can greatly be improved by chosing an appropriate segmentation algorithm. Best results on this class are obtained with LSC, even if results are tight.

Algorithm	Regions	UE (%)	BR (%)	AP (%)	Oracle (%)
SLIC	**20 000**	**10.21**	84.07	75.10	89.91
LSC	22 800	11.37	91.13	71.54	85.83
Quickshift	21 000	11.66	88.34	72.90	83.61
MRS	23 500	13.12	**95.71**	**79.08**	**91.68**
HSEG	21 000	11.39	94.83	78.66	85.25

Table 1: Segmentation metrics on the ISPRS dataset

Algorithm	Regions	Acc. (%)	F1_car	κ
SLIC	**20 000**	82.20	0.54	0.76
LSC	22 800	**82.45**	**0.58**	0.76
Quickshift	21 000	82.05	0.52	0.75
MRS	23 500	80.53	0.56	0.73
HSEG	21 000	79.56	0.54	0.72
SW	23 800	81.22	0.53	0.74

Table 2: Classification metrics on the ISPRS dataset

5. CONCLUSION

In this work, we have aimed to establish that superpixel algorithms provide adequate segmentations for classification of remote sensing images in a deep learning framework. There is no clear universal advantage of using one particular superpixel segmentation method. This depends on the nature of the data, notably if distinguishing objects significantly smaller than others, such as cars compared to buildings, is needed.

This comparison brings new insights on how samples should be extracted from remote sensing data in order to achieve semantic segmentation, i.e segmentation and classification of the regions through a deep learning framework. Superpixel algorithms provide the classifier with compact and homogeneously segmented samples that favors generalization of the learned content. This allows for a better accuracy with fewer samples and a reduced processing time.

Finally, our work shows that there is no direct link between the quality of the segmentation according to the standard metrics (boundary adherence, etc.) and the pixel-wise classification accuracy. Therefore, chosing a segmentation algorithm should be based solely on the classification accuracy achieved, as the impact of the shape, size and homogeneity of the segments is preponderant for training a classifier.

6. ACKNOWLEDGEMENTS

The Vaihingen dataset was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) (`http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html`). Nicolas Audebert's work is supported by the Total-ONERA project NAOMI.

7. REFERENCES

[1] S. Paisitkriangkrai et al., "Effective semantic pixel labelling with convolutional networks and Conditional Random Fields," in *IEEE Conf. on Comp. Vis. and Patt. Rec. Workshops (CVPRW)*, June 2015, pp. 36–43.

[2] A.S. Razavian et al., "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," in *IEEE Conf. on Comp. Vis. and Patt. Rec. Workshops (CVPRW)*, June 2014, pp. 512–519.

[3] A. Krizhevsky et al., "ImageNet Classification with Deep Convolutional Neural Networks," in *Adv. in Neural Info. Proc. Sys. 25*, 2012, pp. 1097–1105.

[4] A. Lagrange et al., "Benchmarking classification of Earth-observation data: From learning explicit features to convolutional networks," in *Int. Geosci. and Remote Sens. Symp. (IGARSS), IEEE*, July 2015, pp. 4173–4176.

[5] Z. Wu et al., "Superpixel-Based Unsupervised Change Detection Using Multi-Dimensional Change Vector Analysis and Svm-Based Classification," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 7, pp. 257–262, July 2012.

[6] R. Achanta et al., "SLIC superpixels," Tech. Rep., 2010.

[7] Z. Li and J. Chen, "Superpixel Segmentation Using Linear Spectral Clustering," in *IEEE Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2015, pp. 1356–1363.

[8] A. Vedaldi and S. Soatto, "Quick Shift and Kernel Methods for Mode Seeking," in *Eur. Conf. on Comp. Vis. (ECCV)*, Oct. 2008, pp. 705–718.

[9] M. Baatz and A. Schäpe, "Multiresolution Segmentation: an optimization approach for high quality multi-scale image segmentation," *Angewandte Geographische Informationsverarbeitung XII: Beiträge zum AGIT-Symposium Salzburg*, pp. 12–23, 2000.

[10] J. Tilton et al., "Best Merge Region-Growing Segmentation With Integrated Non-adjacent Region Object Aggregation," *Trans. on Geosci. and Remote Sens.*, vol. 50, no. 11, pp. 4454–4467, Nov. 2012.

[11] F. Rottensteiner et al., "The ISPRS benchmark on urban object classification and 3D building reconstruction," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 1, pp. 3, 2012.

[12] P. Neubert and P. Protzel, "Superpixel benchmark and comparison," in *Proc. Forum Bildverarbeitung*, 2012, pp. 1–12.