ACTIVE LEARNING ON LARGE HYPERSPECTRAL DATASETS: A PREPROCESSING METHOD

R. Thoreau[1,2], V. Achard[1], L. Risser[3], B. Berthelot[2], X. Briottet[1]*

[1] ONERA-DOTA, University of Toulouse, 31055 Toulouse, France - (romain.thoreau, veronique.achard, xavier.briottet)@onera.fr
[2] Magellium, 31520 Ramonville Saint-Agne, France - beatrice.berthelot@magellium.fr
[3] Toulouse Mathematics Institute (UMR 5219), CNRS, University of Toulouse, 31062 Toulouse, France - lrisser@math.univ-toulouse.fr

KEY WORDS: Airborne hyperspectral imaging, semantic segmentation, active learning

ABSTRACT:

Machine learning algorithms demonstrated promising results for hyperspectral semantic segmentation. However, they strongly rely on the *quality* of training datasets. As far as the annotation of hyperspectral images is often expensive and time-consuming, only a few thousand pixels can be labeled. In this context, active learning algorithms select the most informative pixels to be labeled. In the machine learning community, recent active learning methods have overcome the performance of conventional algorithms but do not always scale to large remote sensing images. Therefore, we introduce in this paper a preprocessing method that allows the use of computationally intensive active learning algorithms without significant impacts on their effectiveness.

1. INTRODUCTION

Airborne hyperspectral imaging with high spectral and spatial resolutions is well suited to map the land cover of urban areas. Applications of land cover maps include the mitigation of urban heat island effects (Zhou et al., 2017) or urban management (Fox et al., 2012). Many machine learning models have been developed to automatically segment hyperspectral images and have demonstrated interesting results (Audebert et al., 2019). However, the diversity of soil materials, the spectral intra-class variabilities, the inter-class similarities, the presence of shadows or of small irrelevant objects (cf fig. 1a) typically hinder the use of machine learning models to large and complex images.

In order to further simplify the automation of land cover mapping with machine learning models, representative training datasets are crucial. Therefore, building an optimal training dataset is a critical step. Labeling hyperspectral images, though, is often hard and expensive. Field campaigns, alongside time-consuming photo-interpretation by experts, can be necessary. Thus, the annotation of images is often limited to a few thousand pixels.

In this context, active learning (AL) methods guide the data annotation, answering the following question: out of millions of pixels, which ones to annotate to quickly improve classification performances? Active learning methods are iterative algorithms that select at each step the most informative samples to be labeled, given an initial training dataset and a classifier (Settles, 2012). An oracle (a user) then labels the given pixels that are added to the training dataset. Many strategies have been developed (Tuia et al., 2011) with various computational requirements. Recently, (Sener and Savarese, 2017) tackled the active learning problem as a coreset problem, showing very high performance on several state-of-the-art machine learning datasets. However, it does not scale to larger hyperspectral datasets such as the Houston image (Prasad et al., 2020), because of its high memory footprint and computational burden. Thus, we introduce in the present paper a preprocessing method that allows the use of such computationally intensive active learning algorithms on large hyperspectral scenes, without significant loss of effectiveness. The paper is organized as follows. In section 2, we describe the active learning framework and present our preprocessing method. In section 3, we present the results of our numerical experiments. Finally, we discuss the results and conclude in section 4.

2. METHOD

2.1 Active Learning Framework

First, we rigorously describe the active learning framework. We denote the reflectance spectrum of one pixel by $s \in \mathcal{S} = [0,1]^B$ and its class $y \in \{1, ..., c\}$ where B is the number of spectral bands and c is the number of classes. The active learning algorithm is defined by a number of steps N_{steps}, a budget b of pixels to be labeled at each step and an acquisition function $a : \mathcal{S}^b \longrightarrow \mathbb{R}$ that takes as input a subset of b pixels and measures how much information its annotation brings.

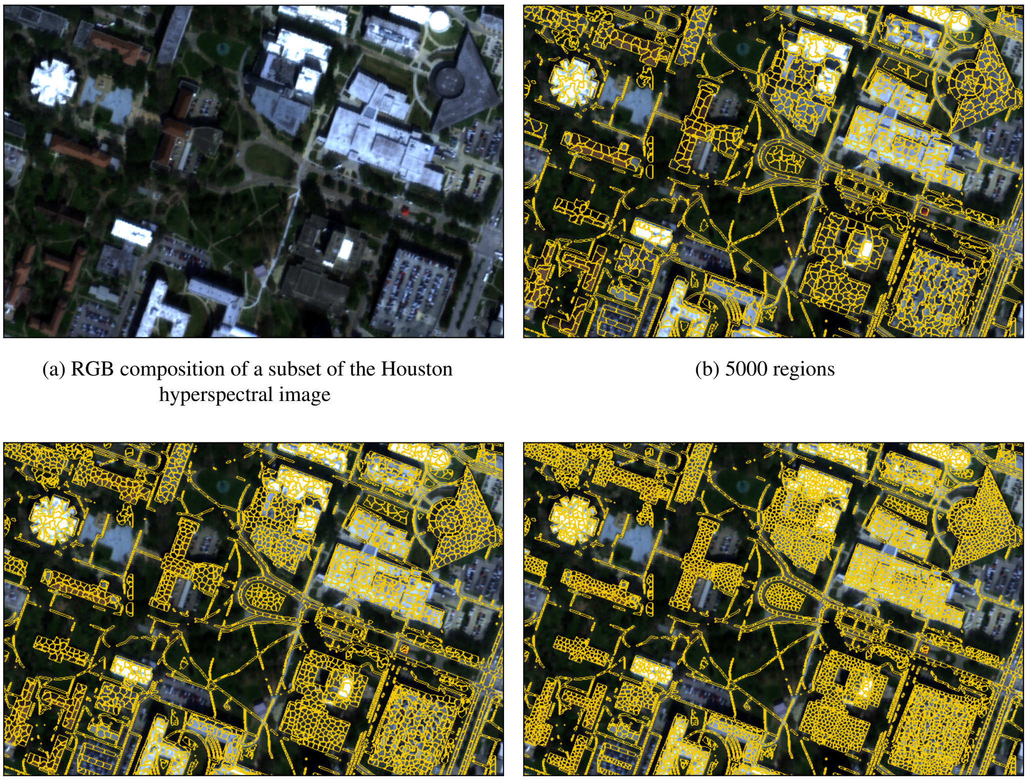
The active learning process (Settles, 2012) works as follows. From a dataset $S = (s_i)_{i \in (1,...,N)}$ of N pixels, an initial training dataset $L_0 = (S_0, Y_0) = (s_i, y_i)_{i \in (1,...,N_0)}$ is manually labeled. Then, at each step t, b unlabeled pixels $\{s_1^*, ..., s_b^*\}$ are queried from the current unlabeled pool $U_t = S \backslash S_t$ so that the acquisition function a is maximized. The acquisition function is parametrized by the labeled and unlabeled dataset, L_t and U_t. Then, an oracle provides the true labels $\{y_1, ..., y_b\}$. In this paper, we keep the acquisition function as a black box.

2.2 Data preprocessing: computational cost reduction

To deal with large datasets, we introduce a preprocessing method that relies on the segmentation of the image in superpixels. The most informative superpixels are selected. Then, a subset of randomly drawn pixels within these superpixels are labeled and added to the training dataset. To perform the image segmentation, we chose the SLIC algorithm (Achanta et al.,

* Corresponding author: romain.thoreau@onera.fr

https://doi.org/10.5194/isprs-archives-XLIII-B3-2022-435-2022 | © Author(s) 2022. CC BY 4.0 License.



(a) RGB composition of a subset of the Houston hyperspectral image

(b) 5000 regions

(c) 10000 regions

(d) 25000 regions

Figure 1. Visualization of different SLIC segmentations (using the first three components of a PCA) on a Houston unlabeled pool

2012) because it is fast, memory efficient, shows state-of-the-art performance and allows to control the amount of superpixels.

In the following, we present the preprocessing algorithm in detail and briefly describe the SLIC algorithm.

2.2.1 Preprocessing algorithm

Our preprocessing method works as follows:

1. The dimension of the image is reduced by averaging over every bands or by using a Principal Component Analysis (PCA),
2. The unlabeled pool is segmented in $N_{regions}$ using the SLIC algorithm,
3. The mean spectrum of each region is computed, resulting in an unlabeled pool $\bar{U}$ of size $N_{regions}$,
4. Regions containing less than N_{th} pixels are left out in order to avoid outliers, resulting in $\bar{U}'$.
5. The active learning algorithm is applied on $\bar{U}'$. In order to avoid too much redundancy, only m pixels out of the total number of pixels in the selected clusters are labeled. They are randomly picked.

Considering superpixels instead of pixels themselves yields to major decreases of computational requirements. In our experiments (cf. section 3), we compared the active learning performances with and without preprocessing. Moreover, this strategy opens the door to interesting properties such as robustness to outliers and easier visualization for the oracle.

2.2.2 SLIC algorithm (Achanta et al., 2012)

The SLIC algorithm (Simple linear iterative clustering) is based on the k-means algorithm. k clusters are initialized so that they spatially spread in a homogeneous way on the image. If N is the number of pixels in the image, then the initial size of a superpixel is $S^2 = \frac{N}{k}$. Then, each pixel is associated with the nearest cluster in a $2S \times 2S$ neighborhood using the euclidean distance. For a given number of iterations, cluster centers are updated and each pixel is reassigned to a new cluster. Usually, few iterations are needed for the cluster centers to converge. For more details, we refer the reader to (Achanta et al., 2012). Compared to other superpixels generation methods, SLIC is at the same time simple, memory efficient and fast.

3. EXPERIMENTS

The objectives of our experiments are to:

- Empirically demonstrate that the preprocessing method dramatically reduces the active learning calculation time without significant loss of effectiveness,
- To study the impact of the superpixels size on the active learning performance,
- To study the impact of the dimensionality reduction technique on the active learning performance.

3.1 Dataset

We carried out experiments on the Houston dataset (Prasad et al., 2020). It is a hyperspectral image acquired by ITRES CASI

1500 that covers the 380-1050 nm spectral range with 48 bands at a 1m ground sampling distance. More than four hundred thousands pixels were labeled over 20 classes (see table 1). However, we only kept in the initial training dataset 200 labeled pixels per class, *i.e.* 4000 pixels in total, which is representative of an operational use case. Five sets of different initial training dataset, unlabeled pools and test datasets were selected from disjoint regions. Here we emphasize that, in this experiment, active learning methods are applied on unlabeled pools with well-defined labels only. In a life-like scenario, active learning methods could query mixed pixels (pixels at the frontier of two materials) or pixels whose classes do not belong to the classes of the initial training dataset.

Table 1. Classes and the numbers of labeled pixels

Class Id	Class label	Number of pixels
1	Healthy grass	9927
2	Stressed grass	32585
3	Artificial turf	2425
4	Evergreen trees	16419
5	Deciduous trees	9398
6	Bare earth	6846
7	Water	673
8	Residential buildings	38271
9	Non-residential buildings	221147
10	Roads	41214
11	Sidewalks	28841
12	Crosswalks	2570
13	Major thoroughfares	44956
14	Highways	9696
15	Railways	9745
16	Paved parking lots	11623
17	Unpaved parking lots	957
18	Cars	4977
19	Trains	8596
20	Stadium seats	11782

3.2 Active learning methods

Three active learning methods were studied.

Coreset (Sener and Savarese, 2017) is the method of main interest because it showed impressive results (Sener and Savarese, 2017) on CIFAR (Krizhevsky et al., 2009) and SVHN (Netzer et al., 2011) but is not usable on the Houston dataset that is approximately five times larger. Coreset has large memory and time requirements as it computes and stores distances between labeled and unlabeled points. It has a subroutine (a k-center greedy algorithm) with a $\mathcal{O}(|U| \cdot b)$ time complexity (Gonzalez, 1985), where $|U|$ is the size of the unlabeled pool. In practice, it could not be applied within a reasonable time on the Houston dataset with our hardware[1].

In order to assess the impact of our preprocessing routine on the performance of active learning methods, we also studied **Breaking Tie (Tong Luo et al., 2004)** and **BALD (Houlsby et al., 2011)** that are two state-of-the-art AL methods. Both methods can be applied without preprocessing on the Houston image, which allowed us to compare the results with and without the preprocessing.

3.3 Hyperparameters

We ran the experiments with the following hyperparameters: $N_{steps} = 15$, $b = 250$, $N_{th} = 0$ and $m = 1$, where we recall that N_{steps} is the number of AL steps, b is the number of queried superpixels at each step, N_{th} is the minimal size of a superpixel and m is the number of randomly queried pixels within a superpixel. We performed 15 steps with 250-superpixels queries because it was enough for accuracy metrics to converge. As far as there are no abnormal objects in the Houston dataset, we left N_{th} to 0 but we argue that a non zero value would be interesting in life-like scenarios. Moreover, we have varied the number of superpixels within $\{20000, 10000, 5000\}$. We found that it yielded quite spectrally homogeneous and spatially consistent regions while dramatically reducing the size of the pool. Segmentation results, with a PCA dimensionality reduction, are shown on figures 1b, 1c and 1d, and segmentation results with a panchromatic dimensionality reduction, is shown on figure 3b.

3.4 Metrics

At each step of the AL process (15 steps in total), we trained a SVM classifier with a rbf kernel and kept track of three metrics:

- **Overall Accuracy (OA)**. The number of correct predictions over the total number of predictions,
- **Mean Intersect Over Union (mIoU)**. The mean of the IoU score over every classes. For one class, the IoU score is defined as $IoU = \frac{TP}{TP+FN+FP}$ where TP, FN and FP respectively denote the true positives, the false negatives and the false positives. A 0.5 IoU score means that there are as many good predictions as there are confusions,
- **The proportion of added pixels in each class** after steps 5, 10 and 15.

3.5 Results

3.5.1 Impact of the preprocessing on the computational requirements

If the decrease of the query time for Breaking Tie and BALD is not interesting in regard of the additional segmentation time, major time gains are achieved for Coreset as table 2 shows.

Table 2. Approximate time requirements in minutes with our hardware[1].

Number of regions	$\emptyset$	20000	10000	5000
Segmentation time		2.5	1.2	0.7
Query time	Coreset			
		37	5	1
	Breaking Tie			
	0.050	0.017	0.012	0.010
	BALD			
	0.21	0.02	0.015	0.013

3.5.2 Impact of the preprocessing on OA and mIoU

Overall Accuracy. Figures 2, 3 and 4 show that the preprocessing makes very little differences in terms of overall accuracy (approximately less than 2%). For each method, the preprocessing on the contrary slightly increases the overall accuracy.

Mean Intersect Over Union. Figures 2, 3 and 4 also show that the preprocessing makes very little differences in terms of mIoU score (approximately less than 2%), especially for

[1] Intel Cascade Lake CLX-6230 20c 2,1GHz and 64GB memory

Coreset and Breaking Tie. Moreover, the preprocessing reduces the standard deviation of the mIoU. Specifically, the larger the superpixels (ie the fewer regions there are), the lower is the standard deviation. The biggest gap is observed for the BALD method between the runs without preprocessing and with a 5000-regions preprocessing. To better understand this difference, we inspected the IoU score per class. It appeared that, with the preprocessing, the IoU score for *Residential buildings* and *Roads* was respectively increased from 0.17 to 0.40 and from 0.020 to 0.16.

Proportion of added pixels. Figures 5, 6 and 7 show that preprocessing, especially when large superpixels are used, tends to smooth the distribution of queried pixels over the classes. It is particularly pronounced for Coreset and for BALD where classes 3 (*Artificial turf*) and 7 (*Water*) are much less queried with a 5000-regions segmentation.

3.5.3 Influence of the dimensionality reduction technique on the segmentation

Experiments (shown on fig. 9 in Annex) demonstrate that the dimensionality reduction technique has low influence on the results. Using the panchromatic reduction technique yields to more squared regions whereas PCA enforces spectrally consistent regions.

4. DISCUSSION AND CONCLUSION

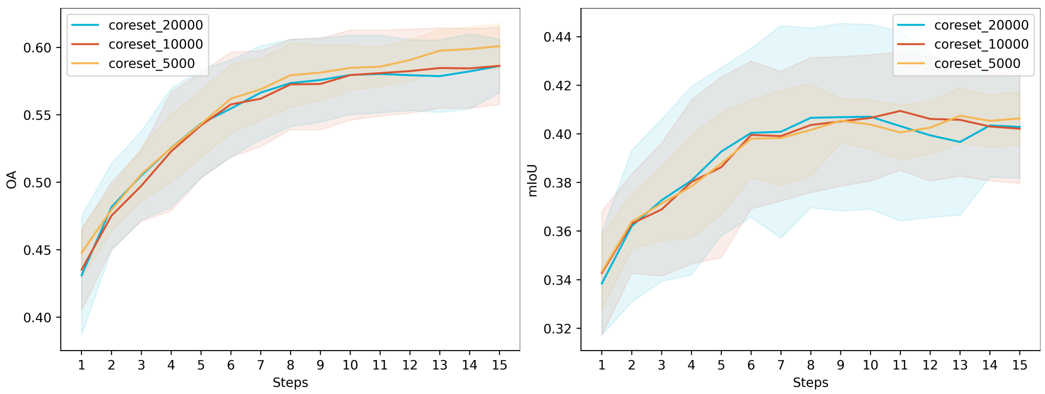
We introduced a preprocessing method that allows to use computationally intensive active learning algorithms on very large hyperspectral datasets without loss of effectiveness. This allows the use of methods that would normally have long calculation time during field campaigns, facilitating the annotation of experts. Besides, we argue that our method could be used on other kind of data, such as very high resolution satellite images.

The main side effect of our method is a reduction of redundancy alongside additional randomness. Firstly, the preprocessing reduces redundancy, because when a superpixel is highly informative according to the acquisition function, only few pixels within the superpixel (one in our experiments) are labeled. Without the preprocessing, almost every pixels within the superpixel would be labeled. This explains why much less pixels from classes *Artificial turf* and *Water* were queried by the BALD method. Secondly, as the superpixels sizes increase, their spectral heterogeneity increases and more randomness is introduced. As a matter of fact, pixels from different classes are more likely to belong to the same superpixel as its gets larger. Therefore, a 5000-regions segmentation yields more randomness than a 20000-regions segmentation. Nevertheless, even a 5000-regions segmentation yielded very homogeneous superpixels and the effect of this additional randomness did not seem to be significant.

Even without setting a minimum superpixel size, we think that our method increases the robustness to outliers. As a matter of fact, considering individual pixels is prone to outliers selection because they can exhibit anomalous spectral signatures that yield high uncertainty. On the contrary, outliers spectral signatures are mixed with in-distribution samples when we use superpixels. This is very likely the reason why BALD with a 5000-regions segmentation achieved better performances, especially for classes *Roads* and *Residential buildings*, that can include anomalous pixels (such as chimneys). BALD can indeed be sensitive to outliers as far as it selects samples far from training samples in the data space (Houlsby et al., 2011).

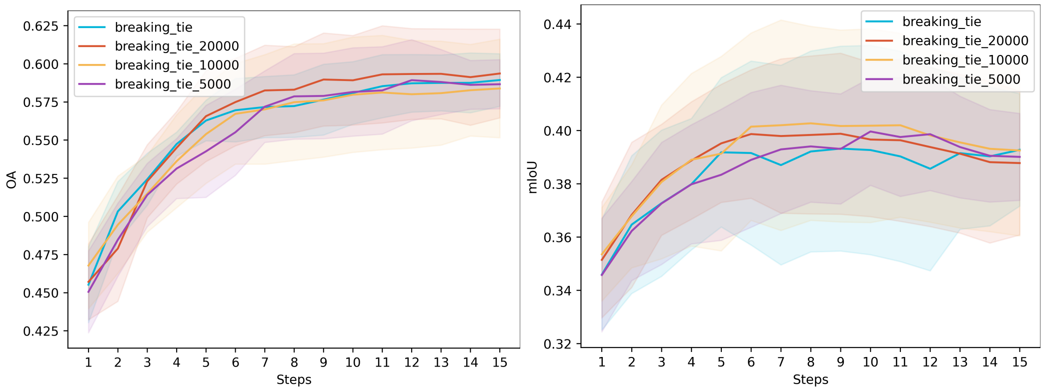
In further work, we could use superpixels to easily incorporate spatial and geometric features. Instead of randomly select pixels within a superpixel, we could select the pixel at the center of the superpixel to decrease the risk of drawing mixed pixels. We could also run an additional active learning step on the pixels of the selected superpixels and combine different AL methods.

https://doi.org/10.5194/isprs-archives-XLIII-B3-2022-435-2022



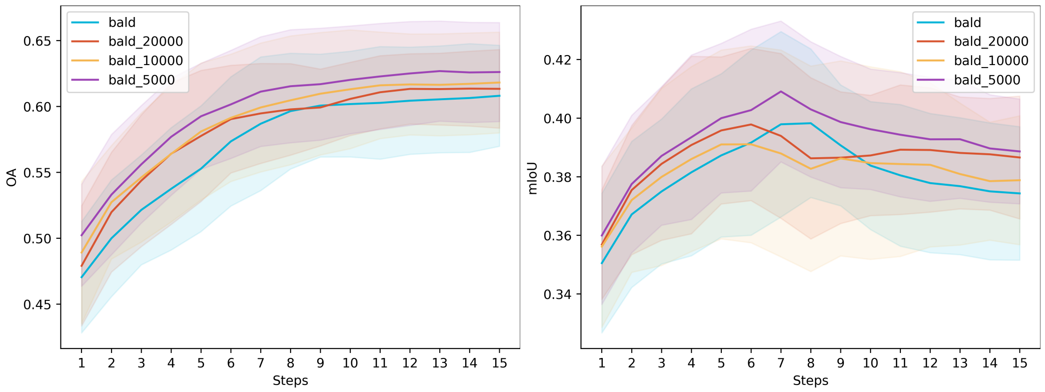
(a) OA (b) mIoU

Figure 2. Accuracy metrics over the **Coreset** process. *coreset* corresponds to the run without the data preprocessing. *coreset_20000*, *coreset_10000* and *coreset_5000* correspond respectively to the runs with a 20000, 10000 and 5000-regions preprocessing.



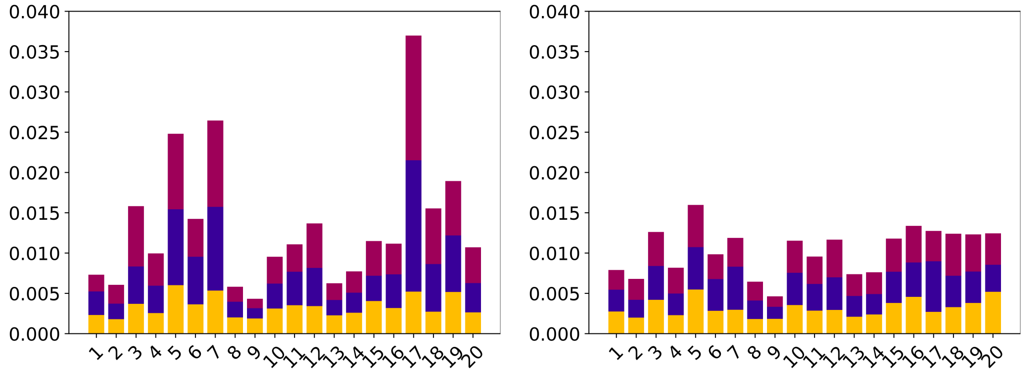
(a) OA (b) mIoU

Figure 3. Accuracy metrics over the **Breaking Tie** process. *breaking_tie* corresponds to the run without the data preprocessing. *breaking_tie_20000*, *breaking_tie_10000* and *breaking_tie_5000* correspond respectively to the runs with a 20000, 10000 and 5000-regions preprocessing.



(a) OA (b) mIoU

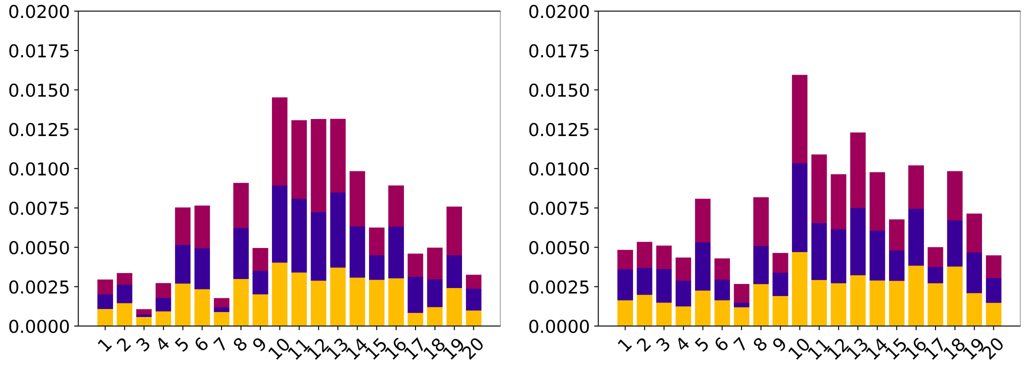
Figure 4. Accuracy metrics over the **BALD** process. *bald* corresponds to the run without the data preprocessing. *bald_20000*, *bald_10000* and *bald_5000* correspond respectively to the runs with a 20000, 10000 and 5000-regions preprocessing.



(a) 20000-regions preprocessing

(b) 5000-regions preprocessing

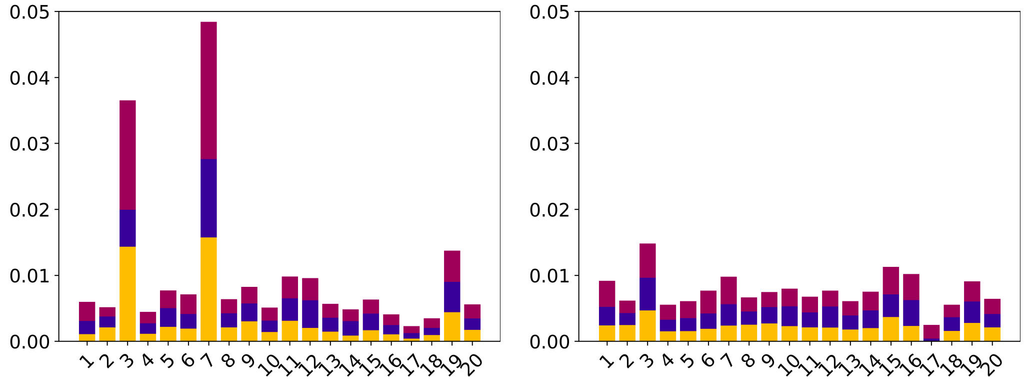
Figure 5. Proportion of added pixels for Coreset at steps 5, 10 and 15.



(a) No preprocessing

(b) 5000-regions preprocessing

Figure 6. Proportion of added pixels for Breaking Tie at steps 5, 10 and 15.



(a) No preprocessing

(b) 5000-regions preprocessing

Figure 7. Proportion of added pixels for BALD at steps 5, 10 and 15.

REFERENCES

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2012. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274-2282.

Audebert, N., Le Saux, B., Lefèvre, S., 2019. Deep Learning for Classification of Hyperspectral Data: A Comparative Review. *IEEE Geoscience and Remote Sensing Magazine*, 7(2), 159-173.

Fox, D., Witz, E., Blanc, V., Soulié, C., Penalver-Navarro, M., Dervieux, A., 2012. A Case Study of Land Cover Change (1950–2003) and Runoff in a Mediterranean Catchment. *Applied Geography*, 32, 810–821.

Gonzalez, T. F., 1985. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38, 293-306. https://www.sciencedirect.com/science/article/pii/0304397585902245.

Houlsby, N., Huszár, F., Ghahramani, Z., Lengyel, M., 2011. Bayesian active learning for classification and preference learning.

Krizhevsky, A., Hinton, G. et al., 2009. Learning multiple layers of features from tiny images.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y., 2011. Reading digits in natural images with unsupervised feature learning.

Prasad, S., Le Saux, B., Yokoya, N., Hansch, R., 2020. 2018 IEEE GRSS Data Fusion Challenge – Fusion of Multispectral LiDAR and Hyperspectral Data. https://dx.doi.org/10.21227/jnh9-nz89.

Sener, O., Savarese, S., 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

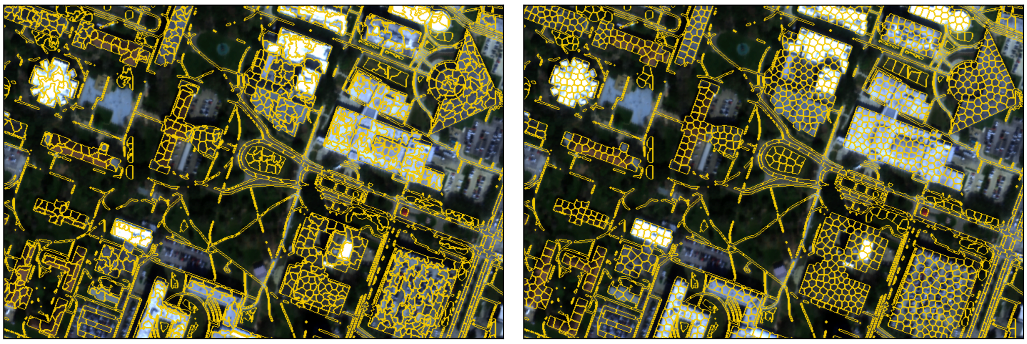
Settles, B., 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool Publishers.

Tong Luo, Kramer, K., Samson, S., Remsen, A., Goldgof, D. B., Hall, L. O., Hopkins, T., 2004. Active learning to recognize multiple types of plankton. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 3, 478–481 Vol.3.

Tuia, D., Volpi, M., Copa, L., Kanevski, M., Munoz-Mari, J., 2011. A Survey of Active Learning Algorithms for Supervised Remote Sensing Image Classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(3), 606-617.

Zhou, W., Wang, J., Cadenasso, M. L., 2017. Effects of the spatial configuration of trees on urban heat mitigation: A comparative study. *Remote Sensing of Environment*, 195, 1-12. https://www.sciencedirect.com/science/article/pii/S0034425717301463.

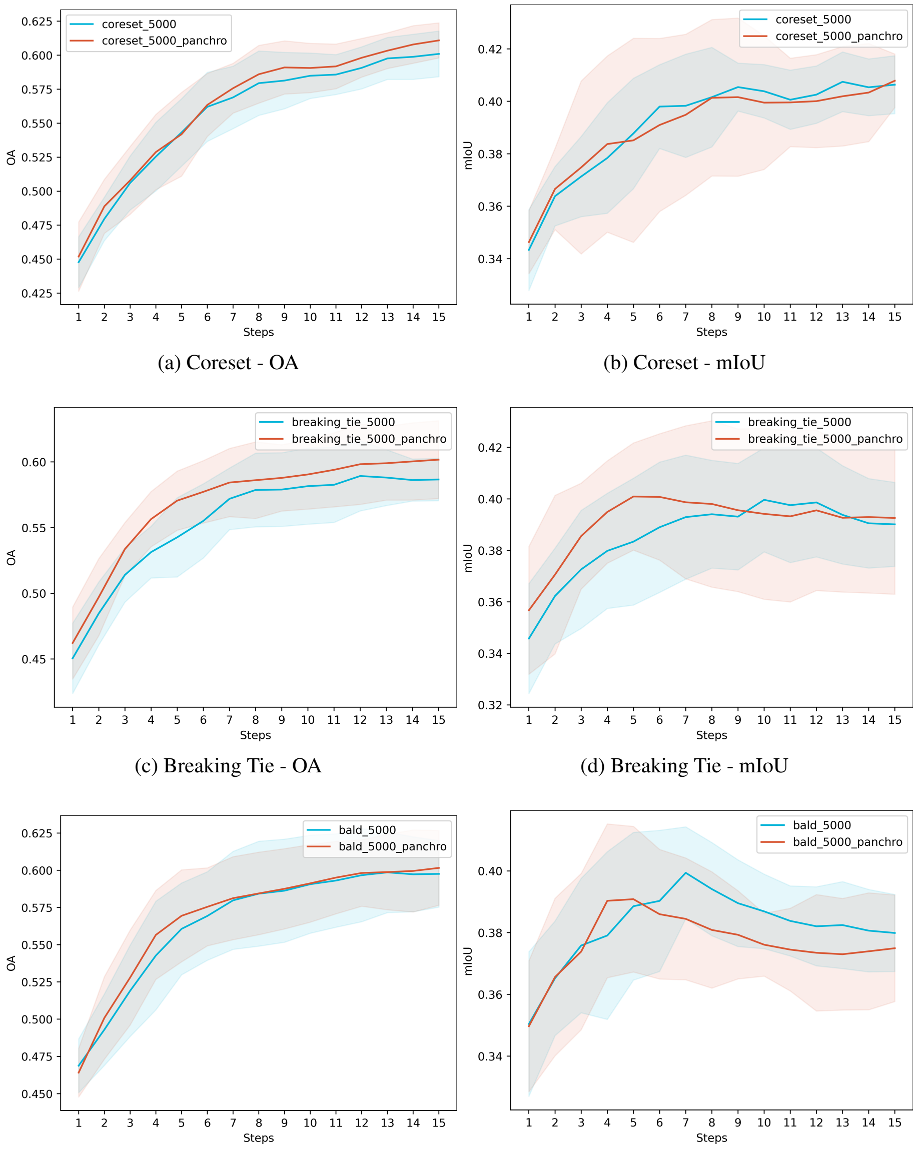
5. APPENDIX



(a) SLIC segmentation - PCA dimensionality reduction - 5000 regions

(b) SLIC segmentation - panchromatic dimensionality reduction - 5000 regions

Figure 8. Difference between the segmentation obtained after PCA and panchromatic dimensionality reduction



(a) Coreset - OA

(b) Coreset - mIoU

(c) Breaking Tie - OA

(d) Breaking Tie - mIoU

(e) BALD - OA

(f) BALD - mIoU

Figure 9. Accuracy metrics with a 5000-regions segmentation. Red curves correspond to the ”panchromatic” segmentation” while the blue curves correspond to the ”PCA” segmentation.